User's Manual

32

RA8D1 Group

User's Manual: Hardware

32-Bit MCU
Renesas Advanced (RA) Family
Renesas RA8 Series

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

Renesas Electronics
www.renesas.com

Rev.1.30 Dec 2025

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Contact Information

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

Trademarks

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

© 2025 Renesas Electronics Corporation. All rights reserved.

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

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

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

Preface

1. About This Document

This manual is generally organized into an overview of the product, descriptions of the CPU, system control functions, peripheral functions, electrical characteristics, and usage notes. This manual describes the product specification of the microcontroller (MCU) superset. Depending on your product, some pins, registers, or functions might not exist. Address space that store unavailable registers are reserved.

2. Audience

This manual is written for system designers who are designing and programming applications using the Renesas Microcontroller. The user is expected to have basic knowledge of electrical circuits, logic circuits, and the MCU.

3. Renesas Publications

Renesas provides the following documents. Before using any of these documents, visit www.renesas.com for the most up-to-date version of the document.

Component	Document Type	Description
Microcontrollers	Datasheet	Features, overview, and electrical characteristics of the MCU
	User's Manual: Hardware	MCU specifications such as pin assignments, memory maps, peripheral functions, electrical characteristics, timing diagrams, and operation descriptions
	Application Notes	Technical notes, board design guidelines, and software migration information
	Technical Update (TU)	Preliminary reports on product specifications such as restriction and errata
Software	User's Manual: Software	API reference and programming information
	Application Notes	Project files, guidelines for software programming, and application examples to develop embedded software applications
Tools & Kits, Solutions	User's Manual: Development Tools	User's manual and quick start guide for developing embedded software applications with Development Kits (DK), Starter Kits (SK), Promotion Kits (PK), Product Examples (PE), and Application Examples (AE)
	User's Manual: Software	
	Quick Start Guide	
	Application Notes	Project files, guidelines for software programming, and application examples to develop embedded software applications

4. Numbering Notation

The following numbering notation is used throughout this manual:

Example	Description
011b	Binary number. For example, the binary equivalent of the number 3 is 011b.
0x1F	Hexadecimal number. For example, the hexadecimal equivalent of the number 31 is described 0x1F. In some cases, a hexadecimal number is shown with the suffix "h".
1234	Decimal number. A decimal number is followed by this symbol only when the possibility of confusion exists. Decimal numbers are generally shown without a suffix.

5. Typographic Notation

The following typographic notation is used throughout this manual:

Example	Description
AAA.BBB.CCC	Periods separated a function module symbol (AAA), register symbol (BBB), and bit field symbol (CCC).
AAA.BBB	A period separated a function module symbol (AAA) and register symbol (BBB).
BBB.DDD	A period separated a register symbol (BBB) and bit field symbol (DDD).
EEE[3:0]	Numbers in brackets expresses a bit number. For example, EEE[3:0] occupies bits 3 to 0.

6. Unit and Unit Prefix

The following units and unit prefixes are sometimes misleading. Those unit prefixes are described throughout this manual with the following meaning:

Symbol	Name	Description
b	Binary Digit	Single 0 or 1
B	Byte	This unit is generally used for memory specification of the MCU and address space.
k	kilo-	$1000 = 10^3$. k is also used to denote 1024 (2^{10}) but this unit prefix is used to denote 1000 (10^3) throughout this manual.
K	Kilo-	$1024 = 2^{10}$. This unit prefix is used to denote 1024 (2^{10}) not 1000 (10^3) throughout this manual.

7. Special Terms

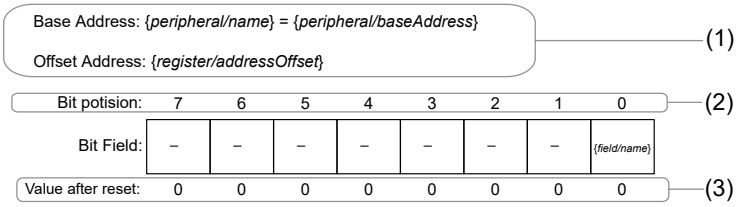
The following terms have special meanings.

Term	Description
NC	Not connected pin. This pin should be left floating unless specified otherwise.
Hi-Z	High impedance.
x	Don't care or undefined.

8. Register Description

Each register description includes both a register diagram that shows the bit assignments and a register bit table that describes the content of each bit. The example of symbols used in these tables are described in the sections that follow. The following is an example of a register description and associated bit field definition.

XX.X.X {*register/name*} : {*register/description*}



(4) (5) (6)

Bit	Symbol	Function	R/W
0	{*field/name*}	{*field/description*} 0: {*enumeratedValue/description*} 1: {*enumeratedValue/description*}	{*access*}
7:1	–	These bits are read as 0. The write value should be 0.	R/W

(1) Function module symbol, register symbol, and address assignment

Function module symbol, {*pheripheral/name*}, register symbol, {*register/name*}, and address assignment of this register are generally expressed. Base address and Offset address mean {*regiser/name*} : {*register/description*} of {*peripheral/name*} is assigned to address {*peripheral/baseAddress*} + {*register/addressOffset*}.

(2) Bit number

This number indicates the bit number. This bits are shown in order from bits 31 to 0 for 32-bit register, from bits 15 to 0 for 16-bit register, and from bits 7 to 0 for 8-bit register.

(3) Value after reset

This symbol or number indicate the value of each bit after a hard reset. The value is shown in binary unless specified otherwise.

- 0: Indicates that the value is 0 after a reset.
- 1: Indicates that the value is 1 after a reset.
- x: Indicates that the value is undefined after a reset.

(4) Symbol

{*filed/name*} indicates the short name of bit field. Reserved bit is expressed with a —.

(5) Function

Function indicates the full name of the bit field, {*field/description*}, and enumerated values.

(6) R/W

The R/W column indicates access type whether the bit field is readable or writable.

- R/W: The bit field is readable and writable.
- R: The bit field is readable only. Writing to this bit field has no effect.
- W: The bit field is writable only. The read value is the same as after a reset unless specified otherwise.

9. Abbreviations

Abbreviations used in this document are shown in the following table.

Abbreviation	Description
AES	Advanced Encryption Standard
AHB	Advanced High-performance Bus
AHB-AP	AHB Access Port
APB	Advanced Peripheral Bus
ARC	Alleged RC
ATB	Advanced Trace Bus
BCD	Binary Coded Decimal
BSDL	Boundary Scan Description Language
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
ETB	Embedded Trace Buffer
ETM	Embedded Trace Macrocell
FLL	Frequency Locked Loop
FPU	Floating Point Unit
HMI	Human Machine Interface
IrDA	Infrared Data Association
LSB	Least Significant Bit
MSB	Most Significant Bit
NVIC	Nested Vector Interrupt Controller
PC	Program Counter
PFS	Port Function Select
PLL	Phase Locked Loop
POR	Power-on reset
PWM	Pulse Width Modulation
RSA	Rivest Shamir Adleman
SHA	Secure Hash Algorithm
S/H	Sample and Hold
SP	Stack Pointer
SWD	Serial Wire Debug
SW-DP	Serial Wire-Debug Port
TRNG	True Random Number Generator
UART	Universal Asynchronous Receiver/Transmitter
VCO	Voltage Controlled Oscillator

10. Proprietary Notice

All text, graphics, photographs, trademarks, logos, artwork and computer code, collectively known as content, contained in this document is owned, controlled or licensed by or to Renesas, and is protected by trade dress, copyright, patent and trademark laws, and other intellectual property rights and unfair competition laws. Except as expressly provided herein, no part of this document or content may be copied, reproduced, republished, posted, publicly displayed, encoded, translated, transmitted or distributed in any other medium for publication or distribution or for any commercial enterprise, without prior written consent from Renesas.

Arm® and Cortex® are registered trademarks of Arm Limited. CoreSight™ is a trademark of Arm Limited.

CoreMark® is a registered trademark of the Embedded Microprocessor Benchmark Consortium.

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

Other brands and names mentioned in this document may be the trademarks or registered trademarks of their respective holders.

11. Feedback on the Product

If you have any comments or suggestions about this product, go to Contact Us.

Contents

Features **59**
1. Overview **60**
1.1 Function Outline 60
1.2 Block Diagram 67
1.3 Part Numbering 67
1.4 Function Comparison 69
1.5 Pin Functions 71
1.6 Pin Assignments 78
1.7 Pin Lists 82
2. CPU **89**
2.1 Overview 89
2.1.1 CPU 89
2.1.2 Debug 90
2.2 Block Diagram 90
2.3 Implementation Options 91
2.4 Trace Interface 92
2.5 JTAG/SWD Interface 92
2.6 Initial Vector Table Base Address 93
2.7 Debug Function 93
2.7.1 External Debugger Connectivity and Authentication Through JTAG/SWD 93
2.7.2 Debug Protection Mechanism by Software 94
2.7.3 Effect of Debug Function 95
2.8 Registers Description 96
2.8.1 CPU Control Registers 96
2.8.2 Address Spaces for External Debugger 98
2.8.3 CPU Peripheral 99
2.8.4 ROM Table 100
2.8.5 DBGREG 102
2.8.6 OCDREG 109
2.9 CoreSight Cross Trigger Interface (CTI) 115
2.10 CoreSight ATB Funnel 115
2.11 SysTick System Timer 116
2.12 CoreSight Timestamp Generator 117
2.13 OCD Emulator Connection 117
2.13.1 SYOCDCR.DBGEN 119
2.13.2 Restriction in OCD Emulator Connecting 119
2.13.3 Starting Connection While in Low Power Mode 119
2.13.4 Low Power Mode Change During Debugging 119
2.13.5 Connecting Sequence and JTAG/SWD Authentication 120
2.14 References 120
2.15 Usage Notes 121
2.15.1 Restrictions for OSPI 121
3. Operating Modes **122**
3.1 Overview 122
3.2 Operating Mode Types and Selection 122
3.3 Details of Operating Modes 122
3.3.1 Single-Chip Mode 122
3.3.2 JTAG Boot Mode 122

3.3.3 SCI Boot Mode ... 122
3.3.4 USB Boot Mode ... 122
3.4 Operating Modes Transitions ... 122
3.4.1 Operation modes and the relationship of Mode Transition ... 122
4. Address Space ... 124
4.1 Address Space ... 124
4.2 External Address Space ... 126
5. Resets ... 128
5.1 Overview ... 128
5.2 Register Descriptions ... 134
5.2.1 RSTSAR : Reset Security Attribution Register ... 134
5.2.2 RSTSR0 : Reset Status Register 0 ... 135
5.2.3 RSTSR1 : Reset Status Register 1 ... 137
5.2.4 RSTSR2 : Reset Status Register 2 ... 139
5.2.5 SYRSTMSK0 : System Reset Mask Control Register 0 ... 139
5.2.6 SYRSTMSK2 : System Reset Mask Control Register 2 ... 140
5.3 Operation ... 140
5.3.1 RES Pin Reset ... 140
5.3.2 Power-On Reset ... 140
5.3.3 Voltage Monitor Reset ... 141
5.3.4 Deep Software Standby Reset ... 144
5.3.5 Independent Watchdog Timer Reset ... 144
5.3.6 Watchdog Timer Reset ... 145
5.3.7 CPU Lockup Reset ... 145
5.3.8 Software Reset ... 145
5.3.9 Bus Error Reset ... 145
5.3.10 Common Memory Error Reset ... 145
5.3.11 Determination of Cold/Warm Start ... 146
5.3.12 Determination of Reset Generation Source ... 146
5.4 Reset Protection from Non-secure Programs ... 147
6. Option-Setting Memory ... 148
6.1 Overview ... 148
6.2 Register Descriptions ... 149
6.2.1 OFS0 : Option Function Select Register 0 ... 149
6.2.2 OFS2 : Option Function Select Register 2 ... 152
6.2.3 DUALSEL : Dual Mode Select Register ... 152
6.2.4 SAS : Startup Area Setting Register ... 153
6.2.5 OFS1, OFS1_SEC : Option Function Select Register 1 for Non-secure and Secure ... 154
6.2.6 OFS1_SEL : Option Function Select Register 1 for Security Attribution ... 155
6.2.7 BANKSEL, BANKSEL_SEC : Bank Select Register for Non-secure and Secure ... 156
6.2.8 BANKSEL_SEL : Bank Select Register for Security Attribution ... 159
6.2.9 BPS, BPS_SEC, BPS_SEL : Block Protect Setting Register ... 160
6.2.10 PBPS, PBPS_SEC : Permanent Block Protect Setting Register ... 162
6.2.11 FSBLCTRL0 : FSBL Control Register 0 ... 162
6.2.12 FSBLCTRL1 : FSBL Control Register 1 ... 163
6.2.13 FSBLCTRL2 : FSBL Control Register 2 ... 163
6.2.14 SACC0 : Start Address of Code Certificate 0 ... 164
6.2.15 SACC1 : Start Address of Code Certificate 1 ... 164
6.2.16 SAMR : Start Address of Measurement Report ... 164
6.2.17 HOEMRTPK :Hash of OEM_ROOT_PK ... 165

6.2.18 CFGD0LOCK : Configuration Data 0 Lock Bit ... 165
6.2.19 CFGD1LOCK : Configuration Data 1 Lock Bit ... 166
6.2.20 CFGD2LOCK : Configuration Data 2 Lock Bit ... 166
6.2.21 ARCLS : Anti-Rollback Counter Lock Setting ... 167
6.2.22 ARCCS : Anti-Rollback Counter Configuration Setting for Non-secure Application ... 168
6.2.23 ARC_SECn : Anti-Rollback Counter for Secure Application n (n = 0, 1) ... 168
6.2.24 ARC_NSECn : Anti-Rollback Counter for Non-secure Application n (n = 0 to 7) ... 169
6.2.25 ARC_OEMBLn : Anti-Rollback Counter for OEMBL (n = 0, 1) ... 169
6.3 Setting Option-Setting Memory ... 169
6.3.1 Allocation of Data in Option-Setting Memory ... 169
6.3.2 Setting Data for Programming Option-Setting Memory ... 169
6.3.3 Security Attribution of Code Flash Option-setting Memory ... 170
6.3.4 Timing of the Setting Value ... 170
6.4 Usage Notes ... 170
6.4.1 Data for Programming Reserved Areas and Reserved Bits in the Option-Setting Memory 170
6.4.2 FSBL Execution After System Reset ... 171
6.4.3 FSBL Skip Function ... 171
6.4.4 Notes on FSBL Execution ... 171
7. Programmable Voltage Detection (PVD) ... 172
7.1 Overview ... 172
7.2 Register Descriptions ... 173
7.2.1 PVDSAR : Programable Voltage Detection Security Attribution Register ... 173
7.2.2 PVDmCMPCR : Voltage Monitor m Comparator Control Register (m = 1, 2) ... 174
7.2.3 PVDmCR0 : Voltage Monitor m Circuit Control Register 0 (m = 1, 2) ... 175
7.2.4 PVDmCR1 : Voltage Monitor m Circuit Control Register 1 (m = 1, 2) ... 176
7.2.5 PVDmSR : Voltage Monitor m Circuit Status Register (m = 1, 2) ... 177
7.2.6 PVDmFCR : Voltage Monitor m Function Control Register (m = 1, 2) ... 177
7.3 VCC Input Voltage Monitor ... 178
7.3.1 Monitoring V_{det0} ... 178
7.3.2 Monitoring V_{detm} ... 178
7.4 Reset from Voltage Monitor 0 ... 178
7.5 Interrupt and Reset from Voltage Monitor m (m = 1, 2) ... 179
7.6 Event Link Controller (ELC) Output ... 182
7.6.1 Interrupt Handling and Event Linking ... 183
8. Clock Generation Circuit ... 184
8.1 Overview ... 184
8.2 Register Descriptions ... 189
8.2.1 CGFSAR : Clock Generation Function Security Attribute Register ... 189
8.2.2 SCKDIVCR : System Clock Division Control Register ... 192
8.2.3 SCKDIVCR2 : System Clock Division Control Register 2 ... 195
8.2.4 SYRACCR : System Register Access Control Register ... 195
8.2.5 SCKSCR : System Clock Source Control Register ... 196
8.2.6 PLLCCR : PLL Clock Control Register ... 197
8.2.7 PLLCCR2 : PLL Clock Control Register 2 ... 198
8.2.8 PLLCR : PLL Control Register ... 199
8.2.9 PLL2CCR : PLL2 Clock Control Register ... 199
8.2.10 PLL2CCR2 : PLL2 Clock Control Register 2 ... 201
8.2.11 PLL2CR : PLL2 Control Register ... 202
8.2.12 BCKCR : External Bus Clock Control Register ... 202
8.2.13 MOSCCR : Main Clock Oscillator Control Register ... 203

8.2.14 SOSCCR : Sub-Clock Oscillator Control Register 204
8.2.15 LOCOCR : Low-Speed On-Chip Oscillator Control Register 205
8.2.16 HOCOCR : High-Speed On-Chip Oscillator Control Register 205
8.2.17 HOCOCR2 : High-Speed On-Chip Oscillator Control Register2 206
8.2.18 MOCOCR : Middle-Speed On-Chip Oscillator Control Register 207
8.2.19 FLLCR1 : FLL Control Register1 207
8.2.20 FLLCR2 : FLL Control Register2 208
8.2.21 OSCSF : Oscillation Stabilization Flag Register 209
8.2.22 OSCMONR : Oscillator Monitor Register 210
8.2.23 OSTDCR : Oscillation Stop Detection Control Register 211
8.2.24 OSTDSR : Oscillation Stop Detection Status Register 212
8.2.25 MOSCWTCR : Main Clock Oscillator Wait Control Register 213
8.2.26 MOMCR : Main Clock Oscillator Mode Oscillation Control Register 213
8.2.27 SOMCR : Sub-Clock Oscillator Mode Control Register 214
8.2.28 CKOCR : Clock Out Control Register 215
8.2.29 EBCKOCR : External Bus Clock Output Control Register 216
8.2.30 SDCKOCR : SDRAM Clock Output Control Register 216
8.2.31 LOCOUTCR : LOCO User Trimming Control Register 216
8.2.32 MOCOUTCR : MOCO User Trimming Control Register 217
8.2.33 HOCOUTCR : HOCO User Trimming Control Register 218
8.2.34 MOSCSCR : Main Clock Oscillator Standby Control Register 218
8.2.35 HOCOSCR : High-Speed On-Chip Oscillator Standby Control Register 219
8.2.36 USBCKDIVCR : USB Clock Division Control Register 219
8.2.37 OCTACKDIVCR : Octal-SPI Clock Division Control Register 220
8.2.38 CANFDCKDIVCR : CANFD Core Clock Division Control Register 220
8.2.39 USB60CKDIVCR : USB60 Clock Division Control Register 221
8.2.40 I3CCKDIVCR : I3C Clock Division Control Register 221
8.2.41 USBCKCR : USB Clock Control Register 222
8.2.42 OCTACKCR : Octal-SPI Clock Control Register 223
8.2.43 CANFDCKCR : CANFD Core Clock Control Register 224
8.2.44 USB60CKCR : USB60 Clock Control Register 225
8.2.45 I3CCKCR : I3C Clock Control Register 227
8.2.46 SCICKDIVCR : SCI Clock Division Control Register 228
8.2.47 SPICKDIVCR : SPI Clock Division Control Register 228
8.2.48 LCDCKDIVCR : LCD Clock Division Control Register 229
8.2.49 SCICKCR : SCI Clock Control Register 229
8.2.50 SPICKCR : SPI Clock Control Register 231
8.2.51 LCDCKCR : LCD Clock Control Register 232
8.2.52 TRCKCR : Trace Clock Control Register 233
8.3 Main Clock Oscillator 234
8.3.1 Connecting a Crystal Resonator 234
8.3.2 External Clock Input 235
8.3.3 Notes on External Clock Input 235
8.3.4 Note on Oscillation Keep in Software Standby mode 236
8.4 Sub-Clock Oscillator 236
8.4.1 Connecting a 32.768-kHz Crystal Resonator 236
8.4.2 Pin Handling When the Sub-Clock Oscillator Is Not Used 237
8.4.3 External Clock Input 237
8.5 Low speed On Chip Oscillator (LOCO) 237
8.5.1 LOCO Operating conditions 237

8.5.2 LOCO Stop conditions 238
8.5.3 Restrictions on LOCO operation and stopping 238
8.6 Middle speed On Chip Oscillator (MOCO) 239
8.6.1 MOCO Operating conditions 239
8.6.2 MOCO Stop conditions 239
8.6.3 Restrictions on MOCO operation and stopping 239
8.7 High speed On Chip Oscillator (HOCO) 240
8.7.1 Frequency Locked Loop Function 240
8.7.2 Note on Oscillation Keep in Software Standby mode 241
8.7.3 Note on using as TRACE clock source 241
8.8 PLL Circuit 241
8.9 Oscillation Stop Detection Function 242
8.9.1 Oscillation Stop Detection and Operation after Detection 242
8.9.2 Oscillation Stop Detection Interrupts 244
8.10 Internal Clock 245
8.10.1 CPU Clock (CPUCLK) 246
8.10.2 System Clock (ICLK) 246
8.10.3 Debugger Clock (DCLK) 247
8.10.4 Peripheral Module Clock (PCLKA, PCLKB, PCLKC, PCLKD, PCLKE) 247
8.10.5 FlashIF Clock (FCLK) 247
8.10.6 External Bus Clock (BCLK, EBCLK) 248
8.10.7 SDRAM Clock (SDCLK) 248
8.10.8 Trace Clock (TRCLK) 248
8.10.9 SCI Clock (SCICLK) 249
8.10.10 SPI Clock (SPICLK) 249
8.10.11 Octal-SPI clock (OCTACLK, OCTADIVCLK) 249
8.10.12 CANFD Core clock (CANFDCLK) 250
8.10.13 USB Clock (USBCLK) 250
8.10.14 USB Clock (USB60CLK) 250
8.10.15 MOSC clock for USBHS (USBMCLK) 251
8.10.16 LCD Clock (LCDCLK) 251
8.10.17 I3C Clock (I3CCLK) 251
8.10.18 CAN Clock (CANMCLK) 251
8.10.19 MIPI Clock (MIPIMCLK) 251
8.10.20 ULPT Clock (ULPTLCLK, ULPTSCLK) 252
8.10.21 CAC Clock (CACCLK) 252
8.10.22 RTC Clock (RTCSCLK, RTCLCLK) 252
8.10.23 IWDT Clock (IWDTCLK) 252
8.10.24 AGT Clock (AGTSCLK, AGTLCLK) 252
8.10.25 SysTick Timer Clock (SYSTICKCLK) 252
8.10.26 External Pin Output Clock (CLKOUT) 252
8.10.27 JTAG Clock (JTAGTCK) 252
8.11 Clock Setting 252
8.11.1 System Clock Setting 253
8.11.2 Peripheral module-dedicated Clock Setting 258
8.11.3 Main Clock Oscillator Setting 260
8.11.4 Sub-Clock Oscillator Setting 260
8.11.5 High-speed On Chip Oscillator Setting 261
8.11.6 PLL Setting 262
8.12 Usage Notes 264

8.12.1 Notes on Clock Generation Circuit 264
8.12.2 Notes on Resonator 265
8.12.3 Notes on Board Design 265
8.12.4 Notes on Resonator Connect Pin 265
8.12.5 Notes on Using Sub-Clock Oscillator 265
8.12.6 Notes on MOSCSCR.MOSCSOKP 265
9. Clock Frequency Accuracy Measurement Circuit (CAC) 268
9.1 Overview 268
9.2 Register Descriptions 269
9.2.1 CACR0 : CAC Control Register 0 269
9.2.2 CACR1 : CAC Control Register 1 270
9.2.3 CACR2 : CAC Control Register 2 271
9.2.4 CAICR : CAC Interrupt Control Register 272
9.2.5 CASTR : CAC Status Register 273
9.2.6 CAULVR : CAC Upper-Limit Value Setting Register 274
9.2.7 CALLVR : CAC Lower-Limit Value Setting Register 274
9.2.8 CACNTBR : CAC Counter Buffer Register 274
9.3 Operation 275
9.3.1 Measuring Clock Frequency 275
9.3.2 Digital Filtering of Signals on CACREF Pin 276
9.4 Interrupt Requests 276
9.5 Usage Notes 276
9.5.1 Settings for the Module-Stop Function 276
10. Low Power Modes 277
10.1 Overview 277
10.2 Register Descriptions 284
10.2.1 LPMSAR : Low Power Mode Security Attribution Register 284
10.2.2 PGCSAR : Power Gating Control Security Attribution Register 285
10.2.3 DPFSAR : Deep Software Standby Interrupt Factor Security Attribution Register 286
10.2.4 RSCSAR : RAM Standby Control Security Attribution Register 288
10.2.5 MSTPCRA : Module Stop Control Register A 289
10.2.6 MSTPCRB : Module Stop Control Register B 290
10.2.7 MSTPCRC : Module Stop Control Register C 291
10.2.8 MSTPCRD : Module Stop Control Register D 293
10.2.9 MSTPCRE : Module Stop Control Register E 294
10.2.10 SBYCR : Standby Control Register 296
10.2.11 OPCCR : Operating Power Control Register 296
10.2.12 PDCTRGD : Graphics Power Domain Control Register 298
10.2.13 PDRAMSCR0 : SRAM Power Domain Standby Control Register 0 298
10.2.14 PDRAMSCR1 : SRAM Power Domain Standby Control Register 1 299
10.2.15 SSCR1 : Software Standby Control Register 1 300
10.2.16 LPSCR : Low Power State Control Register 300
10.2.17 DPSBYCR : Deep Software Standby Control Register 301
10.2.18 DPSWCR : Deep Software Standby Wait Control Register 302
10.2.19 DPSIER0 : Deep Software Standby Interrupt Enable Register 0 303
10.2.20 DPSIER1 : Deep Software Standby Interrupt Enable Register 1 304
10.2.21 DPSIER2 : Deep Software Standby Interrupt Enable Register 2 304
10.2.22 DPSIER3 : Deep Software Standby Interrupt Enable Register 3 305
10.2.23 DPSIFR0 : Deep Software Standby Interrupt Flag Register 0 306
10.2.24 DPSIFR1 : Deep Software Standby Interrupt Flag Register 1 307

10.2.25 DPSIFR2 : Deep Software Standby Interrupt Flag Register 2 308
10.2.26 DPSIFR3 : Deep Software Standby Interrupt Flag Register 3 309
10.2.27 DPSIEGR0 : Deep Software Standby Interrupt Edge Register 0 311
10.2.28 DPSIEGR1 : Deep Software Standby Interrupt Edge Register 1 312
10.2.29 DPSIEGR2 : Deep Software Standby Interrupt Edge Register 2 313
10.2.30 SYOCDCR : System Control OCD Control Register 313
10.2.31 PLL1LDOCR : PLL1-LDO Control Register 314
10.2.32 PLL2LDOCR : PLL2-LDO Control Register 314
10.2.33 HOCOLDOCR : HOCO-LDO Control Register 315
10.3 Reducing Power Consumption by Switching Clock Signals 316
10.4 Module-Stop Function 316
10.5 Function for Lower Operating Power Consumption 318
10.5.1 Setting Operating Power Control Mode 318
10.6 Power Gating Control 318
10.6.1 Power Gating Control Using Power Domain Control Register 318
10.7 Low Power Modes 319
10.7.1 CPU Sleep Mode 319
10.7.2 CPU Deep Sleep Mode 321
10.7.3 Software Standby Mode 322
10.7.4 Deep Software Standby Mode 325
10.8 Usage Notes 328
10.8.1 Register Access 328
10.8.2 I/O Port States 330
10.8.3 Module-Stop State of DMAC and DTC 330
10.8.4 Internal Interrupt Sources 330
10.8.5 Input Buffer Control by DIRQnE Bit 330
10.8.6 Transition to Low Power Modes 330
10.8.7 Timing of WFI Instructions 330
10.8.8 Writing WDT/IWDT Registers by DMAC or DTC 331
10.8.9 Module Stop Bit Write Timing 331
10.8.10 Notes on Transitioning to or Canceling Low Power State 331
10.8.11 Interruption During Transition to Low Power Modes 333
11. Battery Backup Function 337
11.1 Overview 337
11.1.1 Battery backup function block diagram 337
11.1.2 Features of Battery Backup Function 338
11.1.3 Battery Power Supply Switch 339
11.1.4 VBATT_R voltage drop detection function 339
11.1.5 Backup Registers 339
11.1.6 Time Capture Function 339
11.1.7 Tamper Detection Function 339
11.1.8 VBATT Voltage Monitor Function 339
11.2 Register Descriptions 339
11.2.1 BBFSAR : Battery Backup Function Security Attribute Register 339
11.2.2 VBRSABAR : VBATT Backup Register Security Attribute Boundary Address Register 340
11.2.3 VBRPABARS : VBATT Backup Register Privilege Attribute Boundary Address Register for Secure Region 341
11.2.4 VBRPABARNS : VBATT Backup Register Privilege Attribute Boundary Address Register for Non-secure Region 342
11.2.5 VBATTMNSELR : Battery Backup Voltage Monitor Function Select Register 342

11.2.6 VBTBER : VBATT Backup Enable Register 343
11.2.7 VBTBKR[n] : VBATT Backup Register (n = 0 to 127) 343
11.2.8 VBTICTLR : VBATT Input Control Register 344
11.2.9 VBTICTLR2 : VBATT Input Control Register 2 345
11.2.10 VBTIMONR : VBATT Input Monitor Register 345
11.2.11 VBTBPCR1 : VBATT Battery Power Supply Control Register 1 346
11.2.12 VBTBPCR2 : VBATT Battery Power Supply Control Register 2 346
11.2.13 VBTBPSR : VBATT Battery Power Supply Status Register 347
11.2.14 VBTADSR : VBATT Tamper detection Status Register 348
11.2.15 VBTADCR1 : VBATT Tamper detection Control Register 1 349
11.2.16 VBTADCR2 : VBATT Tamper detection Control Register 2 350
11.3 Operation 350
11.3.1 Battery Backup Function 350
11.3.2 VBATT Battery Power Supply Switch Usage 352
11.3.3 VBATT_R voltage drop detection function Usage 353
11.3.4 VBATT Backup Register Usage 353
11.3.5 Tamper Detection Usage 353
11.3.6 VBATT voltage monitor function Usage 354
11.3.7 Initial Settings Examples 355
11.4 Interrupt Sources 356
11.5 Usage Notes 356
12. Register Write Protection 357
12.1 Overview 357
12.2 Register Descriptions 357
12.2.1 PRCR_S : Protect Register for Secure (PRCR_S) 357
12.2.2 PRCR_NS : Protect Register for Non-secure (PRCR_NS) 358
13. Interrupt Controller Unit (ICU) 360
13.1 Overview 360
13.2 Register Descriptions 361
13.2.1 ICUSARA : Interrupt Controller Unit Security Attribution Register A 361
13.2.2 ICUSARB : Interrupt Controller Unit Security Attribution Register B 362
13.2.3 ICUSARE : Interrupt Controller Unit Security Attribution Register E 363
13.2.4 ICUSARF : Interrupt Controller Unit Security Attribution Register F 364
13.2.5 ICUSARG : Interrupt Controller Unit Security Attribution Register G 365
13.2.6 ICUSARH : Interrupt Controller Unit Security Attribution Register H 365
13.2.7 ICUSARI : Interrupt Controller Unit Security Attribution Register I 366
13.2.8 TEVTRCR : Trusted Event Route Control Register 366
13.2.9 IRQCRi : IRQ Control Register i (i = 0 to 15) 367
13.2.10 NMISR : Non-Maskable Interrupt Status Register 368
13.2.11 NMIER : Non-Maskable Interrupt Enable Register 371
13.2.12 NMICLR : Non-Maskable Interrupt Status Clear Register 372
13.2.13 NMICR : NMI Pin Interrupt Control Register 373
13.2.14 IELSRn : ICU Event Link Setting Register n (n = 0 to 95) 374
13.2.15 WUPEN0 : Wake Up Interrupt Enable Register 0 376
13.2.16 WUPEN1 : Wake Up interrupt enable register 1 379
13.3 Vector Table 380
13.3.1 Interrupt Vector Table 380
13.3.2 Event Number 383
13.4 Maskable Interrupt Operation 393
13.4.1 Detecting Interrupts 393

13.5 Maskable Interrupt Setting Procedure 394
13.5.1 Operations During an Interrupt 394
13.5.2 Enabling Interrupt Requests 394
13.5.3 Disabling Interrupt Requests 394
13.5.4 Polling for Interrupts 395
13.5.5 Selecting Interrupt Request Destinations 395
13.5.6 Digital Filter 396
13.5.7 External Pin Interrupts 396
13.6 Non-Maskable Interrupt Operation 397
13.6.1 Correspondence to TrustZone-M by NMI 397
13.7 Security 398
13.7.1 Security Related to CPU Interrupt Inputs 398
13.7.2 Trusted Interrupt Management 399
13.7.3 Trusted IELSR Setting Procedure 399
13.8 Return from Low Power Modes 400
13.8.1 Return from CPU Sleep Mode 400
13.8.2 Return from CPU Deep Sleep Mode 400
13.8.3 Return from Software Standby Mode 401
13.9 Using the WFI Instruction with Non-Maskable Interrupts 401
13.10 Reference 401
14. Buses 402
14.1 Overview 402
14.2 Description of Buses 405
14.2.1 Arbitration 405
14.2.2 External Bus 405
14.2.3 Parallel Operation 408
14.2.4 Bus Settings 408
14.2.5 Restrictions 408
14.3 Register Descriptions 409
14.3.1 BUSSARA : Bus Security Attribution Register A 409
14.3.2 BUSSARB : Bus Security Attribution Register B 409
14.3.3 BUSSARC : Bus Security Attribution Register C 410
14.3.4 BUSPARC : Bus Privileged Attribution Register C 411
14.3.5 CSnCR : CSn Control Register (n = 0 to 7) 412
14.3.6 CSnREC : CSn Recovery Cycle Register (n = 0 to 7) 413
14.3.7 CSRECEN : CS Recovery Cycle Insertion Enable Register 414
14.3.8 CSnMOD : CSn Mode Register (n = 0 to 7) 416
14.3.9 CSnWCR1 : CSn Wait Control Register 1 (n = 0 to 7) 417
14.3.10 CSnWCR2 : CSn Wait Control Register 2 (n = 0 to 7) 419
14.3.11 SDCCR : SDC Control Register 422
14.3.12 SDCMOD : SDC Mode Register 422
14.3.13 SDAMOD : SDRAM Access Mode Register 423
14.3.14 SDSELF : SDRAM Self-Refresh Control Register 423
14.3.15 SDRFCR : SDRAM Refresh Control Register 424
14.3.16 SDRFEN : SDRAM Auto-Refresh Control Register 425
14.3.17 SDICR : SDRAM Initialization Sequence Control Register 425
14.3.18 SDIR : SDRAM Initialization Register 426
14.3.19 SDADR : SDRAM Address Register 427
14.3.20 SDTR : SDRAM Timing Register 428
14.3.21 SDMOD : SDRAM Mode Register 429

14.3.22 SDSR : SDRAM Status Register 430
14.3.23 BUSDIVBYP : Bus Divider Bypass Register 431
14.3.24 BUSMABT : Bus Master Arbitration Control Register 432
14.3.25 BUSSABT0<slave> : Bus Slave Arbitration Control Register 0(<slave> = FLBI, STBYSBI, ECBI, EOBI, PBBI, PABI, PIBI, PSBI) 433
14.3.26 BUSSABT1<slave> : Bus Slave Arbitration Control Register 1(<slave> = FHBI, S0BI, S1BI) 434
14.3.27 BUSnERRADD : BUS Error Address Register (n = 4 to 9) 435
14.3.28 BUSnERRRW : BUS Error Read Write (n = 4 to 9) 436
14.3.29 BMSAnERRADD : Bus Master Security Attribution Unit Error Address (n = 4 to 9) 437
14.3.30 BMSAnERRRW : BUS Master Security Attribution Unit Error Read Write (n = 4 to 9) 438
14.3.31 BUSnERRSTAT : BUS Error Status Register (n = 1 to 12) 438
14.3.32 BUSnERRCLR : BUS Error Clear Register (n = 1 to 12) 440
14.3.33 BUSOAD : BUS Operation After Detection Register 441
14.3.34 BUSOADPT : BUS Operation After Detection Protect Register 442
14.3.35 MBWERRSTAT : Master Bufferable Write Error Status Register 442
14.3.36 MBWERRCLR : Master Bufferable Write Error Clear Register 444
14.3.37 SBWERRSTAT : Slave Bufferable Write Error Status Register 445
14.3.38 SBWERRCLR : Slave Bufferable Write Error Clear Register 446
14.4 Endian and Data Alignment 447
14.4.1 Data Alignment Control for the CS Area 447
14.4.2 Data Alignment Control for the SDRAM Area 452
14.5 Operation of CS Area Controller 457
14.5.1 Separate Bus 457
14.5.2 Address/Data Multiplexed Bus 471
14.5.3 External Wait Function 473
14.5.4 Insertion of Recovery Cycles 476
14.5.5 No Access State 479
14.5.6 Write Buffer Function (External Bus) 479
14.5.7 Limitations 480
14.6 SDRAM Area Controller Operation 481
14.6.1 Enabling/Disabling SDRAM Access and Setting the SDRAM Bus Width 481
14.6.2 No Access State 481
14.6.3 Insertion of Recovery Cycles 481
14.6.4 Write Buffer Function 482
14.6.5 SDRAM Commands 482
14.6.6 Conditions for Setting the SDRAMC Registers 482
14.6.7 Self-Refresh 483
14.6.8 Auto-Refresh 484
14.6.9 Initialization Sequencer 485
14.6.10 Read/Write Access 486
14.6.11 Setting the Mode Register 488
14.6.12 SDRAMC Setting Examples 489
14.6.13 Address Multiplexing 499
14.6.14 Example SDRAM Connections 500
14.6.15 Constraints 506
14.7 Bus Error Monitoring Section 507
14.7.1 Bus Error Types 507
14.7.2 Operations When a Bus Error Occurs 507
14.7.3 Conditions Leading to Illegal Address Access Errors 510

14.7.4 Timeout 512
14.8 References 512
14.9 Usage Notes 513
14.9.1 BC0 pin when in Software Standby Mode (SSTBY) and SBYCR.OPE=1 513
15. Memory Protection Unit (MPU) 514
15.1 Overview 514
15.2 Arm MPU 514
15.3 Bus master MPU 514
15.3.1 Register Descriptions 515
15.3.2 Operation 541
15.4 References 545
16. DMA Controller (DMAC) 546
16.1 Overview 546
16.2 Register Descriptions 548
16.2.1 DMACCHSAR : DMA channel Security Attribution Register 548
16.2.2 DMACCHPAR : DMA channel Privilege Attribution Register 549
16.2.3 DMACSAR : DMAC Controller Security Attribution Register 550
16.2.4 DELSRn : DMAC Event Link Setting Register n (n = 0 to 7) 550
16.2.5 DMSAR : DMA Source Address Register 552
16.2.6 DMSRR : DMA Source Reload Address Register 552
16.2.7 DMDAR : DMA Destination Address Register 552
16.2.8 DMDRR : DMA Destination Reload Address Register 553
16.2.9 DMCRA : DMA Transfer Count Register 553
16.2.10 DMCRB : DMA Block Transfer Count Register 554
16.2.11 DMTMD : DMA Transfer Mode Register 555
16.2.12 DMINT : DMA Interrupt Setting Register 556
16.2.13 DMAMD : DMA Address Mode Register 557
16.2.14 DMOFR : DMA Offset Register 560
16.2.15 DMCNT : DMA Transfer Enable Register 560
16.2.16 DMREQ : DMA Software Start Register 561
16.2.17 DMSTS : DMA Status Register 562
16.2.18 DMSBS : DMA Source Buffer Size Register 563
16.2.19 DMDBS : DMA Destination Buffer Size Register 564
16.2.20 DMBWR : DMA Bufferable Write Enable Register 565
16.2.21 DMAST : DMA Module Activation Register 566
16.2.22 DMECHR : DMAC Error Channel Register 566
16.3 Operation 568
16.3.1 Transfer Mode 568
16.3.2 Extended Repeat Area Function 577
16.3.3 Free-running Function 579
16.3.4 Address Update Function using Offset 580
16.3.5 Address Update Function in Repeat-Block Transfer Mode 585
16.3.6 Example of Using Repeat-Block Transfer Mode 587
16.3.7 Activation Sources 590
16.3.8 Operation Timing 590
16.3.9 Activating the DMAC 591
16.3.10 Starting DMA Transfer 593
16.3.11 Registers during DMA Transfer 593
16.3.12 Channel Priority 595
16.3.13 Channel Security 595

16.3.14 Channel Privilege 595
16.4 Ending DMA Transfer 595
16.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations 596
16.4.2 Transfer End by Repeat Size End Interrupt 596
16.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow 596
16.5 Processing on DMA Transfer Error 597
16.6 Interrupts 597
16.6.1 Transfer End Interrupt 597
16.6.2 Transfer Error Interrupt 599
16.7 Event Link 600
16.8 Low-Power Consumption Function 600
16.9 Usage Notes 600
16.9.1 DMA Transfer to External Devices 600
16.9.2 Access to the Registers during DMA Transfer 600
16.9.3 DMA Transfer to Reserved Areas 601
16.9.4 Setting of DMAC Event Link Setting Register of the Interrupt Controller Unit (DELSRn) 601
16.9.5 Suspending or Restarting DMAC Activation 601
16.9.6 Precautions for Resuming DMA Transfer 601
16.9.7 Restriction for OSPI 602
17. Data Transfer Controller (DTC) 603
17.1 Overview 603
17.2 Register Descriptions 604
17.2.1 DTCSAR : DTC Controller Security Attribution Register 604
17.2.2 MRA : DTC Mode Register A 605
17.2.3 MRB : DTC Mode Register B 606
17.2.4 SAR : DTC Transfer Source Register 607
17.2.5 DAR : DTC Transfer Destination Register 607
17.2.6 CRA : DTC Transfer Count Register A 607
17.2.7 CRB : DTC Transfer Count Register B 608
17.2.8 DTCCR : DTC Control Register 609
17.2.9 DTCCR_SEC : DTC Control Register for Secure Region 609
17.2.10 DTCVBR : DTC Vector Base Register 610
17.2.11 DTCVBR_SEC : DTC Vector Base Register for Secure Region 610
17.2.12 DTCST : DTC Module Start Register 610
17.2.13 DTCSTS : DTC Status Register 611
17.2.14 DTEVR : DTC Error Vector Register 612
17.3 Activation Sources 613
17.3.1 Allocating Transfer Information and DTC Vector Table 613
17.4 Operation 616
17.4.1 Transfer Information Read Skip Function 618
17.4.2 Transfer Information Write-Back Skip Function 618
17.4.3 Normal Transfer Mode 619
17.4.4 Repeat Transfer Mode 620
17.4.5 Block Transfer Mode 621
17.4.6 Chain Transfer 622
17.4.7 Operation Timing 623
17.4.8 Vector Security 626
17.4.9 Vector privilege 626
17.5 DTC Setting Procedure 626
17.6 Examples of DTC Usage 627

17.6.1 Normal Transfer 627
17.6.2 Chain transfer 627
17.6.3 Chain Transfer when Counter = 0 629
17.7 Processing on DTC Transfer Error 630
17.8 Interrupt 631
17.8.1 Interrupt Request of Transfer End 631
17.8.2 Interrupt Request of Transfer Error 631
17.9 Event Link 631
17.10 Low Power Consumption Function 631
17.11 Usage Notes 632
17.11.1 Transfer Information Start Address 632
17.11.2 Restriction for OSPI 632
18. Event Link Controller (ELC) 633
18.1 Overview 633
18.2 Register Descriptions 635
18.2.1 ELCR : Event Link Controller Register 635
18.2.2 ELSEGRn : Event Link Software Event Generation Register n (n = 0, 1) 635
18.2.3 ELSRn : Event Link Setting Register n (n = 0 to 17, 30) 636
18.2.4 ELCSARA : Event Link Controller Security Attribution Register A 644
18.2.5 ELCSARB : Event Link Controller Security Attribution Register B 644
18.2.6 ELCPARA : Event Link Controller Privilege Attribution Register A 645
18.2.7 ELCPARB : Event Link Controller Privilege Attribution Register B 646
18.3 Operation 646
18.3.1 Relation between Interrupt Handling and Event Linking 646
18.3.2 Linking Events 646
18.3.3 Example of Procedure for Linking Events 647
18.4 Usage Notes 647
18.4.1 Linking DMAC/DTC Transfer End Signals as Events 647
18.4.2 Setting Clocks 647
18.4.3 Module-Stop Function Setting 647
18.4.4 ELC Delay Time 647
19. I/O Ports 649
19.1 Overview 649
19.2 Register Descriptions 651
19.2.1 PCNTR1/PODR/PDR : Port Control Register 1 651
19.2.2 PCNTR2/EIDR/PIDR : Port Control Register 2 652
19.2.3 PCNTR3/PORR/POSR : Port Control Register 3 653
19.2.4 PCNTR4/EORR/EOSR : Port Control Register 4 654
19.2.5 PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0 to 9, A, B, n = 00 to 15) 655
19.2.6 PWPR_NS : Write-Protect Register for Non-secure 657
19.2.7 PFENET : Ethernet Control Register 657
19.2.8 PWPR_S : Write-Protect Register for Secure 658
19.2.9 PmSAR : Port Security Attribution register (m = 0 to 9, A, B) 658
19.2.10 LVOCR : Low Voltage Operation Control Register 659
19.3 Operation 659
19.3.1 General I/O Ports 659
19.3.2 Port Function Select 659
19.3.3 Port Group Function for ELC 660
19.4 Handling of Unused Pins 662

19.5 Usage Notes 663
19.5.1 Procedure for Specifying the Pin Functions 663
19.5.2 Procedure for Using Port Group Input 663
19.5.3 Port Output Data Register (PODR) Summary 663
19.5.4 Notes on Using Analog Functions 664
19.5.5 I/O Buffer Specification 664
19.5.6 Selecting Pins USB_DP and USB_DM 665
19.5.7 Pull-up/Pull-down Setting for P814 and P815 at USBFS/GPIO Function 666
19.6 Peripheral Select Settings for Each Product 666
20. Port Output Enable for GPT (POEG) 677
20.1 Overview 677
20.2 Register Descriptions 678
20.2.1 POEGGn : POEG Group n Setting Register (n = A to D) 678
20.3 Output-Disable Control Operation 679
20.3.1 Pin Input Level Detection Operation 680
20.3.2 Output-Disable Requests from the GPT 681
20.3.3 Comparator Interrupt Detection 681
20.3.4 Output-Disable Control Using Detection of Stopped Oscillation 681
20.3.5 Output-Disable Control Using Registers 681
20.3.6 Release from Output-Disable 681
20.4 Interrupt Sources 682
20.5 External Trigger Output to the GPT 682
20.6 Usage Notes 683
20.6.1 Transition to Software Standby Mode 683
20.6.2 Specifying Pins Associated with the GPT 683
21. General PWM Timer (GPT) 684
21.1 Overview 684
21.2 Register Descriptions 687
21.2.1 GTWP : General PWM Timer Write-Protection Register 687
21.2.2 GTSTR : General PWM Timer Software Start Register 689
21.2.3 GTSTP : General PWM Timer Software Stop Register 690
21.2.4 GTCLR : General PWM Timer Software Clear Register 691
21.2.5 GTSSR : General PWM Timer Start Source Select Register 692
21.2.6 GTPSR : General PWM Timer Stop Source Select Register 695
21.2.7 GTCSR : General PWM Timer Clear Source Select Register 698
21.2.8 GTUPSR : General PWM Timer Up Count Source Select Register 701
21.2.9 GTDNSR : General PWM Timer Down Count Source Select Register 704
21.2.10 GTICASR : General PWM Timer Input Capture Source Select Register A 708
21.2.11 GTICBSR : General PWM Timer Input Capture Source Select Register B 711
21.2.12 GTCR : General PWM Timer Control Register 714
21.2.13 GTUDDTYC : General PWM Timer Count Direction and Duty Setting Register 716
21.2.14 GTIOR : General PWM Timer I/O Control Register 718
21.2.15 GTINTAD : General PWM Timer Interrupt Output Setting Register 722
21.2.16 GTST : General PWM Timer Status Register 724
21.2.17 GTBER : General PWM Timer Buffer Enable Register 729
21.2.18 GTCNT : General PWM Timer Counter 731
21.2.19 GTCCRk : General PWM Timer Compare Capture Register k (k = A to F) 732
21.2.20 GTPR : General PWM Timer Cycle Setting Register 732
21.2.21 GTPBR : General PWM Timer Cycle Setting Buffer Register 733
21.2.22 GTADTRk : A/D Conversion Start Request Timing Register k (k = A, B) 733

21.2.23 GTADTBRk : A/D Conversion Start Request Timing Buffer Register k (k = A, B) 733
21.2.24 GTADTDBRk : A/D Conversion Start Request Timing Double-Buffer Register k (k = A, B) 734
21.2.25 GTDTCR : General PWM Timer Dead Time Control Register 734
21.2.26 GTDVU : General PWM Timer Dead Time Value Register U 735
21.2.27 GTADSMR : General PWM Timer A/D Conversion Start Request Signal Monitoring Register 735
21.2.28 GTICLF : General PWM Timer Inter Channel Logical Operation Function Setting Register 737
21.2.29 GTPC : General PWM Timer Period Count Register 738
21.2.30 GTSECSR : General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register 739
21.2.31 GTSECR : General PWM Timer Operation Enable Bit Simultaneous Control Register 741
21.2.32 OPSCR : Output Phase Switching Control Register 743
21.3 Operation 745
21.3.1 Basic Operation 745
21.3.2 Buffer Operation 753
21.3.3 PWM Output Operating Mode 763
21.3.4 Automatic Dead Time Setting Function 773
21.3.5 Count Direction Changing Function 777
21.3.6 Function of Output Duty 0% and 100% 778
21.3.7 Hardware Count Start/Count Stop and Clear Operation 779
21.3.8 Synchronized Operation 784
21.3.9 PWM Output Operation Examples 789
21.3.10 Period Count Function 794
21.3.11 Phase Counting Function 795
21.3.12 Output Phase Switching (GPT_OPS) 805
21.3.13 Inter Channel Logical Operation Function 812
21.4 Interrupt Sources 814
21.4.1 Interrupt Sources 814
21.4.2 DMAC and DTC Activation 816
21.5 A/D Conversion Start Request 816
21.6 Operations Linked by ELC 819
21.6.1 Event Signal Output to ELC 819
21.6.2 Event Signal Inputs from ELC 819
21.7 Noise Filter Function 819
21.8 Protection Function 820
21.8.1 Write-Protection for Registers 820
21.8.2 Disabling of Buffer Operation 820
21.8.3 GTIOCnm Pin Output Negate Control (n = 0 to 13, m = A, B) 822
21.9 Initialization Method of Output Pins 823
21.9.1 Pin Settings after Reset 823
21.9.2 Pin Initialization Due to Error During Operation 824
21.10 Usage Notes 824
21.10.1 Module-Stop Function Setting 824
21.10.2 GTCCRn Settings during Compare Match Operation (n = A to F) 824
21.10.3 Setting Range for GTCNT Counter 825
21.10.4 Starting and Stopping the GTCNT Counter 825
21.10.5 Priority Order of Each Event 825
21.10.6 Prohibit Invalid Register Setting 826
22. Low Power Asynchronous General Purpose Timer (AGT) 827

22.1 Overview ... 827
22.2 Register Descriptions ... 829
22.2.1 AGT : AGT Counter Register ... 829
22.2.2 AGTCMA : AGT Compare Match A Register ... 829
22.2.3 AGTCMB : AGT Compare Match B Register ... 830
22.2.4 AGTCR : AGT Control Register ... 830
22.2.5 AGTMR1 : AGT Mode Register 1 ... 832
22.2.6 AGTMR2 : AGT Mode Register 2 ... 833
22.2.7 AGTIOC : AGT I/O Control Register ... 834
22.2.8 AGTISR : AGT Event Pin Select Register ... 835
22.2.9 AGTCMSR : AGT Compare Match Function Select Register ... 836
22.2.10 AGTIOSEL : AGT Pin Select Register ... 836
22.3 Operation ... 837
22.3.1 Reload Register and Counter Rewrite Operation ... 837
22.3.2 Reload Register and AGT Compare Match A/B Register Rewrite Operation ... 838
22.3.3 Timer Mode ... 839
22.3.4 Pulse Output Mode ... 840
22.3.5 Event Counter Mode ... 841
22.3.6 Pulse Width Measurement Mode ... 842
22.3.7 Pulse Period Measurement Mode ... 843
22.3.8 Compare Match function ... 844
22.3.9 Output Settings for Each Mode ... 845
22.3.10 Standby Mode ... 846
22.3.11 Interrupt Sources ... 847
22.3.12 Event Signal Output to ELC ... 847
22.4 Usage Notes ... 848
22.4.1 Count Operation Start and Stop Control ... 848
22.4.2 Access to Counter Register ... 848
22.4.3 When Changing Mode ... 848
22.4.4 Output pin setting ... 848
22.4.5 Digital Filter ... 848
22.4.6 How to Calculate Event Number, Pulse Width, and Pulse Period ... 848
22.4.7 When Count is Forcibly Stopped by TSTOP Bit ... 849
22.4.8 When Selecting AGT0 Underflow as the Count Source ... 849
22.4.9 Module-stop function ... 849
22.4.10 When Switching Source Clock ... 849
23. Ultra-Low-Power Timer (ULPT) ... 850
23.1 Overview ... 850
23.2 Configuration ... 850
23.3 Register Descriptions ... 851
23.3.1 ULPTCNT : ULPT Counter Register ... 851
23.3.2 ULPTCMA : ULPT Compare Match A Register ... 852
23.3.3 ULPTCMB : ULPT Compare Match B Register ... 852
23.3.4 ULPTCR : ULPT Control Register ... 853
23.3.5 ULPTMR1 : ULPT Mode Register 1 ... 855
23.3.6 ULPTMR2 : ULPT Mode Register 2 ... 855
23.3.7 ULPTMR3 : ULPT Mode Register 3 ... 856
23.3.8 ULPTIOC : ULPT I/O Control Register ... 858
23.3.9 ULPTISR : ULPT Event Pin Select Register ... 858
23.3.10 ULPTCMSR : ULPT Compare Match Function Select Register ... 859

23.4 Operation 860
23.4.1 Count operation 860
23.4.2 Rewriting the counter and reload register 870
23.4.3 Rewriting the compare circuits and reload registers for compare match A/B 873
23.4.4 Pulse output 874
23.4.5 Compare match function 875
23.4.6 Input and output settings for each mode 876
23.4.7 Standby mode 877
23.4.8 Interrupt sources 878
23.4.9 Event output to the ELC 879
23.5 Usage notes 879
23.5.1 Start and stop control of the counter 879
23.5.2 Access to flags (TUNDF, TCMAF, and TCMBF bits in ULPTCR register) 880
23.5.3 Access to the ULPTCNT, ULPTCMA, and ULPTCMB registers 880
23.5.4 Mode changes 880
23.5.5 Setting the ULPTOn, ULPTOAn, and ULPTOBn pins 880
23.5.6 Calculating the number of events 880
23.5.7 When the counter is stopped forcibly by the TSTOP bit 880
23.5.8 Digital filter 880
23.5.9 Restrictions on Event Counting during Deep Software Standby Mode 1 880
23.5.10 Module-stop function 881
23.5.11 Setting the ULPTEEn and ULPTEVIn pins 881
24. Realtime Clock (RTC) 882
24.1 Overview 882
24.2 Register Descriptions 883
24.2.1 R64CNT : 64-Hz Counter 884
24.2.2 RSECCNT : Second Counter (in Calendar Count Mode) 884
24.2.3 RMINCNT : Minute Counter (in Calendar Count Mode) 885
24.2.4 RHRCNT : Hour Counter (in Calendar Count Mode) 885
24.2.5 RWKCNT : Day-of-Week Counter (in Calendar Count Mode) 886
24.2.6 BCNTn : Binary Counter n (n = 0 to 3) (in Binary Count Mode) 886
24.2.7 RDAYCNT : Day Counter 887
24.2.8 RMONCNT : Month Counter 887
24.2.9 RYRCNT : Year Counter 888
24.2.10 RSECAR : Second Alarm Register (in Calendar Count Mode) 888
24.2.11 RMINAR : Minute Alarm Register (in Calendar Count Mode) 889
24.2.12 RHRAR : Hour Alarm Register (in Calendar Count Mode) 889
24.2.13 RWKAR : Day-of-Week Alarm Register (in Calendar Count Mode) 890
24.2.14 BCNTnAR : Binary Counter n Alarm Register (n = 0 to 3) (in Binary Count Mode) 891
24.2.15 RDAYAR : Date Alarm Register (in Calendar Count Mode) 891
24.2.16 RMONAR : Month Alarm Register (in Calendar Count Mode) 892
24.2.17 RYRAR : Year Alarm Register (in Calendar Count Mode) 893
24.2.18 RYRAREN : Year Alarm Enable Register (in Calendar Count Mode) 893
24.2.19 BCNTnAER : Binary Counter n Alarm Enable Register (n = 0 to 3) (in Binary Count Mode) 894
24.2.20 RCR1 : RTC Control Register 1 894
24.2.21 RCR2 : RTC Control Register 2 (in Calendar Count Mode) 895
24.2.22 RCR2 : RTC Control Register 2 (in Binary Count Mode) 897
24.2.23 RCR4 : RTC Control Register 4 899
24.2.24 RFRL : Frequency Register L 899

24.2.25 RFRH : Frequency Register H 900
24.2.26 RADJ : Time Error Adjustment Register 900
24.2.27 RTCCRn : Time Capture Control Register n (n = 0 to 2) 901
24.2.28 RSECCPn : Second Capture Register n (n = 0 to 2) (in Calendar Count Mode) 902
24.2.29 RMINCPn : Minute Capture Register n (n = 0 to 2) (in Calendar Count Mode) 902
24.2.30 RHRCPn : Hour Capture Register n (n = 0 to 2) (in Calendar Count Mode) 903
24.2.31 RDAYCPn : Date Capture Register n (n = 0 to 2) (in Calendar Count Mode) 903
24.2.32 RMONCPn : Month Capture Register n (n = 0 to 2) (in Calendar Count Mode) 904
24.2.33 BCNTnCPm : BCNTn Capture Register m (n = 0 to 3, m = 0 to 2) (in Binary Count Mode) 904
24.3 Operation 905
24.3.1 Outline of Initial Settings of Registers after Power On 905
24.3.2 Clock and Count Mode Setting Procedure 905
24.3.3 Setting the Time 906
24.3.4 30-Second Adjustment 907
24.3.5 Reading 64-Hz Counter and Time 908
24.3.6 Alarm Function 910
24.3.7 Procedure for Disabling Alarm Interrupt 911
24.3.8 Time Error Adjustment Function 911
24.3.9 Capturing the time 913
24.4 Interrupt Sources 914
24.5 Event Link Output 915
24.5.1 Interrupt Handling and Event Linking 916
24.6 Usage Notes 916
24.6.1 Register Writing during Counting 916
24.6.2 Use of Periodic Interrupts 916
24.6.3 RTCOUT (1-Hz/64-Hz) Clock Output 917
24.6.4 Transitions to Low Power Modes after Setting Registers 917
24.6.5 Notes on Writing to and Reading from Registers 917
24.6.6 Changing the Count Mode 917
24.6.7 Initialization Procedure When the RTC Is Not to Be Used 917
24.6.8 When Switching Source Clock 918
24.6.9 When writing to the RTC registers 918
25. Watchdog Timer (WDT) 919
25.1 Overview 919
25.2 Register Descriptions 920
25.2.1 WDTRR : WDT Refresh Register 920
25.2.2 WDTCR : WDT Control Register 921
25.2.3 WDTSR : WDT Status Register 923
25.2.4 WDTRCR : WDT Reset Control Register 924
25.2.5 WDTCSTPR : WDT Count Stop Control Register 925
25.2.6 Option Function Select Register 0 (OFS0) 925
25.3 Operation 925
25.3.1 Count Operation in each Start Mode 925
25.3.2 Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers 929
25.3.3 Refresh Operation 929
25.3.4 Status Flags 930
25.3.5 Reset Output 930
25.3.6 Interrupt Sources 930
25.3.7 Reading the Down-Counter Value 931

25.3.8 Association between Option Function Select Register 0 (OFS0) and WDT Registers 931
25.4 Output to the Event Link Controller (ELC) 932
25.5 Usage Notes 932
25.5.1 ICU Event Link Setting Register n (IELSRn) Setting 932
26. Independent Watchdog Timer (IWDT) 933
26.1 Overview 933
26.1.1 Block Diagram 933
26.2 Register specifications 934
26.2.1 IWDTRR : IWDT Refresh Register 934
26.2.2 IWDTCR : IWDT Control Register 935
26.2.3 IWDTSR : IWDT Status Register 937
26.2.4 IWDTRCR : IWDT Reset Control Register 938
26.2.5 IWDTCSTPR : IWDT Count Stop Control Register 939
26.3 Operation 939
26.3.1 Count operation in each start mode 939
26.3.2 Controlling writes to the IWDTCR, IWDTRCR, and IWDTCSTPR registers 942
26.3.3 Refresh operation 942
26.3.4 Status flags 945
26.3.5 Reset output 945
26.3.6 Interrupt sources 945
26.3.7 Reading the down-counter value 945
26.3.8 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers 947
26.4 Link operation by the event link function 947
27. Ethernet MAC Controller (ETHERC) 948
27.1 Overview 948
27.2 Register Descriptions 953
27.2.1 ECMR : ETHERC Mode Register 953
27.2.2 RFLR : Receive Frame Maximum Length Register 955
27.2.3 ECSR : ETHERC Status Register 955
27.2.4 ECSIPR : ETHERC Interrupt Enable Register 956
27.2.5 PIR : PHY Interface Register 957
27.2.6 PSR : PHY Status Register 958
27.2.7 RDMLR : Random Number Generation Counter Upper Limit Setting Register 958
27.2.8 IPGR : Interpacket Gap Register 958
27.2.9 APR : Automatic PAUSE Frame Register 959
27.2.10 MPR : Manual PAUSE Frame Register 959
27.2.11 RFCF : Received PAUSE Frame Counter 960
27.2.12 TPAUSER : PAUSE Frame Retransmit Count Setting Register 960
27.2.13 TPAUSECR : PAUSE Frame Retransmit Counter 961
27.2.14 BCFRR : Broadcast Frame Receive Count Setting Register 961
27.2.15 MAHR : MAC Address Upper Bit Register 961
27.2.16 MALR : MAC Address Lower Bit Register 962
27.2.17 TROCR : Transmit Retry Over Counter Register 962
27.2.18 CDCR : Late Collision Detect Counter Register 963
27.2.19 LCCR : Lost Carrier Counter Register 963
27.2.20 CNDCR : Carrier Not Detect Counter Register 963
27.2.21 CEFCR : CRC Error Frame Receive Counter Register 964
27.2.22 FRECR : Frame Receive Error Counter Register 964
27.2.23 TSFRCR : Too-Short Frame Receive Counter Register 964

27.2.24 TLFRCR : Too-Long Frame Receive Counter Register 965
27.2.25 RFCR : Received Alignment Error Frame Counter Register 965
27.2.26 MAFCR : Multicast Address Frame Receive Counter Register 966
27.3 Operation 966
27.3.1 Transmission 966
27.3.2 Reception 968
27.3.3 Frame Timing 969
27.3.4 Accessing the MII and RMII Registers 971
27.3.5 Magic Packet Detection 973
27.3.6 Adjusting Transmission Efficiency by Changing the IPG 973
27.3.7 Flow Control 974
27.4 Interrupts 976
27.5 Usage Notes 976
27.5.1 Preventing the LCHNG Flag from Erroneously Setting to 1 976
27.5.2 Input to RMII0_RX_ER Pin While RMII Is Selected 976
27.5.3 Collision Occurrence in Half-Duplex Mode 976
28. Ethernet DMA Controller (EDMAC) 977
28.1 Overview 977
28.2 Register Descriptions 978
28.2.1 EDMR : EDMAC Mode Register 978
28.2.2 EDTRR : EDMAC Transmit Request Register 979
28.2.3 EDRRR : EDMAC Receive Request Register 980
28.2.4 TDLAR : Transmit Descriptor List Start Address Register 980
28.2.5 RDLAR : Receive Descriptor List Start Address Register 981
28.2.6 EESR : ETHERC/EDMAC Status Register 981
28.2.7 EESIPR : ETHERC/EDMAC Status Interrupt Enable Register 984
28.2.8 TRSCER : ETHERC/EDMAC Transmit/Receive Status Copy Enable Register 986
28.2.9 RMFCR : Missed-Frame Counter Register 987
28.2.10 TFTR : Transmit FIFO Threshold Register 987
28.2.11 FDR : FIFO Depth Register 988
28.2.12 RMCR : Receive Method Control Register 988
28.2.13 TFUCR : Transmit FIFO Underflow Counter 989
28.2.14 RFOCR : Receive FIFO Overflow Counter 989
28.2.15 IOSR : Independent Output Signal Setting Register 990
28.2.16 FCFTR : Flow Control Start FIFO Threshold Setting Register 990
28.2.17 RPADIR : Receive Data Padding Insert Register 991
28.2.18 TRIMD : Transmit Interrupt Setting Register 991
28.2.19 RBWAR : Receive Buffer Write Address Register 992
28.2.20 RDFAR : Receive Descriptor Fetch Address Register 992
28.2.21 TBRAR : Transmit Buffer Read Address Register 993
28.2.22 TDFAR : Transmit Descriptor Fetch Address Register 993
28.3 Operation 993
28.3.1 Descriptor Lists and Data Buffers 993
28.3.2 Transmission 997
28.3.3 Reception 998
28.3.4 Multi-Buffer Frame Transmission 999
28.3.5 Bus Transfer Error 1001
28.4 Interrupts 1003
28.5 Usage Notes 1003
28.5.1 Settings for the Module-Stop Function 1003

28.5.2 Stopping the EDMAC during Operation 1004
29. USB 2.0 Full-Speed Module (USBFS) 1005
29.1 Overview 1005
29.2 Register Descriptions 1007
29.2.1 SYSCFG : System Configuration Control Register 1007
29.2.2 SYSSTS0 : System Configuration Status Register 0 1008
29.2.3 DVSTCTR0 : Device State Control Register 0 1009
29.2.4 CFIFO/CFIFOL : CFIFO Port Register 1012
29.2.5 DnFIFO/DnFIFOL : DnFIFO Port Register (n = 0, 1) 1013
29.2.6 CFIFOSEL : CFIFO Port Select Register 1014
29.2.7 DnFIFOSEL : DnFIFO Port Select Register (n = 0, 1) 1016
29.2.8 CFIFOCTR : CFIFO Port Control Register 1017
29.2.9 DnFIFOCTR : DnFIFO Port Control Register (n = 0, 1) 1019
29.2.10 INTENB0 : Interrupt Enable Register 0 1020
29.2.11 INTENB1 : Interrupt Enable Register 1 1021
29.2.12 BRDYENB : BRDY Interrupt Enable Register 1022
29.2.13 NRDYENB : NRDY Interrupt Enable Register 1023
29.2.14 BEMPENB : BEMP Interrupt Enable Register 1024
29.2.15 SOFCFG : SOF Output Configuration Register 1025
29.2.16 INTSTS0 : Interrupt Status Register 0 1026
29.2.17 INTSTS1 : Interrupt Status Register 1 1028
29.2.18 BRDYSTS : BRDY Interrupt Status Register 1031
29.2.19 NRDYSTS : NRDY Interrupt Status Register 1032
29.2.20 BEMPSTS : BEMP Interrupt Status Register 1032
29.2.21 FRMNUM : Frame Number Register 1033
29.2.22 DVCHGR : Device State Change Register 1034
29.2.23 USBADDR : USB Address Register 1034
29.2.24 USBREQ : USB Request Type Register 1035
29.2.25 USBVAL : USB Request Value Register 1036
29.2.26 USBINDX : USB Request Index Register 1037
29.2.27 USBLENG : USB Request Length Register 1037
29.2.28 DCPCFG : DCP Configuration Register 1038
29.2.29 DCPMAXP : DCP Maximum Packet Size Register 1039
29.2.30 DCPCTR : DCP Control Register 1039
29.2.31 PIPESEL : Pipe Window Select Register 1042
29.2.32 PIPECFG : Pipe Configuration Register 1043
29.2.33 PIPEMAXP : Pipe Maximum Packet Size Register 1045
29.2.34 PIPEPERI : Pipe Cycle Control Register 1046
29.2.35 PIPEnCTR : PIPEn Control Registers (n = 1 to 5) 1047
29.2.36 PIPEnCTR : PIPEn Control Registers (n = 6 to 9) 1051
29.2.37 PIPEnTRE : PIPEn Transaction Counter Enable Register (n = 1 to 5) 1053
29.2.38 PIPEnTRN : PIPEn Transaction Counter Register (n = 1 to 5) 1054
29.2.39 DEVADDn : Device Address n Configuration Register (n = 0 to 5) 1055
29.2.40 DPUSR0R : Deep Software Standby USB Transceiver Control/Pin Monitor Register 1055
29.2.41 DPUSR1R : Deep Software Standby USB Suspend/Resume Interrupt Register 1057
29.3 Operation 1058
29.3.1 System Control 1058
29.3.2 Interrupts 1067
29.3.3 Interrupt Descriptions 1071
29.3.4 Pipe Control 1080

29.3.5 FIFO Buffer ... 1084
29.3.6 FIFO Buffer Clearing ... 1084
29.3.7 FIFO Port Functions ... 1085
29.3.8 DMA Transfers (D0FIFO and D1FIFO Ports) ... 1086
29.3.9 Control Transfers Using the DCP ... 1086
29.3.10 Bulk Transfers (Pipes 1 to 5) ... 1088
29.3.11 Interrupt Transfers (Pipes 6 to 9) ... 1088
29.3.12 Isochronous Transfers (Pipes 1 and 2) ... 1089
29.3.13 SOF Interpolation Function ... 1096
29.3.14 Pipe Schedule ... 1096
29.4 Usage Notes ... 1097
29.4.1 Settings for the Module-Stop State ... 1097
29.4.2 Clearing the Interrupt Status Register on Canceling Software Standby Mode ... 1098
29.4.3 Clearing the Interrupt Status Register after Setting Up the Port Function ... 1098
29.4.4 Setting Up the Port Function before setting USB function ... 1098
30. USB 2.0 High-Speed Module (USBHS) ... 1099
30.1 Overview ... 1099
30.2 Register Descriptions ... 1101
30.2.1 SYSCFG : System Configuration Control Register ... 1101
30.2.2 BUSWAIT : CPU Bus Wait Register ... 1103
30.2.3 SYSSTS0 : System Configuration Status Register ... 1103
30.2.4 PLLSTA : PLL Status Register ... 1105
30.2.5 DVSTCTR0 : Device State Control Register 0 ... 1105
30.2.6 TESTMODE : USB Test Mode Register ... 1108
30.2.7 CFIFO, DnFIFO : FIFO Port Register (n = 0, 1) ... 1109
30.2.8 CFIFOSEL : CFIFO Port Selection Register ... 1111
30.2.9 DnFIFOSEL : DnFIFO Port Selection Register (n = 0, 1) ... 1112
30.2.10 CFIFOCTR, DnFIFOCTR : FIFO Port Control Register (n = 0, 1) ... 1114
30.2.11 INTENB0 : Interrupt Enable Register 0 ... 1116
30.2.12 INTENB1 : Interrupt Enable Register 1 ... 1117
30.2.13 BRDYENB : BRDY Interrupt Enable Register ... 1118
30.2.14 NRDYENB : NRDY Interrupt Enable Register ... 1118
30.2.15 BEMPENB : BEMP Interrupt Enable Register ... 1119
30.2.16 SOFCFG : SOF Output Configuration Register ... 1119
30.2.17 PHYSET : PHY Setting Register ... 1120
30.2.18 INTSTS0 : Interrupt Status Register 0 ... 1121
30.2.19 INTSTS1 : Interrupt Status Register 1 ... 1123
30.2.20 BRDYSTS : BRDY Interrupt Status Register ... 1126
30.2.21 NRDYSTS : NRDY Interrupt Status Register ... 1127
30.2.22 BEMPSTS : BEMP Interrupt Status Register ... 1127
30.2.23 FRMNUM : Frame Number Register ... 1128
30.2.24 UFRMNUM : µFrame Number Register ... 1128
30.2.25 USBADDR : USB Address Register ... 1129
30.2.26 USBREQ : USB Request Type Register ... 1130
30.2.27 USBVAL : USB Request Value Register ... 1130
30.2.28 USBINDX : USB Request Index Register ... 1131
30.2.29 USBLENG : USB Request Length Register ... 1131
30.2.30 DCPCFG : DCP Configuration Register ... 1132
30.2.31 DCPMAXP : DCP Maximum Packet Size Register ... 1132
30.2.32 DCPCTR : DCP Control Register ... 1133

30.2.33 PIPESEL : Pipe Window Select Register ... 1137
30.2.34 PIPECFG : Pipe Configuration Register ... 1137
30.2.35 PIPEBUF : Pipe Buffer Register ... 1140
30.2.36 PIPEMAXP : Pipe Maximum Packet Size Register ... 1141
30.2.37 PIPEPERI : Pipe Cycle Control Register ... 1142
30.2.38 PIPEnCTR : Pipe n Control Register (n = 1 to 9) ... 1143
30.2.39 PIPEnTRE : Pipe n Transaction Counter Enable Register (n = 1 to 5) ... 1147
30.2.40 PIPEnTRN : Pipe n Transaction Counter Register (n = 1 to 5) ... 1148
30.2.41 DEVADDn : Device Address n Configuration Register (n = 0 to 9, A) ... 1149
30.2.42 LPCTRL : Low Power Control Register ... 1150
30.2.43 LPSTS : Low Power Status Register ... 1151
30.2.44 BCCTRL : Battery Charging Control Register ... 1152
30.2.45 PL1CTRL1 : Function L1 Control Register 1 ... 1153
30.2.46 PL1CTRL2 : Function L1 Control Register 2 ... 1154
30.2.47 HL1CTRL1 : Host L1 Control Register 1 ... 1155
30.2.48 HL1CTRL2 : Host L1 Control Register 2 ... 1156
30.2.49 DPUSR0R : Deep Software Standby USB Transceiver Control/Pin Monitor Register ... 1157
30.2.50 DPUSR1R : Deep Software Standby USB Suspend/Resume Interrupt Register ... 1158
30.2.51 DPUSR2R : Deep Software Standby USB Suspend/Resume Interrupt Register ... 1158
30.2.52 DPUSRCR : Deep Software Standby USB Suspend/Resume Command Register ... 1159
30.3 Operation ... 1159
30.3.1 System Control ... 1159
30.3.2 Controlling the USB data bus using resistors ... 1160
30.3.3 Supplying the Clock ... 1160
30.3.4 Constraints on Stopping the Clock ... 1161
30.3.5 Interrupts ... 1162
30.3.6 Interrupt Descriptions ... 1166
30.3.7 Pipe Control ... 1176
30.3.8 FIFO Buffer ... 1181
30.3.9 Control Transfers Using the DCP ... 1185
30.3.10 Bulk Transfers (Pipes 1 to 5) ... 1187
30.3.11 Interrupt Transfers (Pipes 6 to 9) ... 1188
30.3.12 Isochronous Transfers (Pipes 1 and 2) ... 1188
30.3.13 SOF Complementation Function ... 1195
30.3.14 Pipe Schedule ... 1196
30.3.15 Battery charging detection processing ... 1197
30.3.16 Link Power Management Processing ... 1201
30.3.17 Deep Software Standby Mode 1 Because of USB Suspend/Resume Interrupts ... 1203
30.3.18 Example External Connection Circuits ... 1208
30.4 Usage Notes ... 1212
30.4.1 Settings for the Module-Stop Function ... 1212
30.4.2 Setup for Transitioning to Deep Software Standby Mode 1 ... 1212
30.4.3 Clearing the Interrupt Status Register on Exiting Software Standby Mode ... 1212
30.4.4 Clearing the Interrupt Status Register after Setting Up the Port Function ... 1213
30.4.5 Restriction of register access ... 1213
31. Serial Communications Interface (SCI) ... 1214
31.1 Overview ... 1214
31.2 Register Descriptions ... 1219
31.2.1 RSR : Receive Shift Register ... 1219
31.2.2 RDR/RDR_BY : Receive Data Register ... 1220

31.2.3 TDR/TDRLL/TDRLH : Transmit Data Register 1221
31.2.4 TSR : Transmit Shift Register 1222
31.2.5 CCR0 : Common Control Register 0 1222
31.2.6 CCR1 : Common Control Register 1 1225
31.2.7 CCR2 : Common Control Register 2 1229
31.2.8 CCR3 : Common Control Register 3 1243
31.2.9 CCR4 : Common Control Register 4 1249
31.2.10 ICR : Simple IIC Control Register 1252
31.2.11 FCR : FIFO Control Register 1254
31.2.12 MCR : Manchester Control Register 1256
31.2.13 DCR : Driver Control Register 1259
31.2.14 XCR0 : Simple LIN Control Register 0 1260
31.2.15 XCR1 : Simple LIN Control Register 1 1262
31.2.16 XCR2 : Simple LIN Control Register 2 1263
31.2.17 CSR : Common Status Register 1264
31.2.18 ISR : Simple IIC Status Register 1270
31.2.19 FRSR : FIFO Receive Status Register 1271
31.2.20 FTSR : FIFO Transmit Status Register 1272
31.2.21 MSR : Manchester Status Register 1272
31.2.22 XSR0 : Simple LIN Status Register 0 1275
31.2.23 XSR1 : Simple LIN Status Register 1 1277
31.2.24 CFCLR : Common Flag Clear Register 1277
31.2.25 ICFCLR : Simple IIC Flag Clear Register 1278
31.2.26 FFCLR : FIFO Flag Clear Register 1279
31.2.27 MFCLR : Manchester Flag Clear Register 1279
31.2.28 XFCLR : Simple LIN Flag Clear Register 1280
31.2.29 CESR : Communication Enable Status Register 1280
31.3 Operation in Asynchronous Mode 1281
31.3.1 Serial Data Transfer Format 1281
31.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode 1282
31.3.3 Clock 1283
31.3.4 Double-Speed Operation and Frequency of 6 or 4 Times the Bit Rate 1284
31.3.5 CTS and RTS Functions 1284
31.3.6 Address Match (Receive Data Match Detection) Function 1285
31.3.7 SCI Initialization in Asynchronous Mode 1287
31.3.8 Serial Data Transmission in Asynchronous Mode 1289
31.3.9 Serial Data Reception in Asynchronous Mode 1295
31.3.10 The function of adjust receive sampling timing (Asynchronous Mode) 1302
31.3.11 The function of adjust transmit timing (Asynchronous Mode) 1305
31.4 Multi-Processor Communication Function 1310
31.4.1 Multi-Processor Serial Data Transmission 1311
31.4.2 Multi-Processor Serial Data Reception 1314
31.5 Operation in Manchester mode 1320
31.5.1 Frame Format 1321
31.5.2 Clock 1325
31.5.3 Initialization of the SCI in Manchester Mode 1325
31.5.4 Double-speed operation 1326
31.5.5 CTS and RTS functions 1326
31.5.6 Serial data transmission in Manchester mode 1327
31.5.7 Serial Data Reception in Manchester Mode 1330

31.5.8 Operation When Multi-Processor Bit Is Used ... 1334
31.5.9 Receive Retiming ... 1334
31.5.10 Polarity Setting for Manchester Code ... 1335
31.5.11 Errors in Manchester Mode ... 1336
31.6 Operation in Clock Synchronous Mode ... 1341
31.6.1 Clock ... 1341
31.6.2 CTS and RTS Functions ... 1341
31.6.3 SCI Initialization in Clock Synchronous Mode ... 1342
31.6.4 Serial Data Transmission in Clock Synchronous Mode ... 1343
31.6.5 Serial Data Reception in Clock Synchronous Mode ... 1347
31.6.6 Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode ... 1351
31.6.7 Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with internal clock used ... 1353
31.7 Operation in Smart Card Interface Mode ... 1354
31.7.1 Example Connection ... 1354
31.7.2 Data Format (Except in Block Transfer Mode) ... 1355
31.7.3 Block Transfer Mode ... 1356
31.7.4 Receive Data Sampling Timing and Reception Margin ... 1356
31.7.5 SCI Initialization (Smart Card Interface Mode) ... 1357
31.7.6 Serial Data Transmission (Except in Block Transfer Mode) ... 1358
31.7.7 Serial Data Reception (Except in Block Transfer Mode) ... 1361
31.7.8 Clock Output Control ... 1363
31.8 Operation in Simple IIC Mode ... 1363
31.8.1 Generation of Start, Restart, and Stop Conditions ... 1365
31.8.2 Clock Synchronization ... 1366
31.8.3 SDAn Output Delay ... 1366
31.8.4 SCI Initialization in Simple IIC Mode ... 1367
31.8.5 Operation in Master Transmission in Simple IIC Mode ... 1368
31.8.6 Master Reception in Simple IIC Mode ... 1371
31.9 Operation in Simple SPI Mode ... 1373
31.9.1 States of Pins in Master and Slave Modes ... 1374
31.9.2 SS Function in Master Mode ... 1375
31.9.3 SS Function in Slave Mode ... 1375
31.9.4 Relationship between Clock and Transmit/Receive Data ... 1375
31.9.5 SCI Initialization in Simple SPI Mode ... 1376
31.9.6 Transmission and Reception of Serial Data in Simple SPI Mode ... 1376
31.9.7 Reception Sampling Timing Adjustment Function in Simple SPI Mode with internal clock used ... 1377
31.10 Bit Rate Modulation Function ... 1377
31.11 Simple LIN mode ... 1377
31.11.1 Simple LIN Start Frame Transmission ... 1378
31.11.2 Simple LIN Start Frame Reception ... 1380
31.11.3 Simple LIN Bus Conflict Detection Function ... 1385
31.11.4 Simple LIN Bit Rate Measurement Function ... 1386
31.12 Interrupt Sources ... 1387
31.12.1 Buffer Operation for SCIn_TXI and SCIn_RXI Interrupts ... 1387
31.12.2 Interrupts in Asynchronous, Manchester, Clock Synchronous, and Simple SPI Modes ... 1388
31.12.3 Interrupts in Smart Card Interface Mode ... 1389
31.12.4 Interrupts in Simple IIC Mode ... 1390
31.12.5 Interrupts in Simple LIN mode ... 1390

31.13 Event Linking 1391
31.14 Noise Cancellation Function 1393
31.15 RS-485 Driver Control Function 1394
31.16 Loopback Function 1394
31.17 Half-Duplex communication Function 1395
31.18 Synchronizer Bypass Function 1395
31.19 Usage Notes 1396
31.19.1 Settings for the Module-Stop Function 1396
31.19.2 SCI Operation during Low Power State 1396
31.19.3 Break Detection and Processing 1399
31.19.4 Mark State and Production of Breaks 1400
31.19.5 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode) 1400
31.19.6 Writing Data to TDR 1400
31.19.7 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode) 1400
31.19.8 Restrictions on Using DMAC or DTC 1401
31.19.9 Notes on Starting Transfer 1401
31.19.10 Limitations on Simple SPI Mode 1402
31.19.11 Notes on Transmit Enable bit (CCR0.TE) 1402
31.19.12 Notes on Simple LIN mode 1403
31.19.13 Notes on RS-485 Driver Control function 1403
31.19.14 Notes on Loopback function 1403
31.19.15 Notes regarding register access when operation clock (TCLK) is slower than bus clock (PCLK) 1403
31.19.16 Notes on interrupting operation 1403
31.19.17 Notes on CCR3.BPEN bit setting 1403
32. I^2C Bus Interface (IIC) 1404
32.1 Overview 1404
32.2 Register Descriptions 1406
32.2.1 ICCR1 : I^2C Bus Control Register 1 1406
32.2.2 ICCR2 : I^2C Bus Control Register 2 1408
32.2.3 ICMR1 : I^2C Bus Mode Register 1 1411
32.2.4 ICMR2 : I^2C Bus Mode Register 2 1412
32.2.5 ICMR3 : I^2C Bus Mode Register 3 1413
32.2.6 ICFER : I^2C Bus Function Enable Register 1415
32.2.7 ICSER : I^2C Bus Status Enable Register 1417
32.2.8 ICIER : I^2C Bus Interrupt Enable Register 1418
32.2.9 ICSR1 : I^2C Bus Status Register 1 1419
32.2.10 ICSR2 : I^2C Bus Status Register 2 1421
32.2.11 ICWUR : I^2C Bus Wakeup Unit Register 1425
32.2.12 ICWUR2 : I^2C Bus Wakeup Unit Register 2 1426
32.2.13 SARLy : Slave Address Register Ly (y = 0 to 2) 1427
32.2.14 SARUy : Slave Address Register Uy (y = 0 to 2) 1428
32.2.15 ICBRL : I^2C Bus Bit Rate Low-Level Register 1428
32.2.16 ICBRH : I^2C Bus Bit Rate High-Level Register 1429
32.2.17 ICDRT : I^2C Bus Transmit Data Register 1430
32.2.18 ICDRR : I^2C Bus Receive Data Register 1430

32.2.19 ICDRS : I2C Bus Shift Register 1431
32.3 Operation 1431
32.3.1 Communication Data Format 1431
32.3.2 Initial Settings 1432
32.3.3 Master Transmit Operation 1433
32.3.4 Master Receive Operation 1437
32.3.5 Slave Transmit Operation 1442
32.3.6 Slave Receive Operation 1445
32.4 SCL Synchronization Circuit 1447
32.5 SDA Output Delay Function 1448
32.6 Digital Noise Filter Circuits 1448
32.7 Address Match Detection 1449
32.7.1 Slave-Address Match Detection 1449
32.7.2 Detection of General Call Address 1452
32.7.3 Device-ID Address Detection 1453
32.7.4 Host Address Detection 1454
32.8 Wakeup Function 1455
32.8.1 Normal Wakeup Mode 1 1456
32.8.2 Normal Wakeup Mode 2 1459
32.8.3 Command Recovery Mode and EEP Response Mode (Special Wakeup Modes) 1461
32.9 Automatic Low-Hold Function for SCL 1464
32.9.1 Function to Prevent Wrong Transmission of Transmit Data 1464
32.9.2 NACK Reception Transfer Suspension Function 1465
32.9.3 Function to Prevent Failure to Receive Data 1466
32.10 Arbitration-Lost Detection Functions 1467
32.10.1 Master Arbitration-Lost Detection (MALE Bit) 1467
32.10.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit) 1469
32.10.3 Slave Arbitration-Lost Detection (SALE Bit) 1470
32.11 Start, Restart, and Stop Condition Issuing Function 1471
32.11.1 Issuing a Start Condition 1471
32.11.2 Issuing a Restart Condition 1471
32.11.3 Issuing a Stop Condition 1473
32.12 Bus Hanging 1474
32.12.1 Timeout Function 1474
32.12.2 Extra SCL Clock Cycle Output Function 1475
32.12.3 IIC Reset and Internal Reset 1476
32.13 SMBus Operation 1476
32.13.1 SMBus Timeout Measurement 1476
32.13.2 Packet Error Code (PEC) 1478
32.13.3 SMBus Host Notification Protocol (Notify ARP Master Command) 1478
32.14 Interrupt Sources 1478
32.14.1 Buffer Operation for IICn_TXI and IICn_RXI Interrupts 1479
32.15 State of Registers When Issuing Each Condition 1479
32.16 Event Link Output 1480
32.16.1 Interrupt Handling and Event Linking 1480
32.17 Usage Notes 1480
32.17.1 Settings for the Module-Stop Function 1480
32.17.2 Notes on Starting Transfer 1480
33. I3C Bus Interface (I3C) 1482
33.1 Overview 1482

33.1.1 Functional Overview 1482
33.1.2 Block Diagram 1484
33.2 Registers 1485
33.2.1 PRTS : Protocol Selection Register 1485
33.2.2 BCTL : Bus Control Register 1486
33.2.3 MSDVAD : Master Device Address Register 1487
33.2.4 RSTCTL : Reset Control Register 1488
33.2.5 PRSST : Present State Register 1490
33.2.6 INST : Internal Status Register 1492
33.2.7 INSTE : Internal Status Enable Register 1493
33.2.8 INIE : Internal Interrupt Enable Register 1494
33.2.9 INSTFC : Internal Status Force Register 1494
33.2.10 DVCT : Device Characteristic Table Register 1495
33.2.11 IBINCTL : IBI Notify Control Register 1496
33.2.12 BFCTL : Bus Function Control Register 1496
33.2.13 SVCTL : Slave Control Register 1498
33.2.14 REFCKCTL : Reference Clock Control Register 1500
33.2.15 STDBR : Standard Bit Rate Register 1500
33.2.16 EXTBR : Extended Bit Rate Register 1503
33.2.17 BFRECDT : Bus Free Condition Detection Time Register 1504
33.2.18 BAVLCDT : Bus Available Condition Detection Time Register 1504
33.2.19 BIDLCDT : Bus Idle Condition Detection Time Register 1505
33.2.20 OUTCTL : Output Control Register 1505
33.2.21 INCTL : Input Control Register 1507
33.2.22 TMOCTL : Timeout Control Register 1508
33.2.23 WUCTL : Wake Up Unit Control Register 1509
33.2.24 ACKCTL : Acknowledge Control Register 1510
33.2.25 SCSTRCTL : SCL Stretch Control Register 1511
33.2.26 SCSTLCTL : SCL Stalling Control Register 1512
33.2.27 SVTDLG0 : Slave Transfer Data Length Register 0 1513
33.2.28 STCTL : Synchronous Timing Control Register 1514
33.2.29 ATCTL : Asynchronous Timing Control Register 1514
33.2.30 ATTRG : Asynchronous Timing Trigger Register 1515
33.2.31 ATCCNTE : Asynchronous Timing Control Counter Enable Register 1515
33.2.32 CNDCTL : Condition Control Register 1516
33.2.33 NCMDQP : Normal Command Queue Port Register 1517
33.2.34 NRSPQP : Normal Response Queue Port Register 1518
33.2.35 NTDTBP0/NTDTBP0_BY : Normal Transfer Data Buffer Port Register 0 1518
33.2.36 NIBIQP : Normal IBI Queue Port Register 1519
33.2.37 NRSQP : Normal Receive Status Queue Port Register 1520
33.2.38 HCMDQP : High Priority Command Queue Port Register 1520
33.2.39 HRSPQP : High Priority Response Queue Port Register 1520
33.2.40 HTDTBP : High Priority Transfer Data Buffer Port Register 1521
33.2.41 NQTHCTL : Normal Queue Threshold Control Register 1521
33.2.42 NTBTHCTL0 : Normal Transfer Data Buffer Threshold Control Register 0 1523
33.2.43 NRQTHCTL : Normal Receive Status Queue Threshold Control Register 1524
33.2.44 HQTHCTL : High Priority Queue Threshold Control Register 1525
33.2.45 HTBTHCTL : High Priority Transfer Data Buffer Threshold Control Register 1526
33.2.46 BST : Bus Status Register 1527
33.2.47 BSTE : Bus Status Enable Register 1531

33.2.48 BIE : Bus Interrupt Enable Register 1532
33.2.49 BSTFC : Bus Status Force Register 1534
33.2.50 NTST : Normal Transfer Status Register 1535
33.2.51 NTSTE : Normal Transfer Status Enable Register 1539
33.2.52 NTIE : Normal Transfer Interrupt Enable Register 1541
33.2.53 NTSTFC : Normal Transfer Status Force Register 1542
33.2.54 HTST : High Priority Transfer Status Register 1544
33.2.55 HTSTE : High Priority Transfer Status Enable Register 1546
33.2.56 HTIE : High Priority Transfer Interrupt Enable Register 1547
33.2.57 HTSTFC : High Priority Transfer Status Force Register 1548
33.2.58 BCST : Bus Condition Status Register 1549
33.2.59 SVST : Slave Status Register 1551
33.2.60 WUST : Wake Up Unit Operating Status Register 1553
33.2.61 MRCCPT : MsyncCNT Counter Capture Register 1555
33.2.62 DATBASm : Device Address Table Basic Register m (m = 0 to 7) 1555
33.2.63 EXDATBAS : Extended Device Address Table Basic Register 1556
33.2.64 SDATBASn : Slave Device Address Table Basic Register n (n = 0 to 2) 1557
33.2.65 MSDCTm : Master Device Characteristic Table Register m (m = 0 to 7) 1558
33.2.66 SVDCT : Slave Device Characteristic Table Register 1560
33.2.67 SDCTPIDL : Slave Device Characteristic Table Provisional ID Low Register 1561
33.2.68 SDCTPIDH : Slave Device Characteristic Table Provisional ID High Register 1562
33.2.69 SVDVADn : Slave Device Address Register n (n = 0 to 2) 1562
33.2.70 CSECMD : CCC Slave Events Command Register 1564
33.2.71 CEACTST : CCC Enter Activity State Register 1565
33.2.72 CMWLG : CCC Max Write Length Register 1565
33.2.73 CMRLG : CCC Max Read Length Register 1566
33.2.74 CETSTMD : CCC Enter Test Mode Register 1567
33.2.75 CGDVST : CCC Get Device Status Register 1568
33.2.76 CMDSPW : CCC Max Data Speed W (Write) Register 1569
33.2.77 CMDSPR : CCC Max Data Speed R (Read) Register 1569
33.2.78 CMDSPT : CCC Max Data Speed T (Turnaround) Register 1570
33.2.79 CETSM : CCC Exchange Timing Support Information M (Mode) Register 1570
33.2.80 CETSS : CCC Exchange Timing Support Information S (State) Register 1572
33.2.81 CGHDRCAP : CCC Get HDR Capability Register 1573
33.2.82 BITCNT : Bit Count Register 1574
33.2.83 NQSTLV : Normal Queue Status Level Register 1575
33.2.84 NDBSTLV0 : Normal Data Buffer Status Level Register 0 1576
33.2.85 NRSQSTLV : Normal Receive Status Queue Status Level Register 1576
33.2.86 HQSTLV : High Priority Queue Status Level Register 1577
33.2.87 HDBSTLV : High Priority Data Buffer Status Level Register 1577
33.2.88 PRSTDBG : Present State Debug Register 1578
33.2.89 MSERRCNT : Master Error Counters Register 1578
33.2.90 SC1CPT : SC1 Capture Monitor Register 1579
33.2.91 SC2CPT : SC2 Capture Monitor Register 1580
33.2.92 CECTL : Clock Enable Control Register 1580
33.3 Operation 1581
33.3.1 Data Structures 1581
33.3.2 Details of Function 1597
33.3.3 Operation 1698
33.4 Interrupt Sources 1718

33.4.1 Overview ... 1718
33.4.2 Buffer Operation for Buffer Full/Empty Interrupts ... 1720
33.5 Event Link Output ... 1720
33.5.1 Interrupt Handling and Event Linking ... 1720
33.6 Reset Descriptions ... 1721
33.7 Usage Notes ... 1736
33.7.1 Settings for the Operating Clock ... 1736
33.7.2 Settings for the SCL frequency Clock ... 1737
33.7.3 Module-stop function ... 1737
33.7.4 Notes on Rewriting the PRTS.PRTMD Bit ... 1737
34. CAN with Flexible Data-rate (CANFD) ... 1738
34.1 Overview ... 1738
34.1.1 CANFD Module ... 1738
34.1.2 Clock Restriction ... 1740
34.2 Register Descriptions ... 1740
34.2.1 Register Table ... 1740
34.2.2 CFDC0NCFG : Nominal Bitrate Configuration Register ... 1741
34.2.3 CFDC0CTR : Control Register ... 1742
34.2.4 CFDC0STS : Status Register ... 1747
34.2.5 CFDC0ERFL : Error Flag Register ... 1749
34.2.6 CFDC0DCFG : Data Bitrate Configuration Register ... 1754
34.2.7 CFDC0FDCFG : CANFD Configuration Register ... 1756
34.2.8 CFDC0FDCTR : CANFD Control Register ... 1758
34.2.9 CFDC0FDSTS : CANFD Status Register ... 1759
34.2.10 CFDC0FDCRC : CANFD CRC Register ... 1761
34.2.11 CFDGCFG : Global Configuration Register ... 1762
34.2.12 CFDGCTR : Global Control Register ... 1764
34.2.13 CFDGSTS : Global Status Register ... 1766
34.2.14 CFDGERFL : Global Error Flag Register ... 1767
34.2.15 CFDGTINTSTS : Global TX Interrupt Status Register ... 1768
34.2.16 CFDGTSC : Global Timestamp Counter Register ... 1770
34.2.17 CFDGAFLECTR : Global Acceptance Filter List Entry Control Register ... 1770
34.2.18 CFDGAFLCFG : Global Acceptance Filter List Configuration Register ... 1771
34.2.19 CFDGAFLIDr : Global Acceptance Filter List ID Registers (r = 1 to 16) ... 1771
34.2.20 CFDGAFLMr : Global Acceptance Filter List Mask Registers (r = 1 to 16) ... 1772
34.2.21 CFDGAFLP0r : Global Acceptance Filter List Pointer 0 Registers (r = 1 to 16) ... 1774
34.2.22 CFDGAFLP1r : Global Acceptance Filter List Pointer 1 Registers (r = 1 to 16) ... 1776
34.2.23 CFDRMNB : RX Message Buffer Number Register ... 1777
34.2.24 CFDRMND : RX Message Buffer New Data Register ... 1777
34.2.25 CFDRFCCa : RX FIFO Configuration/Control Registers a (a = 0 to 1) ... 1778
34.2.26 CFDRFSTSa : RX FIFO Status Registers a (a = 0 to 1) ... 1780
34.2.27 CFDRFPCTRa : RX FIFO Pointer Control Registers a (a = 0 to 1) ... 1781
34.2.28 CFDCFCC : Common FIFO Configuration/Control Register ... 1782
34.2.29 CFDCFSTS : Common FIFO Status Register ... 1785
34.2.30 CFDCFPCTR : Common FIFO Pointer Control Register ... 1787
34.2.31 CFDFESTS : FIFO Empty Status Register ... 1788
34.2.32 CFDFFSTS : FIFO Full Status Register ... 1788
34.2.33 CFDFMSTS : FIFO Message Lost Status Register ... 1789
34.2.34 CFDRFISTS : RX FIFO Interrupt Flag Status Register ... 1790
34.2.35 CFDCDTCT : DMA Transfer Control Register ... 1790

34.2.36 CFDCDTSTS : DMA Transfer Status Register ... 1791
34.2.37 CFDTMCi : TX Message Buffer Control Registers i (i = 0 to 3) ... 1792
34.2.38 CFDTMSTSj : TX Message Buffer Status Registers j (j = 0 to 3) ... 1793
34.2.39 CFDTMTRSTS : TX Message Buffer Transmission Request Status Register ... 1795
34.2.40 CFDTMTARSTS : TX Message Buffer Transmission Abort Request Status Register ... 1795
34.2.41 CFDTMTCSTS : TX Message Buffer Transmission Completion Status Register ... 1796
34.2.42 CFDTMTASTS : TX Message Buffer Transmission Abort Status Register ... 1796
34.2.43 CFDTMIEC : TX Message Buffer Interrupt Enable Configuration Register ... 1797
34.2.44 CFDTXQCC : TX Queue Configuration/Control Register ... 1798
34.2.45 CFDTXQSTS : TX Queue Status Register ... 1799
34.2.46 CFDTXQPCTR : TX Queue Pointer Control Register ... 1800
34.2.47 CFDTHLCC : TX History List Configuration/Control Register ... 1801
34.2.48 CFDTHLSTS : TX History List Status Register ... 1802
34.2.49 CFDTHLACC0 : TX History List Access Register 0 ... 1803
34.2.50 CFDTHLACC1 : TX History List Access Register 1 ... 1804
34.2.51 CFDTHLPCTR : TX History List Pointer Control Register ... 1805
34.2.52 CFDGRSTC : Global SW reset Register ... 1805
34.2.53 CFDGTSTCFG : Global Test Configuration Register ... 1806
34.2.54 CFDGTSTCTR : Global Test Control Register ... 1807
34.2.55 CFDGFDCFG : Global FD Configuration Register ... 1807
34.2.56 CFDGLOCKK : Global Lock Key Register ... 1808
34.2.57 CFDRPGACCk : RAM Test Page Access Registers k (k = 0 to 63) ... 1809
34.2.58 CFDGAFLIGNENT : Global AFL Ignore Entry Register ... 1809
34.2.59 CFDGAFLIGNCTR : Global AFL Ignore Control Register ... 1810
34.2.60 CFDRMIEC : RX Message Buffer Interrupt Enable Configuration Register ... 1810
34.2.61 Message Buffer Component Structure ... 1811
34.3 Modes of Operation ... 1831
34.3.1 Overview ... 1831
34.3.2 Global Modes ... 1831
34.3.3 Channel Modes ... 1840
34.3.4 Global Mode and Channel Mode Transition Interactions ... 1845
34.4 Initialization ... 1847
34.4.1 Initialization of CAN Clock, Bit Timing and Baud Rate ... 1847
34.4.2 CAN Module Configuration after Hardware Reset ... 1854
34.5 Acceptance Filtering Function using Global Acceptance Filter List (AFL) ... 1855
34.5.1 Overview ... 1855
34.5.2 Allocation of AFL Entries ... 1857
34.5.3 AFL Entry Description ... 1857
34.5.4 Entering Entries in the AFL ... 1859
34.5.5 Loopback Modes ... 1861
34.5.6 IDE Masking ... 1861
34.5.7 Updating AFL Entry during Communication ... 1862
34.6 FIFO Buffers and Normal Message Buffer Configuration ... 1864
34.6.1 Normal RX Message Buffers ... 1865
34.6.2 FIFO Buffers ... 1865
34.7 Interrupts and DMA ... 1870
34.7.1 Interrupts ... 1870
34.7.2 DMA Transfer ... 1873
34.8 Reception and Transmission ... 1876
34.8.1 Reception ... 1876

34.8.2 Transmission 1882
34.9 Test Mode 1896
34.9.1 Channel Specific Test Modes 1897
34.9.2 Global Test Modes 1899
34.10 RAM Area Configuration 1903
34.10.1 Examples 1905
34.10.2 OTB Area 1906
34.10.3 RAM Initialization Cycle 1906
34.11 Usage notes 1906
34.11.1 Module-stop function 1906
35. CANFD ECC (CNECC) 1907
35.1 Overview 1907
35.2 Register Descriptions 1907
35.2.1 EC710CTL : ECC Control Register 1907
35.2.2 EC710TMC : ECC Test Mode Control Register 1910
35.2.3 EC710TED : ECC Test Substitute Data Register 1911
35.2.4 EC710EAD0 : ECC Error Address Register 1911
35.3 Operation 1912
35.3.1 **ECC Function Setting** 1912
35.3.2 **ECC Decoder Testing** 1913
35.4 Interrupts 1914
36. Serial Peripheral Interface (SPI) 1915
36.1 Overview 1915
36.2 Register Descriptions 1919
36.2.1 SPDR : SPI Data Register 1919
36.2.2 SPDECR : SPI Delay Control Register 1921
36.2.3 SPCR : SPI Control Register 1922
36.2.4 SPCR2 : SPI Control Register 2 1927
36.2.5 SPCR3 : SPI Control Register 3 1929
36.2.6 SPCMDm : SPI Command Register (m = 0 to 7) 1931
36.2.7 SPDCR : SPI Data Control Register 1934
36.2.8 SPDCR2 : SPI Data Control Register 2 1935
36.2.9 SPSR : SPI Status Register 1936
36.2.10 SPTFSR : SPI Transfer FIFO Status Register 1942
36.2.11 SPRFSR : SPI Receive FIFO Status Register 1942
36.2.12 SPPSR : SPI Polling Register 1943
36.2.13 SPSRC : SPI Status Clear Register 1943
36.2.14 SPFCR : SPI FIFO Clear Register 1944
36.3 Operation 1944
36.3.1 Overview of SPI Operation 1945
36.3.2 Controlling the SPI Pins 1946
36.3.3 SPI System Configuration Examples 1947
36.3.4 Data Formats 1953
36.3.5 Transfer Formats 1966
36.3.6 Communication Operating Mode 1968
36.3.7 Transmit Buffer Empty and Receive Buffer Full Interrupts 1971
36.3.8 Idle Interrupt 1972
36.3.9 Communication End Interrupt 1974
36.3.10 Error Detection 1983
36.3.11 Initializing the SPI 1990

36.3.12 SPI Operation 1990
36.3.13 Clock Synchronous Operation 2010
36.3.14 Loopback Mode 2016
36.3.15 Self-Diagnosis of Parity Bit Function 2017
36.3.16 Interrupt Sources 2018
36.4 Event Link Controller Event Output 2019
36.4.1 Receive Buffer Full Event Output 2019
36.4.2 Transmit Buffer Empty Event Output 2019
36.4.3 Mode-Fault, Underrun, Overrun, Parity Error, or received data ready Event Output 2020
36.4.4 SPI Idle Event Output 2020
36.4.5 Communication End Event Output 2021
36.4.6 Synchronization bypass function 2023
36.5 Usage Notes 2023
36.5.1 Settings for the Module-Stop State 2023
36.5.2 Constraint on Low-Power Functions 2023
36.5.3 Constraints on Starting Transfer 2023
36.5.4 Constraints on Mode-Fault, Underrun, Overrun, Parity Error, or Receive Data Ready Event Output 2023
36.5.5 Constraints on the SPSR.SPRF and SPSR.SPTEF Flags 2024
37. Octal Serial Peripheral Interface (OSPI) 2025
37.1 Overview 2025
37.2 Register Descriptions 2026
37.2.1 OSPI Configuration Registers 2026
37.2.2 OSPI Control Registers 2033
37.2.3 OSPI Status Registers 2047
37.2.4 OSPI Interrupt Registers 2048
37.3 Operation 2052
37.3.1 xSPI Bus 2052
37.3.2 Manual-command 2064
37.3.3 Memory-mapping 2065
37.3.4 Pattern Control 2072
37.3.5 Integrity Checking 2073
37.3.6 Interrupts 2073
37.3.7 Flows of Operations 2074
37.4 Usage Notes 2082
37.4.1 Constraint on Memory Access 2082
37.4.2 Constraint on Burst Length 2082
37.4.3 Memory Write Combination Mode 2082
37.4.4 Module-stop function 2083
37.4.5 Restriction in 8D-8D-8D profile 1.0 format 2083
37.4.6 Byte data order in 8pin-DDR mode 2083
38. Decryption On The Fly (DOTF) 2084
38.1 Overview 2084
38.2 Block Diagram 2084
38.3 Register Descriptions 2084
38.3.1 CONVAREAST : DOTF Conversion Area Start Address Register 2085
38.3.2 CONVAREAD : DOTF Conversion Area End Address Register 2085
38.4 Operation 2086
38.5 System flow 2086
38.6 Usage notes 2086

38.6.1 Module-stop function 2086
39. Serial Sound Interface Enhanced (SSIE) 2087
39.1 Overview 2087
39.2 Register Descriptions 2092
39.2.1 SSICR : Control Register 2092
39.2.2 SSISR : Status Register 2102
39.2.3 SSIFCR : FIFO Control Register 2113
39.2.4 SSIFSR : FIFO Status Register 2119
39.2.5 SSIFTDR : Transmit FIFO Data Register 2122
39.2.6 SSIFRDR : Receive FIFO Data Register 2125
39.2.7 SSIOFR : Audio Format Register 2127
39.2.8 SSISCR : Status Control Register 2130
39.3 Communication Formats 2131
39.3.1 I^2S Format 2132
39.3.2 Monaural Format 2132
39.3.3 TDM Format 2134
39.4 Communication Modes 2135
39.4.1 Slave-mode Communication 2135
39.4.2 Master-mode Communication 2136
39.4.3 Transmission 2136
39.4.4 Reception 2136
39.4.5 Transmission and Reception 2136
39.5 Operation 2136
39.5.1 Idle State 2136
39.5.2 Communication States 2138
39.6 Communication Operation 2142
39.6.1 Start Communication 2142
39.6.2 Transmission 2143
39.6.3 Reception 2144
39.6.4 Transmission and Reception 2145
39.6.5 Halt Communication 2145
39.6.6 Error Handling 2146
39.6.7 Resume Communication 2147
39.7 Interrupts 2148
39.7.1 SSIEn_SSIF Interrupt (n = 0, 1) 2149
39.7.2 SSIE0_SSITXI Interrupt [Full-duplex communication] 2150
39.7.3 SSIE0_SSIRXI Interrupt [Full-duplex communication] 2150
39.7.4 SSIE1_SSIRT Interrupt [Half-duplex communication] 2150
39.8 Software Resets 2151
39.8.1 Software Reset Procedure 2151
39.9 Usage Notes 2152
39.9.1 Notes for Slave-mode Communication 2152
39.9.2 Notes for Master-mode Communication 2153
39.9.3 Notes for Communication Flow 2153
39.9.4 Write Access Restriction 2154
39.9.5 Module-stop function 2155
40. SD/MMC Host Interface (SDHI) 2156
40.1 Overview 2156
40.2 Register Descriptions 2157
40.2.1 SD_CMD : Command Type Register 2157

40.2.2 SD_ARG : SD Command Argument Register 2158
40.2.3 SD_ARG1 : SD Command Argument Register 1 2159
40.2.4 SD_STOP : Data Stop Register 2159
40.2.5 SD_SECCNT : Block Count Register 2160
40.2.6 SD_RSPi : SD Card Response Register i (i = 10, 32, 54) 2161
40.2.7 SD_RSPj : SD Card Response Register j (j = 1, 3, 5) 2161
40.2.8 SD_RSP76 : SD Card Response Register 76 2162
40.2.9 SD_RSP7 : SD Card Response Register 7 2162
40.2.10 SD_INFO1 : SD Card Interrupt Flag Register 1 2163
40.2.11 SD_INFO2 : SD Card Interrupt Flag Register 2 2166
40.2.12 SD_INFO1_MASK : SD INFO1 Interrupt Mask Register 2170
40.2.13 SD_INFO2_MASK : SD INFO2 Interrupt Mask Register 2171
40.2.14 SD_CLK_CTRL : SD Clock Control Register 2172
40.2.15 SD_SIZE : Transfer Data Length Register 2173
40.2.16 SD_OPTION : SD Card Access Control Option Register 2173
40.2.17 SD_ERR_STS1 : SD Error Status Register 1 2175
40.2.18 SD_ERR_STS2 : SD Error Status Register 2 2176
40.2.19 SD_BUF0 : SD Buffer Register 2177
40.2.20 SDIO_MODE : SDIO Mode Control Register 2178
40.2.21 SDIO_INFO1 : SDIO Interrupt Flag Register 2179
40.2.22 SDIO_INFO1_MASK : SDIO INFO1 Interrupt Mask Register 2180
40.2.23 SD_DMAEN : DMA Mode Enable Register 2181
40.2.24 SOFT_RST : Software Reset Register 2182
40.2.25 SDIF_MODE : SD Interface Mode Setting Register 2182
40.2.26 EXT_SWAP : Swap Control Register 2183
40.3 Operation 2183
40.3.1 SD/MMC Interface 2183
40.3.2 Card Detect/Write Protect 2186
40.3.3 Interrupt Request and DMA Transfer Request 2187
40.3.4 Communication Errors and Timeouts 2189
40.3.5 Command without Data Transfer (SD/MMC) 2190
40.3.6 Single Block Read (SD/MMC) 2192
40.3.7 Single Block Write (SD/MMC) 2194
40.3.8 Multiple Block Read (SD/MMC) 2195
40.3.9 Multiple Block Write (SD/MMC Using Internal Timer) 2197
40.3.10 Multiple Block Write (MMC using external timer) 2199
40.3.11 IO_RW_DIRECT Command (SD: CMD52) 2201
40.3.12 IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read) 2201
40.3.13 IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write) 2203
40.3.14 DMA Transfer (SD/MMC) 2205
40.3.15 Example of SD_CMD Register Setting 2207
40.4 Usage Notes 2211
40.4.1 SD_BUF Illegal Write Access (SD/MMC) 2211
40.4.2 Block Number Constraint for Multiple Block Read (SD) 2211
40.4.3 Automatic Control of SD/MMC Clock Output (SD/MMC) 2211
40.4.4 Control of the C52PUB Setting for Multiple Block Write (SD) 2212
40.4.5 Notes on SD_CLK_CTRL Register Settings (SD/MMC) 2212
40.4.6 Specification Limitations 2212
40.4.7 STP Bit Setting during Multiple Block Read (SD/MMC) 2212
40.4.8 Register Setting Notes 2213

40.4.9 Module-stop function 2213
41. Cyclic Redundancy Check (CRC) 2214
41.1 Overview 2214
41.2 Register Descriptions 2215
41.2.1 CRCCR0 : CRC Control Register 0 2215
41.2.2 CRCCR1 : CRC Control Register 1 2215
41.2.3 CRCDIR/CRCDIR_BY : CRC Data Input Register 2216
41.2.4 CRCDOR/CRCDOR_HA/CRCDOR_BY : CRC Data Output Register 2216
41.2.5 CRCSAR : Snoop Address Register 2217
41.3 Operation 2218
41.3.1 Basic Operation 2218
41.3.2 CRC Snoop Function 2221
41.4 Usage Notes 2222
41.4.1 Settings for the Module-Stop State 2222
41.4.2 Note on Transmission 2222
42. Boundary Scan 2224
42.1 Overview 2224
42.2 Register Descriptions 2224
42.2.1 JTIR : Instruction Register 2225
42.2.2 JTIDR : ID Code Register 2225
42.2.3 JTBPR : Bypass Register 2226
42.2.4 JTBSR : Boundary Scan Register 2226
42.3 Operation 2226
42.3.1 TAP Controller 2226
42.3.2 Commands 2227
42.4 Usage Notes 2228
43. Security Features 2230
43.1 Features 2230
43.2 Tamper Detection 2231
43.3 Arm Security Features 2231
43.3.1 Arm TrustZone Technology 2231
43.3.2 Privileged Control 2232
43.3.3 Security Attribution 2232
43.3.4 TrustZone Access Error 2238
43.4 Device Lifecycle Management 2239
43.4.1 Changing the Lifecycle State 2240
43.4.2 Protection and Authentication Level 2240
43.4.3 Serial Programming 2241
43.4.4 Device Lifecycle State and PL Change Example 2242
43.4.5 Failure Analysis 2242
43.5 Secure Key Injection 2242
43.6 Secure Factory Programming 2244
43.7 Secure Boot 2245
43.8 Field Updating in Dual Mode 2252
43.9 Register Description 2254
43.9.1 PSARB : Peripheral Security Attribution Register B 2254
43.9.2 PSARC : Peripheral Security Attribution Register C 2255
43.9.3 PSARD : Peripheral Security Attribution Register D 2257
43.9.4 PSARE : Peripheral Security Attribution Register E 2258
43.9.5 MSSAR : Module Stop Security Attribution Register 2260

43.9.6 PPARB : Peripheral Privilege Attribution Register B 2261
43.9.7 PPARC : Peripheral Privilege Attribution Register C 2262
43.9.8 PPARD : Peripheral Privilege Attribution Register D 2264
43.9.9 PPARE : Peripheral Privilege Attribution Register E 2265
43.9.10 MSPAR : Module Stop Privilege Attribution Register 2267
43.9.11 CFSAMONA : Code Flash Security Attribution Monitor Register A 2267
43.9.12 DFSAMON : Data Flash Security Attribution Monitor Register 2268
43.9.13 DLMMON : Device Lifecycle Management State Monitor Register 2268
43.9.14 MSAOAD : Master Security Attribution Operation After Detection Register 2269
43.9.15 MSAPT : Master Security Attribution Protect Register 2269
43.10 Usage Note 2270
43.10.1 Security or Privilege Bit Write Timing 2270
44. Renesas Secure IP (RSIP-E51A) 2271
44.1 Overview 2271
44.2 Operation 2273
44.2.1 Symmetric cryptography 2273
44.2.2 Asymmetric Cryptography 2274
44.2.3 Hash generator 2275
44.2.4 Encryption and Decryption 2275
44.3 Usage Notes 2276
44.3.1 Software Standby Mode 2276
44.3.2 Module-Stop Function Setting 2276
44.3.3 SPA/DPA Protections 2276
44.3.4 Restrictions on Open process 2276
45. 12-Bit A/D Converter (ADC12) 2277
45.1 Overview 2277
45.2 Register Descriptions 2282
45.2.1 ADDRn : A/D Data Registers n 2282
45.2.2 ADDBLDR : A/D Data Duplexing Register 2283
45.2.3 ADDBLDRn : A/D Data Duplexing Register n (n = A, B) 2284
45.2.4 ADTSDR : A/D Temperature Sensor Data Register 2286
45.2.5 ADOCDR : A/D Internal Reference Voltage Data Register 2287
45.2.6 ADVMDR : A/D VBATT Monitor Data Register 2288
45.2.7 ADRD : A/D Self-Diagnosis Data Register 2289
45.2.8 ADCSR : A/D Control Register 2290
45.2.9 ADANSA0 : A/D Channel Select Register A0 2294
45.2.10 ADANSA1 : A/D Channel Select Register A1 2294
45.2.11 ADANSB0 : A/D Channel Select Register B0 2295
45.2.12 ADANSB1 : A/D Channel Select Register B1 2296
45.2.13 ADADS0 : A/D-Converted Value Addition/Average Channel Select Register 0 2296
45.2.14 ADADS1 : A/D-Converted Value Addition/Average Channel Select Register 1 2297
45.2.15 ADADC : A/D-Converted Value Addition/Average Count Select Register 2298
45.2.16 ADCER : A/D Control Extended Register 2299
45.2.17 ADSTRGR : A/D Conversion Start Trigger Select Register 2301
45.2.18 ADEXICR : A/D Conversion Extended Input Control Registers 2303
45.2.19 ADSSTRn/ADSSTRL/ADSSTRT/ADSSTRO/ADSSTRV: A/D Sampling State Register (n = 0 to 8) 2305
45.2.20 ADSHCR : A/D Sample and Hold Circuit Control Register 2306
45.2.21 ADSHMSR : A/D Sample and Hold Operation Mode Selection Register 2306
45.2.22 ADDISCR : A/D Disconnection Detection Control Register 2307

45.2.23 ADGSPCR : A/D Group Scan Priority Control Register 2308
45.2.24 ADCMPCR : A/D Compare Function Control Register 2309
45.2.25 ADCMPANSR0 : A/D Compare Function Window A Channel Select Register 0 2311
45.2.26 ADCMPANSR1 : A/D Compare Function Window A Channel Select Register 1 2311
45.2.27 ADCMPANSER : A/D Compare Function Window A Extended Input Select Register 2312
45.2.28 ADCMPLR0 : A/D Compare Function Window A Comparison Condition Setting Register 0 2312
45.2.29 ADCMPLR1 : A/D Compare Function Window A Comparison Condition Setting Register 1 2313
45.2.30 ADCMPLER : A/D Compare Function Window A Extended Input Comparison Condition Setting Register 2314
45.2.31 ADCMPDRn : A/D Compare Function Window A Lower-Side/Upper-Side Level Setting Register (n = 0, 1) 2315
45.2.32 ADWINnLB : A/D Compare Function Window B Lower-Side/Upper-Side Level Setting Register (n = L, U) 2317
45.2.33 ADCMPSR0 : A/D Compare Function Window A Channel Status Register 0 2318
45.2.34 ADCMPSR1 : A/D Compare Function Window A Channel Status Register1 2319
45.2.35 ADCMPSER : A/D Compare Function Window A Extended Input Channel Status Register 2319
45.2.36 ADCMPBNSR : A/D Compare Function Window B Channel Select Register 2320
45.2.37 ADCMPBSR : A/D Compare Function Window B Status Register 2322
45.2.38 ADWINMON : A/D Compare Function Window A/B Status Monitor Register 2323
45.2.39 ADBUFEN : A/D Data Buffer Enable Register 2324
45.2.40 ADBUFPTR : A/D Data Buffer Pointer Register 2325
45.2.41 ADBUFn : A/D Data Buffer Registers n (n = 0 to 15) 2325
45.3 Operation 2326
45.3.1 Scanning Operation 2326
45.3.2 Single Scan Mode 2327
45.3.3 Continuous Scan Mode 2336
45.3.4 Group Scan Mode 2343
45.3.5 Compare Function for Windows A and B 2352
45.3.6 Analog Input Sampling and Scan Conversion Time 2356
45.3.7 Usage Example of A/D Data Register Automatic Clearing Function 2360
45.3.8 A/D-Converted Value Addition/Average Mode 2360
45.3.9 Disconnection Detection Assist Function 2360
45.3.10 Starting A/D Conversion with an Asynchronous Trigger 2362
45.3.11 Starting A/D Conversion with a Synchronous Trigger from a Peripheral Module 2363
45.3.12 Security Attribution between ADCn (n = 0, 1) and PORT (PmSAR (m = 0 to 9, A, B)) 2363
45.3.13 Using Data Buffers 2363
45.4 Interrupt Sources and DTC, DMAC Transfer Requests 2364
45.4.1 Interrupt Requests 2364
45.5 Event Link Function 2366
45.5.1 Event Output to the ELC 2366
45.5.2 ADC12 Operation through an Event from the ELC 2366
45.6 Usage Notes 2367
45.6.1 Constraints on Setting the Registers 2367
45.6.2 Constraints on Reading the Data Registers 2367
45.6.3 Constraints on Stopping A/D Conversion 2367
45.6.4 A/D Conversion Restart and Termination Timing 2369
45.6.5 Constraints on Scan End Interrupt Handling 2369
45.6.6 Settings for the Module-Stop Function 2369

45.6.7 Notes on Entering the Low-Power States ... 2369
45.6.8 Notes on Canceling Software Standby Mode ... 2369
45.6.9 Error in Absolute Accuracy When Disconnection Detection Assistance Is in Use ... 2369
45.6.10 Constraints on Operating Modes and Status Bits ... 2370
45.6.11 Notes on Board Design ... 2370
45.6.12 Constraints on Noise Prevention ... 2370
45.6.13 Port Settings When Using the ADC12 Input ... 2371
45.6.14 Relationship between ADC12 Units 0 and 1 and the ACMPHS ... 2371
45.6.15 Calculation for Sampling Time ... 2371
45.6.16 Notes on simultaneous operation with other modules ... 2372
46. 12-Bit D/A Converter (DAC12) ... 2373
46.1 Overview ... 2373
46.2 Register Descriptions ... 2374
46.2.1 DADRn : D/A Data Register n (n = 0, 1) ... 2374
46.2.2 DACR : D/A Control Register ... 2374
46.2.3 DADPR : DADRn Format Select Register ... 2376
46.2.4 DAADSCR : D/A A/D Synchronous Start Control Register ... 2376
46.2.5 DAAMPCR : D/A Output Amplifier Control Register ... 2377
46.2.6 DAASWCR : D/A Amplifier Stabilization Wait Control Register ... 2377
46.2.7 DAADUSR : D/A A/D Synchronous Unit Select Register ... 2378
46.3 Operation ... 2378
46.3.1 Reducing Interference between D/A and A/D Conversion ... 2379
46.4 Event Link Operation Setting Procedure ... 2381
46.4.1 DA0 Event Link Operation Setting Procedure ... 2381
46.4.2 DA1 Event Link Operation Setting Procedure ... 2381
46.5 Analog Output Control by Security Attribution ... 2381
46.6 Usage Notes on Event Link Operation ... 2382
46.7 Usage Notes ... 2382
46.7.1 Settings for the Module-Stop Function ... 2382
46.7.2 DAC12 Operation in the Module-Stop State ... 2382
46.7.3 DAC12 Operation in Software Standby Mode ... 2382
46.7.4 Constraint on Entering Deep Software Standby Mode ... 2382
46.7.5 Initialization Procedure with the Output Amplifier ... 2382
46.7.6 Initialization Procedure of the Output to internal modules ... 2383
46.7.7 Constraint on Usage When Interference Reduction between D/A and A/D Conversion Is Enabled ... 2383
47. Temperature Sensor (TSN) ... 2384
47.1 Overview ... 2384
47.2 Register Descriptions ... 2385
47.2.1 TSCR : Temperature Sensor Control Register ... 2385
47.2.2 TSCDR : Temperature Sensor Calibration Data Register ... 2385
47.3 Using the Temperature Sensor ... 2386
47.3.1 Preparation for Using the Temperature Sensor ... 2386
47.3.2 Procedures for Using the Temperature Sensor ... 2387
47.4 Usage Notes ... 2389
47.4.1 Settings for the Module-Stop Function ... 2389
47.4.2 Settings for the Software Standby Mode ... 2389
48. High-Speed Analog Comparator (ACMPHS) ... 2390
48.1 Overview ... 2390
48.2 Register Descriptions ... 2391

48.2.1 CMPCTL : Comparator Control Register 2391
48.2.2 CMPSEL0 : Comparator Input Select Register 2392
48.2.3 CMPSEL1 : Comparator Reference Voltage Select Register 2393
48.2.4 CMPMON : Comparator Output Monitor Register 2394
48.2.5 CPIOC : Comparator Output Control Register 2394
48.2.6 CPINTCTL : Comparator Interrupt Control Register 2394
48.2.7 CPMSKCTL : Comparator Interrupt Mask Control Register 2395
48.3 Operation 2395
48.4 Noise Filter 2398
48.5 ACMPHS Interrupts 2398
48.6 ACMPHS Output to the Event Link Controller (ELC) 2399
48.7 ACMPHS Pin Output 2399
48.8 Usage Notes 2399
48.8.1 Settings for the Module-Stop Function 2399
48.8.2 Settings for the DAC12 2400
48.8.3 Relationship with the ADC12 2400
48.8.4 ACMPHS Operation in Module-Stop State 2400
48.8.5 ACMPHS Operation in Software Standby Mode 2400
48.8.6 Setting the D/A Converter for Generating Reference Voltage 2400
49. Data Operation Circuit (DOC) 2401
49.1 Overview 2401
49.2 DOC Register Descriptions 2402
49.2.1 DOCR : DOC Control Register 2402
49.2.2 DOSR : DOC Flag Status Register 2403
49.2.3 DOSCR : DOC Flag Status Clear Register 2404
49.2.4 DODIR : DOC Data Input Register 2404
49.2.5 DODSR0 : DOC Data Setting Register 0 2404
49.2.6 DODSR1 : DOC Data Setting Register 1 2405
49.3 Operation 2405
49.3.1 Data Comparison Mode 2405
49.3.2 Data Addition Mode 2407
49.3.3 Data Subtraction Mode 2408
49.4 Interrupt Source 2408
49.5 Event Link Output 2409
49.6 Interrupt Handling and Event Linking 2409
49.7 Usage Notes 2409
49.7.1 Settings for the Module-Stop State 2409
50. SRAM 2410
50.1 Overview 2410
50.2 Register Descriptions 2410
50.2.1 SRAMSABARn : SRAM Security Attribute Boundary Address Register (n = 0, 1) 2410
50.2.2 SRAMSAR : SRAM Security Attribution Register 2411
50.2.3 SRAMPRCR_S : SRAM Protection Control Register for Secure 2412
50.2.4 SRAMPRCR_NS : SRAM Protection Control Register for Non-secure 2412
50.2.5 SRAMWTSC : SRAM Wait State Control Register 2413
50.2.6 SRAMCR0 : SRAM Control Register 0 2414
50.2.7 SRAMCR1 : SRAM Control Register 1 2415
50.2.8 SRAMECCRGN0 : SRAM0 ECC Region Control Register 2416
50.2.9 SRAMESR : SRAM Error Status Register 2416
50.2.10 SRAMESCLR : SRAM Error Status Clear Register 2417

50.2.11 SRAMEARn : SRAM Error Address Register (n = 0 to 2) 2418
50.3 Operation 2419
50.3.1 Module Stop Function 2419
50.3.2 Correction of ECC errors 2419
50.3.3 ECC Error Interrupt Function 2419
50.3.4 ECC Decoder Testing 2420
50.3.5 Parity Calculation Function 2420
50.3.6 TrustZone Filter function 2422
50.3.7 Interrupt Source 2423
50.3.8 Wait state 2423
50.4 Usage Note 2423
50.4.1 Instruction fetch from SRAM area 2423
50.4.2 Note of write to SRAMCR0, SRAMCR1 and SRAMECCRGN0 registers 2424
50.4.3 Notes on Using Error Checking of SRAM 2424
51. Standby SRAM 2425
51.1 Overview 2425
51.2 Register Descriptions 2425
51.2.1 STBRAMSABAR : Standby SRAM Security Attribute Boundary Address Register 2425
51.2.2 STBRAMPABAR_NS : Standby SRAM Privilege Attribute Boundary Address Register for Non-secure 2426
51.2.3 STBRAMPABAR_S : Standby SRAM Privilege Attribute Boundary Address Register for Secure 2426
51.2.4 STBRAMCR : Standby SRAM Control Register 2427
51.2.5 STBRAMEAR : Standby SRAM Error Address Register 2427
51.3 Operation 2427
51.3.1 Data Retention 2427
51.3.2 Module-stop Function 2428
51.3.3 Parity Calculation Function 2428
51.3.4 TrustZone Filter function 2430
51.3.5 Interrupt Source 2431
51.3.6 Wait state 2431
51.4 Usage Notes 2432
51.4.1 Instruction Fetch from the Standby SRAM Area 2432
51.4.2 Notes on Self-Diagnosis of the Standby SRAM 2432
52. Flash Memory 2433
52.1 Overview 2433
52.2 Structure of Memory 2435
52.3 Address Space 2438
52.4 Register Descriptions 2438
52.4.1 FCACHEE : Flash Cache Enable Register 2438
52.4.2 FCACHEIV : Flash Cache Invalidate Register 2439
52.4.3 FLWT : Flash Wait Cycle Register 2439
52.4.4 FSAR : Flash Security Attribution Register 2440
52.4.5 UIDRn : Unique ID Registers n (n = 0 to 3) 2441
52.4.6 PNRn : Part Numbering Register n (n = 0 to 3) 2441
52.4.7 MCUVER : MCU Version Register 2441
52.4.8 FWEPROR : Flash P/E Protect Register 2442
52.4.9 FASTAT : Flash Access Status Register 2442
52.4.10 FAEINT : Flash Access Error Interrupt Enable Register 2444
52.4.11 FRDYIE : Flash Ready Interrupt Enable Register 2444

52.4.12 FSADDR : FACI Command Start Address Register 2445
52.4.13 FEADDR : FACI Command End Address Register 2446
52.4.14 FMEPROT : Flash P/E Mode Entry Protection Register 2446
52.4.15 FCNTSELR : Flash Counter Select Register 2447
52.4.16 FCNTDATARn : Flash Counter Data Register n (n = 0, 1) 2448
52.4.17 FBPROT0 : Flash Block Protection Register 2449
52.4.18 FBPROT1 : Flash Block Protection for Secure Register 2449
52.4.19 FSTATR : Flash Status Register 2450
52.4.20 FENTRYR : Flash P/E Mode Entry Register 2454
52.4.21 FSUINITR : Flash Sequencer Setup Initialization Register 2455
52.4.22 FCMDR : FACI Command Register 2456
52.4.23 FBCCNT : Blank Check Control Register 2457
52.4.24 FBCSTAT : Blank Check Status Register 2457
52.4.25 FPSADDR : Data Flash Programming Start Address Register 2458
52.4.26 FSUASMON : Flash Startup Area Select Monitor Register 2458
52.4.27 FCPSR : Flash Sequencer Processing Switching Register 2459
52.4.28 FPCKAR : Flash Sequencer Processing Clock Notification Register 2460
52.4.29 FSUACR : Flash Startup Area Control Register 2460
52.4.30 FCKMHZ : Data Flash Access Frequency Register 2461
52.5 Flash Cache 2462
52.5.1 Feature of flash cache 2462
52.6 Operating Modes Associated with Flash Memory 2463
52.7 Overview of Functions 2464
52.8 Operating Modes of the Flash Sequencer 2465
52.9 FACI Commands 2466
52.9.1 List of FACI Commands 2466
52.9.2 Relationship between the Flash Sequencer State and FACI Commands 2468
52.9.3 Usage of FACI Commands 2469
52.10 Suspend Operation 2492
52.11 Protection Function 2492
52.11.1 Software Protection 2492
52.11.2 Error Protection 2496
52.11.3 Start-Up Program Protection 2499
52.11.4 Dual Bank Function 2503
52.11.5 Block Swap Function 2505
52.12 Security Function 2507
52.12.1 Security Flag for Startup Area Select 2507
52.12.2 Permanent Block Protect Setting 2507
52.12.3 Flash Memory Protection for TrustZone 2509
52.12.4 Data Flash Configuration Area Protection 2513
52.12.5 Anti-Rollback Counter 2514
52.13 Boot Mode 2518
52.13.1 Boot Mode (for the SCI Interface) 2519
52.13.2 Boot Mode (for the USB Interface) 2519
52.14 Using the Serial Programmer for Rewriting 2520
52.14.1 Environments for Serial Programming 2520
52.15 Programming through Self-Programming 2521
52.15.1 Overview 2521
52.15.2 Background Operation 2522
52.16 Reading Flash Memory 2523

52.16.1 Reading Code Flash Memory 2523
52.16.2 Reading Data Flash Memory 2523
52.17 Usage Notes 2523
53. Capture Engine Unit (CEU) 2526
53.1 Overview 2526
53.2 Register Descriptions 2529
53.2.1 CAPSR : Capture Start Register 2530
53.2.2 CAPCR : Capture Control Register 2534
53.2.3 CAMCR : Capture Interface Control Register 2536
53.2.4 CMCYR : Capture Interface Cycle Register 2540
53.2.5 CAMOR, CAMOR_x : Capture Interface Offset Register (x = B, M) 2541
53.2.6 CAPWR, CAPWR_x : Capture Interface Width Register (x = B, M) 2543
53.2.7 CAIFR : Capture Interface Input Format Register 2544
53.2.8 CRCNTR : CEU Register Control Register 2549
53.2.9 CRCMPR : CEU Register Forcible Control Register 2550
53.2.10 CFLCR, CFLCR_x : Capture Filter Control Register (x = B, M) 2551
53.2.11 CFSZR, CFSZR_x : Capture Filter Size Clip Register (x = B, M) 2553
53.2.12 CDWDR, CDWDR_x : Capture Destination Width Register (x = B, M) 2554
53.2.13 CDAYR, CDAYR_x : Capture Data Address Y Register (x = B, M) 2555
53.2.14 CDACR, CDACR_x : Capture Data Address C Register (x = B, M) 2556
53.2.15 CDBYR, CDBYR_x : Capture Data Bottom-Field Address Y Register (x = B, M) 2558
53.2.16 CDBCR, CDBCR_x : Capture Data Bottom-Field Address C Register (x = B, M) 2559
53.2.17 CBDSR, CBDSR_x : Capture Bundle Destination Size Register (x = B, M) 2560
53.2.18 CFWCR : Firewall Operation Control Register 2561
53.2.19 CLFCR, CLFCR_x : Capture Low-Pass Filter Control Register (x = B, M) 2562
53.2.20 CDOCR, CDOCR_x : Capture Data Output Control Register (x = B, M) 2562
53.2.21 CEIER : Capture Event Interrupt Enable Register 2568
53.2.22 CETCR : Capture Event Flag Clear Register 2569
53.2.23 CSTSR : Capture Status Register 2573
53.2.24 CDSSR : Capture Data Size Register 2574
53.2.25 CDAYR2, CDAYR2_x : Capture Data Address Y Register 2 (x = B, M) 2576
53.2.26 CDACR2, CDACR2_x : Capture Data Address C Register 2 (x = B, M) 2577
53.2.27 CDBYR2, CDBYR2_x : Capture Data Bottom-Field Address Y Register 2 (x = B, M) 2579
53.2.28 CDBCR2, CDBCR2_x : Capture Data Bottom-Field Address C Register 2 (x = B, M) 2580
53.2.29 CBWER : CEU Bufferable Write Enable Register 2581
53.3 Usage Notes 2581
53.3.1 Settings for the Module-Stop Function 2581
53.3.2 Conditions for Connection to an External Module 2582
53.3.3 Restrictions on Input/Output Functions 2582
54. Graphics Subsystem 2583
54.1 Graphics Overview 2583
54.1.1 Start and stop MIPI DSI 2584
55. 2D Drawing Engine (DRW) 2585
55.1 Overview 2585
55.2 Register Descriptions 2588
55.2.1 CONTROL : Geometry Control Register 2589
55.2.2 CONTROL2 : Surface Control Register 2590
55.2.3 IRQCTL : Interrupt Control Register 2593
55.2.4 CACHECTL : Cache Control Register 2594
55.2.5 STATUS : Status Control Register 2595

55.2.6 HWREVISION : Hardware Version and Feature Set ID Register 2596
55.2.7 COLOR1 : Base Color Register 2597
55.2.8 COLOR2 : Secondary Color Register 2597
55.2.9 PATTERN : Pattern Register 2598
55.2.10 LnSTART : Limiter n Start Value Register (n = 1 to 6) 2598
55.2.11 LnXADD : Limiter n X-Axis Increment Register (n = 1 to 6) 2598
55.2.12 LnYADD : Limiter n Y-Axis Increment Register (n = 1 to 6) 2599
55.2.13 LmBAND : Limiter m Band Width Parameter Register (n = 1, 2) 2599
55.2.14 TEXORIGIN : Texture Base Address Register 2599
55.2.15 TEXPITCH : Texels Per Texture Line Register 2600
55.2.16 TEXMASK : Texture Size or Texture Address Mask Register 2600
55.2.17 LUSTART : U Limiter Start Value Register 2600
55.2.18 LUXADD : U Limiter X-Axis Increment Register 2601
55.2.19 LUYADD : U Limiter Y-Axis Increment Register 2601
55.2.20 LVSTARTI : V Limiter Start Value Integer Part Register 2601
55.2.21 LVSTARTF : V Limiter Start Value Fractional Part Register 2602
55.2.22 LVXADDI : V Limiter X-Axis Increment Integer Part Register 2602
55.2.23 LVYADDI : V Limiter Y-Axis Increment Integer Part Register 2602
55.2.24 LVYXADDF : V Limiter Increment Fractional Parts Register 2603
55.2.25 TEXCLADDR : CLUT Start Address Register 2603
55.2.26 TEXCLDATA : CLUT Data Register 2603
55.2.27 TEXCLOFFSET : CLUT Offset Register 2604
55.2.28 COLKEY : Color Key Register 2604
55.2.29 SIZE : Bounding Box Dimension Register 2604
55.2.30 PITCH : Framebuffer Pitch And Spanstore Delay Register 2605
55.2.31 ORIGIN : Framebuffer Base Address Register 2605
55.2.32 DLISTSTART : Display List Start Address Register 2605
55.2.33 PERFTRIGGER : Performance Counters Control Register 2606
55.2.34 PERFCOUNTk : Performance Counter k (k = 1, 2) 2606
55.3 Drawing Features 2607
55.3.1 Drawing Features Summary 2607
55.3.2 Vector Drawing 2608
55.3.3 BitBLT 2609
55.4 Input and Output Data Formats 2609
55.4.1 Source and Destination Data 2610
55.4.2 Framebuffer Color Formats 2610
55.4.3 Texture Color Formats 2610
55.5 Texture Data Processing 2611
55.5.1 Texture Color Format 2611
55.5.2 Run Length Encoded (RLE) Unit 2611
55.5.3 Color Lookup Table (CLUT) 2613
55.5.4 Color Keying 2614
55.6 Rendering Pipeline 2615
55.6.1 Coordinate Transformation 2615
55.6.2 Rasterization 2615
55.6.3 Texturing 2627
55.6.4 Colorization 2631
55.6.5 Blending 2631
55.7 Rendering Modes 2635
55.7.1 Register Mode 2635

55.7.2 Display List Mode 2635
55.7.3 Stopping the Render Process 2638
55.8 Interrupts 2638
55.8.1 Interrupt sources 2639
55.8.2 Interrupt Control 2639
55.9 Performance Counters 2640
55.10 Stopping the 2D Drawing Engine Render Process 2640
55.11 MathML sample 2641
55.12 Usage notes 2642
55.12.1 Module-stop function 2642
55.12.2 Restriction for OSPI 2642
56. Graphics LCD Controller (GLCDC) 2643
56.1 Overview 2643
56.1.1 GLCDC Configuration 2645
56.1.2 Screen Format 2646
56.1.3 Graphics and Color Palette (CLUT) Data Formats 2647
56.1.4 Output Control for Data Format 2648
56.1.5 Output Control for Panel-Oriented Correction Process 2652
56.1.6 Output Control for TCON 2653
56.1.7 Graphics Data Interface 2653
56.1.8 Blending 2654
56.2 Register Descriptions 2656
56.2.1 GRm_CLUTn[y] : Color Palette (m = 1 to 2, n = 0 to 1, y = 0 to 255) 2656
56.2.2 BG_EN : Background Plane Setting Operation Control Register 2657
56.2.3 BG_PERI : Background Plane Setting Free-Running Period Register 2659
56.2.4 BG_SYNC : Background Plane Setting Synchronization Position Register 2659
56.2.5 BG_VSIZE : Background Plane Setting Full Image Vertical Size Register 2660
56.2.6 BG_HSIZE : Background Plane Setting Full Image Horizontal Size Register 2661
56.2.7 BG_BGC : Background Plane Setting Background Color Register 2662
56.2.8 BG_MON : Background Plane Setting Status Monitor Register 2662
56.2.9 GRn_VEN : Graphics n Register Update Control Register (n = 1 to 2) 2663
56.2.10 GRn_FLMRD : Graphics n Frame Buffer Read Control Register (n = 1, 2) 2664
56.2.11 GRn_FLM1 : Graphics n Frame Buffer Control Register 1 (n = 1 to 2) 2665
56.2.12 GRn_FLM2 : Graphics n Frame Buffer Control Register 2 (n = 1 to 2) 2665
56.2.13 GRn_FLM3 : Graphics n Frame Buffer Control Register 3 (n = 1 to 2) 2666
56.2.14 GRn_FLM5 : Graphics n Frame Buffer Control Register 5 (n = 1 to 2) 2666
56.2.15 GRn_FLM6 : Graphics n Frame Buffer Control Register 6 (n = 1, 2) 2667
56.2.16 GRn_AB1 : Graphics n Alpha Blending Control Register 1 (n = 1, 2) 2668
56.2.17 GRn_AB2 : Graphics n Alpha Blending Control Register 2 (n = 1, 2) 2670
56.2.18 GRn_AB3 : Graphics n Alpha Blending Control Register 3 (n = 1, 2) 2671
56.2.19 GRn_AB4 : Graphics n Alpha Blending Control Register 4 (n = 1, 2) 2672
56.2.20 GRn_AB5 : Graphics n Alpha Blending Control Register 5 (n = 1, 2) 2672
56.2.21 GRn_AB6 : Graphics n Alpha Blending Control Register 6 (n = 1, 2) 2673
56.2.22 GRn_AB7 : Graphics n Alpha Blending Control Register 7 (n = 1, 2) 2674
56.2.23 GRn_AB8 : Graphics n Alpha Blending Control Register 8 (n = 1, 2) 2675
56.2.24 GRn_AB9 : Graphics n Alpha Blending Control Register 9 (n = 1, 2) 2676
56.2.25 GRn_BASE : Graphics n Background Color Control Register (n = 1, 2) 2677
56.2.26 GRn_CLUTINT : Graphics n CLUT Table Interrupt Control Register (n = 1, 2) 2677
56.2.27 GRn_MON : Graphics n Status Monitor Register (n = 1, 2) 2678
56.2.28 GAMn_LATCH : Gamma n Register Update Control Register (n = G, B, R) 2679

56.2.29 GAM_SW : Gamma Correction Block Function Switch Register 2680
56.2.30 GAMn_LUT1 : Gamma n Correction Block Table Setting Register 1 (n = G, B, R)..... 2680
56.2.31 GAMn_LUT2 : Gamma n Correction Block Table Setting Register 2 (n = G, B, R)..... 2682
56.2.32 GAMn_LUT3 : Gamma n Correction Block Table Setting Register 3 (n = G, B, R)..... 2683
56.2.33 GAMn_LUT4 : Gamma n Correction Block Table Setting Register 4 (n = G, B, R)..... 2684
56.2.34 GAMn_LUT5 : Gamma n Correction Block Table Setting Register 5 (n = G, B, R)..... 2685
56.2.35 GAMn_LUT6 : Gamma n Correction Block Table Setting Register 6 (n = G, B, R)..... 2686
56.2.36 GAMn_LUT7 : Gamma n Correction Block Table Setting Register 7 (n = G, B, R)..... 2687
56.2.37 GAMn_LUT8 : Gamma n Correction Block Table Setting Register 8 (n = G, B, R)..... 2688
56.2.38 GAMn_AREA1 : Gamma n Correction Block Area Setting Register 1 (n = G, B, R)..... 2688
56.2.39 GAMn_AREA2 : Gamma n Correction Block Area Setting Register 2 (n = G, B, R)..... 2689
56.2.40 GAMn_AREA3 : Gamma n Correction Block Area Setting Register 3 (n = G, B, R)..... 2690
56.2.41 GAMn_AREA4 : Gamma n Correction Block Area Setting Register 4 (n = G, B, R)..... 2691
56.2.42 GAMn_AREA5 : Gamma n Correction Block Area Setting Register 5 (n = G, B, R)..... 2691
56.2.43 OUT_VLATCH : Output Control Block Register Update Control Register..... 2692
56.2.44 OUT_SET : Output Control Block Output Interface Register 2693
56.2.45 OUT_BRIGHT1 : Output Control Block Brightness Correction Register 1 2694
56.2.46 OUT_BRIGHT2 : Output Control Block Brightness Correction Register 2 2695
56.2.47 OUT_CONTRAST : Output Control Block Contrast Correction Register 2696
56.2.48 OUT_PDTHA : Output Control Block Panel Dither Correction Register 2697
56.2.49 OUT_CLKPHASE : Output Control Block Output Phase Control Register 2699
56.2.50 TCON_TIM : TCON Reference Timing Setting Register..... 2701
56.2.51 TCON_STVx1 : TCON Vertical Timing Setting Register x1 (x = A, B) 2703
56.2.52 TCON_STVx2 : TCON Vertical Timing Setting Register x2 (x = A, B) 2704
56.2.53 TCON_STHx1 : TCON Horizontal Timing Setting Register x1 (x = A, B) 2705
56.2.54 TCON_STHx2 : TCON Horizontal Timing Setting Register x2 (x = A, B) 2707
56.2.55 TCON_DE : TCON Data Enable Polarity Setting Register 2708
56.2.56 SYSCNT_DTCTEN : System Control Block State Detection Control Register 2708
56.2.57 SYSCNT_INTEN : System Control Block Interrupt Request Enable Control Register 2709
56.2.58 SYSCNT_STCLR : System Control Block Status Clear Register 2710
56.2.59 SYSCNT_STMON : System Control Block Status Monitor Register..... 2711
56.2.60 SYSCNT_PANEL_CLK : System Control Block Version and Panel Clock Control Register 2712
56.3 Operation..... 2713
56.3.1 Overall Control 2713
56.3.2 Screen Definition..... 2717
56.3.3 Underflow and Interrupts..... 2718
56.4 Usage notes 2720
56.4.1 Module-stop function..... 2720
57. MIPI PHY..... 2721
57.1 Overview..... 2721
57.2 Register Description 2721
57.2.1 DPHYREFCR : D-PHY Reference Clock Setting Register 2721
57.2.2 DPHYPLFCR : D-PHY PLL Frequency Control Register..... 2722
57.2.3 DPHYPLOCR : D-PHY PLL Operation Control Register 2723
57.2.4 DPHYESCCR : D-PHY Escape Mode Clock Control Register 2724
57.2.5 DPHYPWRCR : D-PHY Power Supplying Control Register 2724
57.2.6 DPHYSFR : D-PHY Status Flag Register 2725
57.2.7 DPHYOCR : D-PHY Operation Control Register 2725
57.2.8 DPHYTIM1 : D-PHY Timing Control Register 1 2726

57.2.9 DPHYTIM2 : D-PHY Timing Control Register 2 2726
57.2.10 DPHYTIM3 : D-PHY Timing Control Register 3 2727
57.2.11 DPHYTIM4 : D-PHY Timing Control Register 4 2727
57.2.12 DPHYTIM5 : D-PHY Timing Control Register 5 2729
57.2.13 DPHYTIM6 : D-PHY Timing Control Register 6 2730
57.3 Operation 2732
57.3.1 D-PHY Start-up Procedure 2732
57.3.2 D-PHY Stop Procedure 2732
58. MIPI DSI 2733
58.1 Overview 2733
58.2 Register Description 2733
58.2.1 ISR : Interrupt Status Register 2734
58.2.2 LINKSR : Link Status Register 2735
58.2.3 TXSETR : Transmit Set Register 2736
58.2.4 HSCLKSETR : HS Clock Set Register 2737
58.2.5 ULPSSETR : ULPS Set Register 2738
58.2.6 ULPSCR : ULPS Control Register 2738
58.2.7 RSTCR : Reset Control Register 2739
58.2.8 RSTSR : Reset Status Register 2740
58.2.9 DSISETR : DSI Set Register 2741
58.2.10 TXPPD0R : Transmit Packet Payload Data 0 Register 2743
58.2.11 TXPPD1R : Transmit Packet Payload Data 1 Register 2744
58.2.12 TXPPD2R : Transmit Packet Payload Data 2 Register 2744
58.2.13 TXPPD3R : Transmit Packet Payload Data 3 Register 2745
58.2.14 RXSR : Receive Status Register 2746
58.2.15 RXSCR : Receive Status Clear Register 2749
58.2.16 RXIER : Receive Interrupt Enable Register 2752
58.2.17 PRESPTOBTASETR : Peripheral Response Timeout BTA Set Register 2755
58.2.18 PRESPTOLPSETR : Peripheral Response Timeout LP Set Register 2755
58.2.19 PRESPTOHSSETR : Peripheral Response Timeout HS Set Register 2756
58.2.20 AKEPLATIR : Acknowledge and Error Report Packet Parameter Latest Info Register 2757
58.2.21 AKEPACMSR : Acknowledge and Error Report Packet Parameter Accumulate Status Register 2758
58.2.22 AKEPSCR : Acknowledge and Error Report Packet Parameter Status Clear Register 2759
58.2.23 RXRSSR : Receive Result Saved Status Register 2760
58.2.24 RXRSSCR : Receive Result Saved Status Clear Register 2761
58.2.25 RXRINFOOWSR : Receive Result Info Overwrite Status Register 2762
58.2.26 RXRINFOOWSCR : Receive Result Info Overwrite Status Clear Register 2763
58.2.27 RXRSSxR : Receive Result Save Slot-x Register (x = 0 to 3) 2764
58.2.28 RXPPD0R : Receive Packet Payload Data 0 Register 2766
58.2.29 RXPPD1R : Receive Packet Payload Data 1 Register 2767
58.2.30 RXPPD2R : Receive Packet Payload Data 2 Register 2767
58.2.31 RXPPD3R : Receive Packet Payload Data 3 Register 2768
58.2.32 HSTXTOSETR : HS TX Timeout Set Register 2769
58.2.33 LRXHTOSETR : LRX-H Timeout Set Register 2769
58.2.34 TATOSETR : TA Timeout Set Register 2770
58.2.35 FERRSR : Fatal Error Status Register 2770
58.2.36 FERRSCR : Fatal Error Status Clear Register 2772
58.2.37 FERRIER : Fatal Error Interrupt Enable Register 2773
58.2.38 CLSTPTSETR : Clock Lane Stop Time Set Register 2775

58.2.39 LPTRNSTSETR : LP Transition Time Set Register 2777
58.2.40 PLSR : Physical Lane Status Register 2778
58.2.41 PLSCR : Physical Lane Status Clear Register 2781
58.2.42 PLIER : Physical Lane Interrupt Enable Register 2782
58.2.43 VMSET0R : Video Mode Set 0 Register 2784
58.2.44 VMSET1R : Video Mode Set 1 Register 2785
58.2.45 VMSR : Video Mode Status Register 2786
58.2.46 VMSCR : Video Mode Status Clear Register 2788
58.2.47 VMIER : Video Mode Interrupt Enable Register 2789
58.2.48 VMPPSETR : Video Mode Pixel Packet Set Register 2790
58.2.49 VMVSSETR : Video Mode Vertical Size Set Register 2791
58.2.50 VMVPSETR : Video Mode Vertical Porch Set Register 2791
58.2.51 VMHSSETR : Video Mode Horizontal Size Set Register 2792
58.2.52 VMHPSETR : Video Mode Horizontal Porch Set Register 2793
58.2.53 SQCH0SET0R : Sequence Channel 0 Set 0 Register 2793
58.2.54 SQCH0SR : Sequence Channel 0 Status Register 2794
58.2.55 SQCH0SCR : Sequence Channel 0 Status Clear Register 2796
58.2.56 SQCH0IER : Sequence Channel 0 Interrupt Enable Register 2797
58.2.57 SQCH1SET0R : Sequence Channel 1 Set 0 Register 2799
58.2.58 SQCH1SR : Sequence Channel 1 Status Register 2799
58.2.59 SQCH1SCR : Sequence Channel 1 Status Clear Register 2801
58.2.60 SQCH1IER : Sequence Channel 1 Interrupt Enable Register 2803
58.2.61 SQCHnDSCmAR : Sequence Channel n Descriptor-m A Register (n = 0 to 1, m = 0 to 7) 2805
58.2.62 SQCHnDSCmBR : Sequence Channel n Descriptor-m B Register (n = 0 to 1, m = 0 to 7) 2806
58.2.63 SQCHnDSCmCR : Sequence Channel n Descriptor-m C Register (n = 0 to 1, m = 0 to 7) 2807
58.2.64 SQCHnDSCmDR : Sequence Channel n Descriptor-m D Register (n = 0 to 1, m = 0 to 7) 2808
58.3 Operation 2808
58.3.1 Interrupt 2808
58.3.2 Software Reset 2810
58.3.3 Start of HS Clock 2811
58.3.4 Stop of HS Clock 2811
58.3.5 Command Mode Operation 2811
58.3.6 Video Mode Operation 2815
58.3.7 Transition To/From ULPS 2816
58.3.8 Error Handling 2820
58.4 Interface 2824
58.4.1 Video mode Operation 2824
58.5 Usage Notes 2828
58.5.1 GLCDC register setting when using Video Mode operation 2828
58.5.2 Module-stop function 2828
59. Internal Voltage Regulator 2829
59.1 Overview 2829
59.2 Operation 2829
59.2.1 DCDC mode 2829
59.2.2 External VDD mode 2830
59.3 Usage Notes 2832
60. Electrical Characteristics 2833
60.1 Absolute Maximum Ratings 2834
60.2 DC Characteristics 2835
60.2.1 Tj/Ta Definition 2835

60.2.2 I/O V_{IH}, V_{IL} 2836
60.2.3 I/O I_{OH}, I_{OL} 2838
60.2.4 I/O V_{OH}, V_{OL}, and Other Characteristics 2840
60.2.5 Operating and Standby Current 2843
60.2.6 VCC Rise and Fall Gradient and Ripple Frequency 2858
60.2.7 Thermal Characteristics 2859
60.3 AC Characteristics 2864
60.3.1 Frequency 2864
60.3.2 Clock Timing 2865
60.3.3 Reset Timing 2869
60.3.4 Wakeup Timing 2870
60.3.5 NMI and IRQ Noise Filter 2874
60.3.6 Bus Timing 2876
60.3.7 I/O Ports, POEG, GPT, AGT, ULPT and ADC12 Trigger Timing 2889
60.3.8 CAC Timing 2894
60.3.9 SCI Timing 2894
60.3.10 SPI Timing 2904
60.3.11 OSPI Timing 2912
60.3.12 IIC Timing 2916
60.3.13 I3C Timing 2920
60.3.14 SSIE Timing 2937
60.3.15 SD/MMC Host Interface Timing 2941
60.3.16 ETHERC Timing 2943
60.3.17 CEU Timing 2947
60.3.18 GLCDC Timing 2949
60.3.19 CANFD Timing 2950
60.4 USB Characteristics 2951
60.4.1 USBFS Timing 2951
60.4.2 USBHS Timing 2953
60.5 MIPI D-PHY Characteristics 2956
60.6 ADC12 Characteristics 2958
60.7 DAC12 Characteristics 2965
60.8 TSN Characteristics 2966
60.9 OSC Stop Detect Characteristics 2966
60.10 POR and PVD Characteristics 2967
60.11 VBATT Characteristics 2970
60.12 ACMPHS Characteristics 2971
60.13 Flash Memory Characteristics 2972
60.13.1 Code Flash Memory Characteristics 2972
60.13.2 Data Flash Memory Characteristics 2973
60.13.3 Option Setting Memory (Code flash memory) Characteristics 2974
60.13.4 Option Setting Memory (Data flash memory) Characteristics 2975
60.13.5 Anti-rollback counter Characteristics 2975
60.14 Boundary Scan 2976
60.15 Joint European Test Action Group (JTAG) 2978
60.16 Serial Wire Debug (SWD) 2980
60.17 Embedded Trace Macro Interface (ETM) 2981
Appendix 1. Port States in Each Processing Mode 2983
Appendix 2. Package Dimensions 2985
Appendix 3. I/O Registers 2987

3.1 Peripheral Base Addresses 2987
3.2 Access Cycles 2990
Appendix 4. Note for Register R/W 2992
Appendix 5. Peripheral Variant 2993

Renesas RA8D1 Group

User's Manual

High-performance 480 MHz Arm® Cortex®-M85 core with Helium™, up to 2 MB code flash memory with Dual-bank, background and SWAP operation, 12 KB Data flash memory, and 1 MB SRAM with Parity/ECC. High-integration with Ethernet MAC controller, USB 2.0 High-Speed, CANFD, SDHI, I3C, Octal SPI, Decryption on-the-fly, Graphics LCD Controller, 2D Drawing Engine, MIPI DSI and advanced analog. Integrated Renesas Secure IP with cryptography accelerators, key management support, tamper detection and power analysis resistance in concert with Arm® TrustZone for integrated secure element functionality.

Features

Arm® Cortex®-M85 core with Helium™

- Armv8.1-M architecture profile
- Armv8-M Security Extension
- Maximum operating frequency: 480 MHz
- Memory Protection Unit (Arm MPU)
  - Protected Memory System Architecture (PMSAv8)
  - Secure MPU (MPU_S): 8 regions
  - Non-secure MPU (MPU_NS): 8 regions
- SysTick timer
  - Embeds two Systick timers: Secure and Non-secure instance
  - Driven by CPUCLK or MOCO divided by 8
- CoreSight™ ETM-M85

Memory

- Up to 2 MB code flash memory
- 12 KB data flash memory (100,000 program/erase (P/E) cycles)
- 1 MB SRAM including 128 KB of TCM

Connectivity

- Serial Communications Interface (SCI) × 6 , up to 60 Mbps
  - Asynchronous interfaces
  - 8 bit clock synchronous interface
  - Smart card interface
  - Simple IIC
  - Simple SPI
  - Manchester coding (SCI0)
  - Simple LIN (SCI0, SCI1)
- I^2C bus interface (IIC) × 2
- I^3C bus interface (I3C)
- Serial Peripheral Interface (SPI) × 2, up to 60 Mbps
- Octal Serial Peripheral Interface (OSPI)
- USB 2.0 Full-Speed Module (USBFS)
- USB 2.0 High-Speed Module (USBHS)
- CAN with Flexible Data-rate (CANFD) × 2
- Ethernet MAC/DMA Controller (ETHERC/EDMAC)
- SD/MMC Host Interface (SDHI) × 2
- Serial Sound Interface Enhanced (SSIE) × 2

Analog

- 12-bit A/D Converter (ADC12) × 2
- 12-bit D/A Converter (DAC12) × 2
- High-Speed Analog Comparator (ACMPHS) × 2
- Temperature Sensor (TSN)

Timers

- General PWM Timer 32-bit (GPT32) × 8
- General PWM Timer 16-bit (GPT16) × 6
- Low Power Asynchronous General Purpose Timer (AGT) × 2
- Ultra-Low-Power Timer (ULPT) × 2

Security and Encryption

- Renesas Secure IP (RSIP-E51A)
  - Symmetric cryptography: AES
  - Asymmetric cryptography: RSA, ECC
  - Message digest computation: HASH
  - 128 bit unique ID
- Arm® TrustZone®
  - Up to two or four regions for the code flash, depending on the bank mode
  - Up to two regions for the data flash
  - Up to two regions for the SRAM
  - Individual Secure or Non-secure security attribution for each peripheral
- Privileged control
- Device lifecyle management
- Secure boot
- Decryption on-the-fly (DOTF)
- Pin function
  - Up to three tamper-resistant pins
  - Secure pin multiplexing

System and Power Management

- Low power modes
- Battery backup function (VBATT)
- Realtime Clock (RTC) with calendar and VBATT support
- Event Link Controller (ELC)
- Data Transfer Controller (DTC)
- DMA Controller (DMAC) × 8
- Power-on reset
- Programable Voltage Detection (PVD) with voltage settings
- Watchdog Timer (WDT)
- Independent Watchdog Timer (IWDT)

Human Machine Interface (HMI)

- Graphics LCD Controller (GLCDC)
- 2D Drawing Engine (DRW)
- Capture Engine Unit (CEU)
- MIPI DSI

Multiple Clock Sources

- Main clock oscillator (MOSC) (8 to 48 MHz)
- Sub-clock oscillator (SOSC) (32.768 kHz)
- High-speed on-chip oscillator (HOCO) (16/18/20/32/48 MHz)
- Middle-speed on-chip oscillator (MOCO) (8 MHz)
- Low-speed on-chip oscillator (LOCO) (32.768 kHz)
- Clock trim function for HOCO/MOCO/LOCO
- PLL1/PLL2
- Clock out support

General-Purpose I/O Ports

- 5-V tolerance, open drain, input pull-up, switchable driving ability

Operating Voltage

- VCC: 1.68 to 3.6 V
- VCC2: 1.65 to 3.6 V

Operating Junction Temperature and Packages

- Tj = -40°C to +125°C
  - 176-pin LQFP (24 mm × 24 mm, 0.5 mm pitch)
  - 224-pin BGA (13 mm × 13 mm, 0.8 mm pitch)

1. Overview

The MCU integrates multiple series of software-compatible Arm®-based 32-bit cores that share a common set of Renesas peripherals to facilitate design scalability and efficient platform-based product development.

The MCU in this series incorporates a high-performance Arm® Cortex®-M85 core with Helium™running up to 480 MHz with the following features:

- Up to 2 MB code flash memory
- 1 MB SRAM (128 KB of TCM RAM, 896 KB of user SRAM)
- Octal Serial Peripheral Interface (OSPI)
- Ethernet MAC Controller (ETHERC), USBFS, USBHS, SD/MMC Host Interface
- Graphics LCD Controller (GLCDC)
- 2D Drawing Engine (DRW)
- MIPI DSI interface
- Analog peripherals
- Security and safety features

1.1 Function Outline

Table 1.1 Arm core

Feature	Functional description
Arm® Cortex®-M85 core	• Maximum operating frequency: up to 480 MHz • Arm® Cortex®-M85 core – Revision: (r0p2-00rel0) – ARMv8.1-M architecture profile – Armv8-M Security Extension – Floating Point Unit (FPU) compliant with the ANSI/IEEE Std 754-2008 Scalar half, single, and double-precision floating-point operation – M-profile Vector Extension (MVE) Integer, half-precision, and single-precision floating-point MVE (MVE-F) – Helium™ technology is M-profile Vector Extension (MVE) • Arm® Memory Protection Unit (Arm MPU) – Protected Memory System Architecture (PMSAv8) – Secure MPU (MPU_S): 8 regions – Non-secure MPU (MPU_NS): 8 regions • SysTick timer – Embeds two Systick timers: Secure instance (SysTick_S) and Non-secure instance (SysTick_NS) – Driven by CPUCLK or MOCO divided by 8 • CoreSight™ ETM-M85

Table 1.2 Memory

Feature	Functional description
Code flash memory	Maximum 2 MB of code flash memory. See section 52, Flash Memory.
Data flash memory	12 KB of data flash memory. See section 52, Flash Memory.
Option-setting memory	The option-setting memory determines the state of the MCU after a reset. See section 6, Option-Setting Memory.
SRAM	On-chip high-speed SRAM with either parity bit or Error Correction Code (ECC). SRAM0 is ECC. SRAM1 is Parity check. See section 50, SRAM.
Standby SRAM	On-chip SRAM that can retain data in Deep Software Standby mode 1. See section 51, Standby SRAM.
ROM	On-chip immutable ROM contains First Stage Bootloader (FSBL)

Table 1.3 System

Feature	Functional description
Operating modes	Three operating modes: ● Single-chip mode ● JTAG boot mode ● SCI/USB boot mode See section 3, Operating Modes.
Resets	This MCU provides 13 types of reset. See section 5, Resets.
Programable Voltage Detection (PVD)	The Programable Voltage Detection (PVD) module monitors the voltage level input to the VCC pin. The detection level can be selected by register settings. The PVD module consists of three separate voltage level detectors (PVD0, PVD1, PVD2). PVD0, PVD1, and PVD2 measure the voltage level input to the VCC pin. PVD registers allow your application to configure detection of VCC changes at various voltage thresholds. See section 7, Programmable Voltage Detection (PVD).
Clocks	● Main clock oscillator (MOSC) ● Sub-clock oscillator (SOSC) ● High-speed on-chip oscillator (HOCO) ● Middle-speed on-chip oscillator (MOCO) ● Low-speed on-chip oscillator (LOCO) ● PLL1/PLL2 ● Clock out support See section 8, Clock Generation Circuit.
Clock Frequency Accuracy Measurement Circuit (CAC)	The Clock Frequency Accuracy Measurement Circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock selected as the measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range. When measurement is complete or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated. See section 9, Clock Frequency Accuracy Measurement Circuit (CAC).
Interrupt Controller Unit (ICU)	The Interrupt Controller Unit (ICU) controls which event signals are linked to the Nested Vector Interrupt Controller (NVIC), the DMA Controller (DMAC), and the Data Transfer Controller (DTC) modules. The ICU also controls non-maskable interrupts. See section 13, Interrupt Controller Unit (ICU).
Low power modes	Power consumption can be reduced in multiple ways, including setting clock dividers, controlling EBCLK output, controlling SDCLK output, stopping modules, power gating control, selecting operating power control modes in normal operation, and transitioning to low power modes and processor low power modes. See section 10, Low Power Modes.
Battery backup function	A battery backup function is provided for partial powering by a battery. The battery-powered area includes the RTC, SOSC, backup register, tamper detection and VBATT_R voltage drop detection and switch between VCC and VBATT. See section 11, Battery Backup Function.
Register write protection	The register write protection function protects important registers from being overwritten due to software errors. The registers to be protected are set with the Protect Register (PRCR_S and PRCR_NS). See section 12, Register Write Protection.
Memory Protection Unit (MPU)	All bus masters have Memory Protection Units (MPUs). See section 15, Memory Protection Unit (MPU).

Table 1.4 Event link

Feature	Functional description
Event Link Controller (ELC)	The Event Link Controller (ELC) uses the event requests generated by various peripheral modules as source signals to connect them to different modules, allowing direct link between the modules without CPU intervention. See section 18, Event Link Controller (ELC).

Table 1.5 Direct memory access

Feature	Functional description
Data Transfer Controller (DTC)	A Data Transfer Controller (DTC) module is provided for transferring data when activated by an interrupt request. See section 17, Data Transfer Controller (DTC).
DMA Controller (DMAC)	The 8-channel direct memory access controller (DMAC) that can transfer data without intervention from the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address. See section 16, DMA Controller (DMAC).

Table 1.6 External bus interface

Feature	Functional description
External buses	• CS area (ECBI): Connected to the external devices (external memory interface) • SDRAM area (ECBI): Connected to the SDRAM (external memory interface) • OSPI area (EOBI): Connected to the OSPI (external device interface)

Table 1.7 Timers

Feature	Functional description
General PWM Timer (GPT)	The General PWM Timer (GPT) is a 32-bit timer with GPT32 × 8 channels and a 16-bit timer with GPT16 × 6 channels. PWM waveforms can be generated by controlling the up-counter, down-counter, or the up- and down-counter. In addition, PWM waveforms can be generated for controlling brushless DC motors. The GPT can also be used as a general-purpose timer. See section 21, General PWM Timer (GPT).
Port Output Enable for GPT (POEG)	The Port Output Enable (POEG) function can place the General PWM Timer (GPT) output pins in the output disable state See section 20, Port Output Enable for GPT (POEG).
Low Power Asynchronous General Purpose Timer (AGT)	The Low Power Asynchronous General Purpose Timer (AGT) is a 16-bit timer that can be used for pulse output, external pulse width or period measurement, and counting external events. This timer consists of a reload register and a down counter. The reload register and the down counter are allocated to the same address, and can be accessed with the AGT register. See section 22, Low Power Asynchronous General Purpose Timer (AGT).
Ultra-Low-Power Timer (ULPT)	The Ultra-Low-Power Timer (ULPT) is a 32-bit timer which can be used for outputting pulses or counting external events. This 32-bit timer consists of reload registers and a down-counter. The reload registers and the down-counter are allocated to the same address and can be accessed through the ULPTCNT register. See section 23, Ultra-Low-Power Timer (ULPT)
Realtime Clock (RTC)	The realtime clock (RTC) has two counting modes, calendar count mode and binary count mode, that are used by switching register settings. For calendar count mode, the RTC has a 100-year calendar from 2000 to 2099 and automatically adjusts dates for leap years. For binary count mode, the RTC counts seconds and retains the information as a serial value. Binary count mode can be used for calendars other than the Gregorian (Western) calendar. See section 24, Realtime Clock (RTC).
Watchdog Timer (WDT)	The Watchdog Timer (WDT) is a 14-bit down counter that can be used to reset the MCU when the counter underflows because the system has run out of control and is unable to refresh the WDT. In addition, the WDT can be used to generate a non-maskable interrupt or an underflow interrupt. See section 25, Watchdog Timer (WDT).
Independent Watchdog Timer (IWDT)	The Independent Watchdog Timer (IWDT) has a 14-bit down-counter, which resets the MCU by a reset output when the down-counter underflows. Alternatively, generation of an interrupt request when the counter underflows can be selected. This enables detection of a program runaway taking the refresh interval into account. The IWDT has two start modes: auto start mode, in which counting automatically starts after release from the reset state, and register start mode, in which counting is started by refreshing (writing to a specific register). See section 26, Independent Watchdog Timer (IWDT).

Table 1.8 Communication interfaces (1 of 2)

Feature	Functional description
Serial Communications Interface (SCI)	The Serial Communications Interface (SCI) × 6 channels have asynchronous and synchronous serial interfaces: ● Asynchronous interfaces (UART and Asynchronous Communications Interface Adapter (ACIA)) ● 8-bit clock synchronous interface ● Simple IIC (master-only) ● Simple SPI ● Smart card interface ● Manchester interface ● Simple LIN interface The smart card interface complies with the ISO/IEC 7816-3 standard for electronic signals and transmission protocol. All channels have FIFO buffers to enable continuous and full-duplex communication, and the data transfer speed can be configured independently using an on-chip baud rate generator. The maximum rate supported on this MCU. Refer to the electrical characteristics for the actual rate. See section 31, Serial Communications Interface (SCI).
I^2C Bus interface (IIC)	The I^2C Bus interface (IIC) has 2 channels. The IIC module conforms with and provides a subset of the NXP I^2C (Inter-Integrated Circuit) bus interface functions. See section 32, I^2C Bus Interface (IIC).
I3C Bus Interface (I3C)	The I3C Bus Interface (I3C) has 1 channel. The I3C module conform with and provide a subset of the NXP I^2C (Inter-Integrated Circuit) bus interface functions and a subset of the MIPI I3C. See section 33, I3C Bus Interface (I3C).
Serial Peripheral Interface (SPI)	The Serial Peripheral Interface (SPI) provides high-speed full-duplex synchronous serial communications with multiple processors and peripheral devices. The maximum rate supported on this MCU. Refer to the electrical characteristics for the actual rate. See section 36, Serial Peripheral Interface (SPI).
Control Area Network with Flexible Data-Rate Module (CANFD)	The CAN with Flexible Data-Rate (CANFD) module can handle classical CAN frames and CANFD frames complied with ISO 11898-1 standard. The module supports 4 transmit buffers per channel and 16 receive buffers per channel. See section 34, CAN with Flexible Data-rate (CANFD).
USB 2.0 Full-Speed module (USBFS)	The USB 2.0 Full-Speed module (USBFS) can operate as a host controller or device controller. The module supports full-speed and low-speed (host controller only) transfer as defined in Universal Serial Bus Specification 2.0. The module has an internal USB transceiver and supports all of the transfer types defined in Universal Serial Bus Specification 2.0. The USB has buffer memory for data transfer, providing a maximum of 10 pipes. Pipes 1 to 9 can be assigned any endpoint number based on the peripheral devices used for communication or based on your system. See section 29, USB 2.0 Full-Speed Module (USBFS).
USB 2.0 High-speed Module (USBHS)	The USB 2.0 High-Speed Module (USBHS) that operates as a host or a device controller compliant with the Universal Serial Bus (USB) Specification revision 2.0. The host controller supports USB 2.0 high-speed, full speed, and low-speed transfers, and the device controller supports USB 2.0 high-speed and full-speed transfers. The USBHS has an internal USB transceiver and supports all of the transfer types defined in the USB 2.0 specification. The USBHS has FIFO buffer for data transfers, providing a maximum of 10 pipes. See section 30, USB 2.0 High-Speed Module (USBHS).
Octal Serial Peripheral Interface (OSPI)	The Octal Serial Peripheral Interface (OSPI) is a memory controller that supports EXpanded Serial Peripheral Interface (xSPI) (JEDEC Standard JESD251, JESD251-1 and JESD252). The OSPI supports 1-bit, 2-bit, 4-bit and 8-bit protocols. JESD251 specifies two interface profiles where profile 1.0 is Octal SPI and profile 2.0 is HyperBus™ (HyperRAM™ and HyperFlash™). OSPI supports QSPI protocol. See section 37, Octal Serial Peripheral Interface (OSPI).

Table 1.8 Communication interfaces (2 of 2)

Feature	Functional description
Serial Sound Interface Enhanced (SSIE)	The Serial Sound Interface Enhanced (SSIE) peripheral provides functionality to interface with digital audio devices for transmitting I^2S/Monaural/TDM audio data over a serial bus. The SSIE supports an audio clock frequency of up to 50 MHz, and can be operated as a slave or master receiver, transmitter, or transceiver to suit various applications. The SSIE includes 32-stage FIFO buffers in the receiver and transmitter, and supports interrupts and DMA-driven data reception and transmission. See section 39, Serial Sound Interface Enhanced (SSIE).
SD/MMC Host Interface (SDHI)	The Secure Digital (SD) Card and Multi Media Card (MMC) Host Interface provides the functionality required to connect a variety of external memory cards to the MCU. The SDHI supports both 1- and 4-bit buses for connecting memory cards that support SD, SDHC, and SDXC formats. When developing host devices that are compliant with the SD Specifications, you must comply with the SD Host/Ancillary Product License Agreement (SD HALA). The MMC interface supports 1-bit, 4-bit, and 8-bit MMC buses that provide eMMC 4.51 (JEDEC Standard JESD 84-B451) device access. This interface also provides backward compatibility and supports high-speed SDR transfer modes. See section 40, SD/MMC Host Interface (SDHI).
Ethernet Controller (ETHERC)	One-channel Ethernet Controller (ETHERC) compliant with the Ethernet/IEEE802.3 Media Access Control (MAC) layer protocol. An ETHERC channel provides one channel of the MAC layer interface, connecting the MCU to the physical layer LSI (PHY-LSI) that allows transmission and reception of frames compliant with the Ethernet and IEEE802.3 standards. The ETHERC is connected to the Ethernet DMA Controller (EDMAC) so data can be transferred without using the CPU. See section 27, Ethernet MAC Controller (ETHERC).

Table 1.9 Analog

Feature	Functional description
12-bit A/D Converter (ADC12)	A 12-bit successive approximation A/D Converter is provided. Up to 25 analog input channels are selectable. Temperature sensor output, and internal reference voltage and VBATT 1/3 voltage monitor are selectable for conversion. See section 45, 12-Bit A/D Converter (ADC12).
12-bit D/A Converter (DAC12)	A 12-bit D/A Converter (DAC12) is provided. See section 46, 12-Bit D/A Converter (DAC12).
Temperature Sensor (TSN)	The on-chip Temperature Sensor (TSN) determines and monitors the die temperature for reliable operation of the device. The sensor outputs a voltage directly proportional to the die temperature, and the relationship between the die temperature and the output voltage is fairly linear. The output voltage is provided to the ADC12 for conversion and can be further used by the end application. See section 47, Temperature Sensor (TSN).
High-Speed Analog Comparator (ACMPHS)	The High-Speed Analog Comparator (ACMPHS) can be used to compare an analog input voltage with a reference voltage and to provide a digital output based on the result of conversion. Both the analog input voltage and the reference voltage can be provided to the ACMPHS from internal sources (D/A converter output or internal reference voltage) and an external source. Such flexibility is useful in applications that require go/no-go comparisons to be performed between analog signals without necessarily requiring A/D conversion. See section 48, High-Speed Analog Comparator (ACMPHS).

Table 1.10 Human machine interfaces (1 of 2)

Feature	Functional description
Graphics LCD Controller (GLCDC)	The Graphics LCD Controller (GLCDC) provides multiple functions and supports various data formats and panels. Key GLCDC features include: • GLCDC0BI/GLCDC1BI master function for accessing graphics data • Superimposition of three planes (single-color background plane, graphic 1-plane, and graphic 2-plane) • Support for many types of 32-bit or 16-bit per pixel graphics data and 8-bit, 4-bit, or 1-bit LUT data format • Digital interface signal output supporting a video image size of WXGA. See section 56, Graphics LCD Controller (GLCDC).

Table 1.10 Human machine interfaces (2 of 2)

Feature	Functional description
2D Drawing Engine (DRW)	The 2D Drawing Engine (DRW) provides flexible functions that can support almost any object geometry rather than being bound to only a few specific geometries such as lines, triangles, or circles. The edges of every object can be independently blurred or antialiased. Rasterization is executed at one pixel per clock on the bounding box of the object from left to right and top to bottom. The DRW can also raster from bottom to top to optimize the performance in certain cases. In addition, optimization methods are available to avoid rasterization of many empty pixels of the bounding box. The distances to the edges of the object are calculated by a set of edge equations for every pixel of the bounding box. These edge equations can be combined to describe the entire object. If a pixel is inside the object, it is selected for rendering. If it is outside, it is discarded. If it is on the edge, an alpha value can be chosen proportional to the distance of the pixel to the nearest edge for antialiasing. Every pixel that is selected for rendering can be textured. The resulting ARGB quadruple can be modified by a general raster operation approach independently for each of the four channels. The ARGB quadruples can then be blended with one of the multiple blend modes of the DRW. The DRW provides two inputs (texture read and framebuffer read), and one output (framebuffer write). The internal color format is always ARGB (8888). The color formats from the inputs are converted to the internal format on read and a conversion back is made on write. See section 55, 2D Drawing Engine (DRW).
Capture Engine Unit (CEU)	The Capture Engine Unit (CEU) is a capture module that fetches image data externally input and transfers it to the memory. See section 53, Capture Engine Unit (CEU).
MIPI DSI interface	The MIPI DSI interface module has a Transmitter function for MIPI Alliance Specification for Display Serial Interface 2 (DSI-2). This module supports MIPI Alliance Specification for Display Serial Interface 2 (DSI-2) Specification. And it works with MIPI Alliance Specification for D-PHY Specification. This module provides a solution for transmitting MIPI DSI-2 compliant digital video and packets. See section 57, MIPI PHY.

Table 1.11 Data processing

Feature	Functional description
Cyclic Redundancy Check (CRC) calculator	The Cyclic Redundancy Check (CRC) calculator generates CRC codes to detect errors in the data. The bit order of CRC calculation results can be switched for LSB-first or MSB-first communication. Additionally, various CRC-generation polynomials are available. The snoop function allows monitoring reads from and writes to specific addresses. This function is useful in applications that require CRC code to be generated automatically in certain events, such as monitoring writes to the serial transmit buffer and reads from the serial receive buffer. See section 41, Cyclic Redundancy Check (CRC).
Data Operation Circuit (DOC)	The Data Operation Circuit (DOC) compares, adds, and subtracts 32-bits data. When a selected condition applies, 32-bit data is compared and an interrupt can be generated. See section 49, Data Operation Circuit (DOC).

Table 1.12 Security (1 of 2)

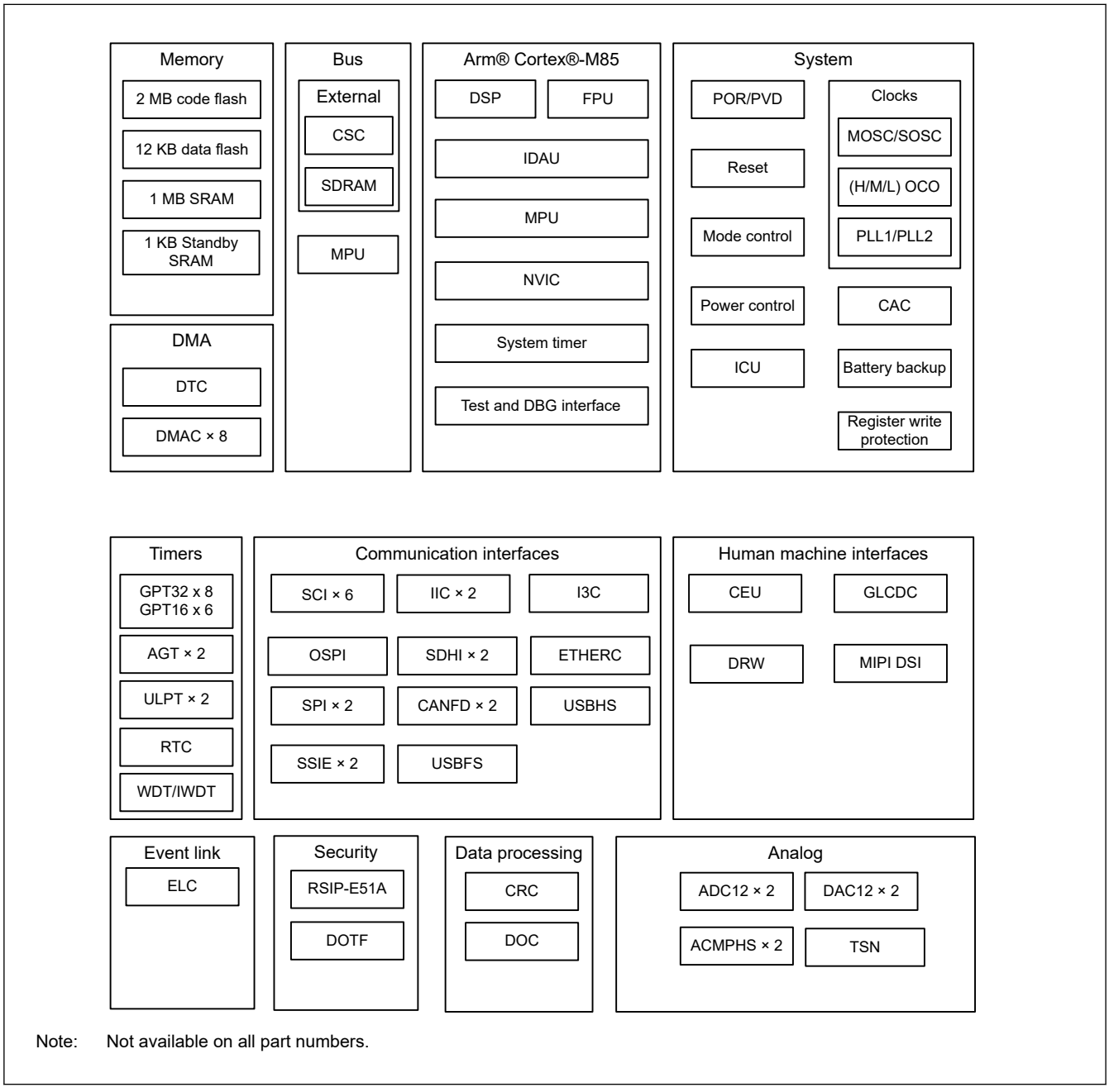
Feature	Functional description
Security function	• ARMv8-M TrustZone security • Privileged control • Device lifecycle management • Authentication Level (AL) • Key injection • Secure pin multiplexing • VBATT backup registers zeroization • Secure boot • Secure factory programming See section 43, Security Features .

Table 1.12 Security (2 of 2)

Feature	Functional description
Renesas Secure IP (RSIP-E51A)	• Symmetric cryptography: AES • Asymmetric cryptography: RSA, ECC • Message digest computation: HASH • 128-bit true random number generation circuit • 256-bit Hardware Unique Key (HUK) • 128-bit unique ID • OEM boot loader version • Key data for the decryption on-the-fly (DOTF) • SPA/DPA Protections See section 44, Renesas Secure IP (RSIP-E51A).
Decryption on-the-fly (DOTF)	Decryption on-the-fly (DOTF) decrypts the encrypted content stored in the external memory in real-time. See section 38, Decryption On The Fly (DOTF).

1.2 Block Diagram

Figure 1.1 shows a block diagram of the MCU superset. Some individual devices within the group have a subset of the features.

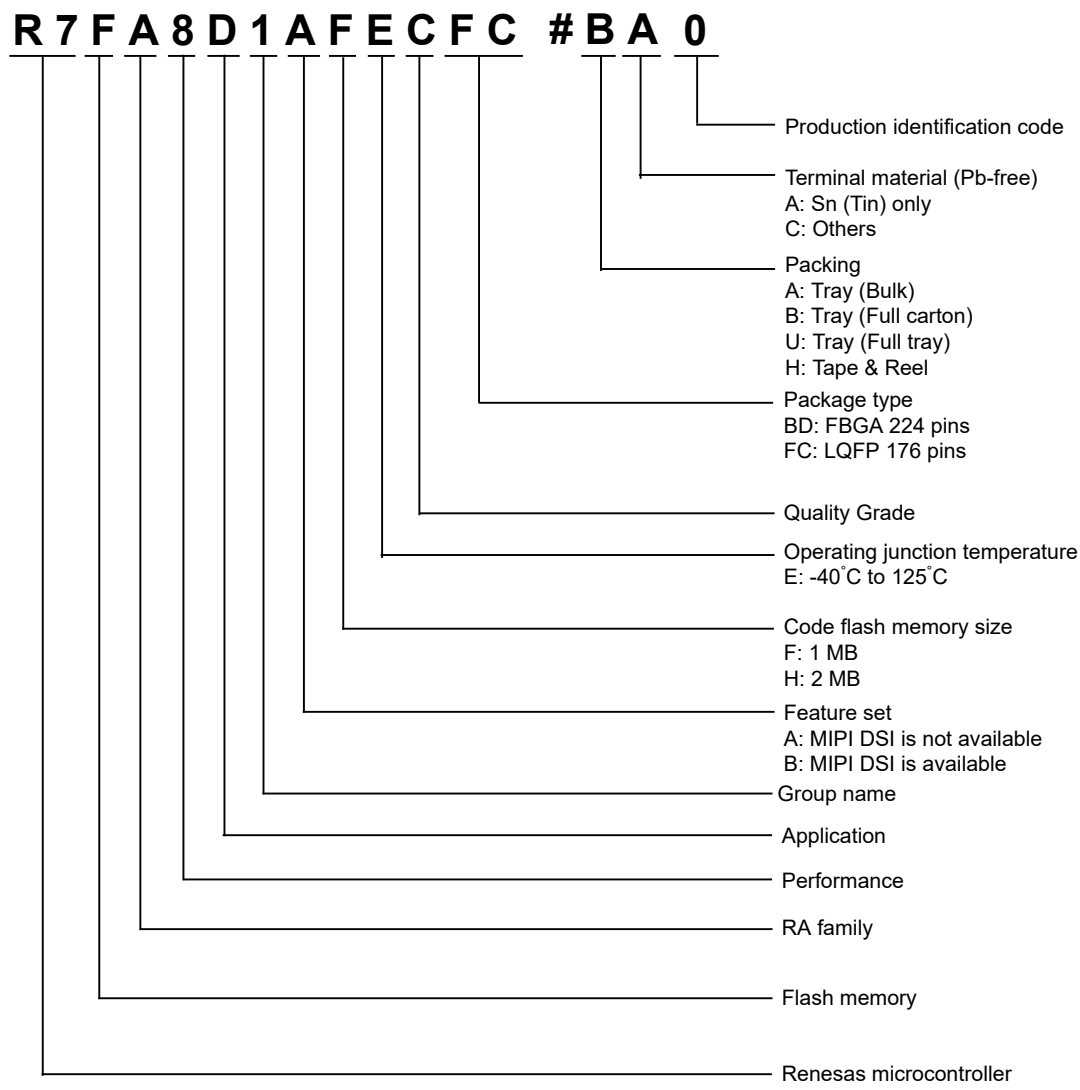


Note: Not available on all part numbers.

Figure 1.1 Block diagram

1.3 Part Numbering

Figure 1.2 shows the product part number information, including memory capacity and package type. Table 1.13 shows a list of products.



Note: Check the order screen for each product on the Renesas website for valid symbols after the #.

Figure 1.2 Part numbering scheme

Table 1.13 Product list

Product part number	MIPI	Package code	Code flash	Data flash	SRAM	Operating junction temperature
R7FA8D1AHECBD	—	PLBG0224GD-A	2 MB	12 KB	1 MB	−40 to +125°C
R7FA8D1BHECBD	✓					
R7FA8D1AHECFC	—	PLQP0176KJ-A				
R7FA8D1BHECFC	✓					
R7FA8D1AFECBD	—	PLBG0224GD-A	1 MB			
R7FA8D1BFECBD	✓					
R7FA8D1AFECFC	—	PLQP0176KJ-A				
R7FA8D1BFECFC	✓					

1.4 Function Comparison

Table 1.14 Function Comparison (1 of 2)

Parts number		R7FA8D1AxECBD	R7FA8D1BxECBD	R7FA8D1AxECFC	R7FA8D1BxECFC
Pin count		224		176	
Package		BGA		LQFP	
I/O Port		174	165	128	119
Code flash memory		2 MB, 1 MB			
Data flash memory		12 KB			
TCM		128 KB			
I/D Caches		32 KB			
SRAM		896 KB			
	Parity	512 KB			
	ECC	384 KB			
Standby SRAM		1 KB			
DMA	DTC	Yes			
	DMAC	8			
BUS	External bus	32-bit bus		16-bit bus	
	SDRAM	32-bit bus		16-bit bus	
System	CPU clock	480 MHz (max.)		400 MHz (max.)	
	CPU clock sources	MOSC, SOSC, HOCO, MOCO, PLL1P			
	CAC	Yes			
	WDT/IWDT	Yes			
	Backup register	128 B			
Communication	SCI	6			
	IIC	2			
	I3C	Yes			
	SPI	2			
	CANFD	2			
	USBFS	Yes			
	USBHS	Yes			
	OSPI	Yes			
	SSIE	2			
	SDHI/MMC	2			
	ETHERC	Yes			
Timers	GPT32*1	8			
	GPT16*1	6			
	AGT*1	2			
	ULPT*1	2			
	RTC	Yes			

Table 1.14 Function Comparison (2 of 2)

Parts number		R7FA8D1AxECBD	R7FA8D1BxECBD	R7FA8D1AxECFC	R7FA8D1BxECFC
Analog	ADC12	Unit 0: 12, Unit 1: 13 Shared channel pin: 4*2		Unit 0: 12, Unit 1: 12 Shared channel pin: 4*2	
	DAC12	2			
	ACMPHS	2			
	TSN	Yes			
HMI	GLCDC	RGB888			
	DRW	Yes			
	MIPI DSI	No	Yes	No	Yes
	CEU	Yes			
Data processing	CRC	Yes			
	DOC	Yes			
Event control	ELC	Yes			
Security		RSIP-E51A, Secure Debug, Secure Boot, Immutable Storage, TrustZone, Secure Factory Programming, and Lifecycle management			

Note: The product name differs depending on the memory size and MIPI DSI is supported. See section 1.3. Part Numbering.
Note 1. Available pins depend on the Pin count. For details, see section 1.7. Pin Lists.
Note 2. Some input channels of the ADC unit share the same port pins.

1.5 Pin Functions

Table 1.15 Pin functions (1 of 7)

Function	Signal	I/O	Description
Power supply	VCC, VCC2	Input	Power supply pin. Connect it to the system power supply. Connect this pin to VSS by a 0.1-μF capacitor. The capacitor should be placed close to the pin.
	VCC_DCDC	Input	Switching regulator power supply pin.
	VLO	I/O	Switching regulator pin.
	VCL	Input	Connect this pin to the VSS pin by the smoothing capacitor used to stabilize the internal power supply. Place the capacitor close to the pin.
	VBATT	Input	Battery Backup power pin
	VSS, VSS_DCDC	Input	Ground pin. Connect it to the system power supply (0 V).
Clock	XTAL	Output	Pins for a crystal resonator. An external clock signal can be input through the EXTAL pin.
	EXTAL	Input	
	XCIN	Input	Input/output pins for the sub-clock oscillator. Connect a crystal resonator between XCOUT and XCIN.
	XCOUT	Output	
	EXCIN	Input	External sub-clock input
	CLKOUT	Output	Clock output pin
Operating mode control	MD	Input	Pin for setting the operating mode. The signal level on this pin must not be changed during operation mode transition on release from the reset state.
System control	RES	Input	Reset signal input pin. The MCU enters the reset state when this signal goes low.
CAC	CACREF	Input	Measurement reference clock input pin
On-chip emulator	TMS	Input	On-chip emulator or boundary scan pins
	TDI	Input	
	TCK	Input	
	TDO	Output	
	TCLK	Output	Output clock for synchronization with the trace data
	TDATA0 to TDATA3	Output	Trace data output
	SWO	Output	Serial wire trace output pin
	SWDIO	I/O	Serial wire debug data input/output pin
	SWCLK	Input	Serial wire clock pin
Interrupt	NMI	Input	Non-maskable interrupt request pin
	IRQn	Input	Maskable interrupt request pins
	IRQn-DS	Input	Maskable interrupt request pins that can also be used in Deep Software Standby mode

Table 1.15 Pin functions (2 of 7)

Function	Signal	I/O	Description
External bus interface	EBCLK	Output	Outputs the external bus clock for external devices
	RD	Output	Strobe signal indicating that reading from the external bus interface space is in progress, active-low.
	WR	Output	Strobe signal indicating that writing to the external bus interface space is in progress, in 1-write strobe mode, active-low.
	WRn	Output	Strobe signals indicating that either group of data bus pins (D07 to D00, D15 to D08, D23 to D16 or D31 to D24) is valid in writing to the external bus interface space, in byte strobe mode, active-low.
	BCn	Output	Strobe signals indicating that either group of data bus pins (D07 to D00, D15 to D08, D23 to D16 or D31 to D24) is valid in access to the external bus interface space, in 1-write strobe mode, active-low.
	ALE	Output	Address latch signal when address/data multiplexed bus is selected.
	WAIT	Input	Input pin for wait request signals in access to the external space, active-low.
	CSn	Output	Select signals for CS areas, active-low
	A00 to A23	Output	Address bus
	D00 to D31	I/O	Data bus
	A00/D00 to A15/D15	I/O	Address/data multiplexed bus
SDRAM interface	SDCLK	Output	Outputs the SDRAM-dedicated clock
	CKE	Output	SDRAM clock enable signal
	SDCS	Output	SDRAM chip select signal, active low
	RAS	Output	SDRAM low address strobe signal, active low
	CAS	Output	SDRAM column address strobe signal, active low
	WE	Output	SDRAM write enable signal, active low
	DQMn	Output	SDRAM I/O data mask enable signal for DQ07 to DQ00, DQ15 to DQ08, DQ23 to DQ16 or DQ31 to DQ24.
	A00 to A16	Output	Address bus
	DQ00 to DQ31	I/O	Data bus
GPT	GTETRGA, GTETRGB, GTETRGC, GTETRGD	Input	External trigger input pins
	GTIOCnA, GTIOCnB	I/O	Input capture, output compare, or PWM output pins
	GTADSM0, GTADSM1	Output	A/D conversion start request monitoring output pins
	GTIU	Input	Hall sensor input pin U
	GTIV	Input	Hall sensor input pin V
	GTIW	Input	Hall sensor input pin W
	GTOUUP	Output	3-phase PWM output for BLDC motor control (positive U phase)
	GTOULO	Output	3-phase PWM output for BLDC motor control (negative U phase)
	GTOVUP	Output	3-phase PWM output for BLDC motor control (positive V phase)
	GTOVLO	Output	3-phase PWM output for BLDC motor control (negative V phase)
	GTOWUP	Output	3-phase PWM output for BLDC motor control (positive W phase)
	GTOWLO	Output	3-phase PWM output for BLDC motor control (negative W phase)

Table 1.15 Pin functions (3 of 7)

Function	Signal	I/O	Description
AGT	AGTEEn	Input	External event input enable signals
	AGTIOn	I/O	External event input and pulse output pins
	AGTOn	Output	Pulse output pins
	AGTOAn	Output	Output compare match A output pins
	AGTOBn	Output	Output compare match B output pins
ULPT	ULPTEEn	Input	External count control input
	ULPTEVIn	Input	External event input
	ULPTOn	Output	Pulse output
	ULPTOAn	Output	Output compare match A output
	ULPTOBn	Output	Output compare match B output
	ULPTEEn-DS	Input	External count control input that can also be used in Deep Software Standby mode1
	ULPTEVIn-DS	Input	External event input that can also be used in Deep Software Standby mode1
	ULPTOn-DS	Output	Pulse output that can also be used in Deep Software Standby mode1
	ULPTOAn-DS	Output	Output compare match A output that can also be used in Deep Software Standby mode1
	ULPTOBn-DS	Output	Output compare match B output that can also be used in Deep Software Standby mode1
RTC	RTCOUT	Output	Output pin for 1-Hz or 64-Hz clock
	RTCICn	Input	Time capture event input pins
SCI	SCKn	I/O	Input/output pins for the clock (clock synchronous mode)
	RXDn	Input	Input pins for received data (asynchronous mode/clock synchronous mode)
	TXDn	Output	Output pins for transmitted data (asynchronous mode/clock synchronous mode)
	CTSn_RTSn	I/O	Input/output pins for controlling the start of transmission and reception (asynchronous mode/clock synchronous mode), active-low.
	CTSn	Input	Input for the start of transmission.
	DEn	Output	Driver enable signal for RS-485
	SCLn	I/O	Input/output pins for the IIC clock (simple IIC mode)
	SDAn	I/O	Input/output pins for the IIC data (simple IIC mode)
	SCKn	I/O	Input/output pins for the clock (simple SPI mode)
	MISOn	I/O	Input/output pins for slave transmission of data (simple SPI mode)
	MOSIn	I/O	Input/output pins for master transmission of data (simple SPI mode)
	SSn	Input	Chip-select input pins (simple SPI mode), active-low
IIC	SCLn	I/O	Input/output pins for the clock
	SDAn	I/O	Input/output pins for data
I3C	I3C_SCL0	I/O	Input/output pins for the clock
	I3C_SDA0	I/O	Input/output pins for data

Table 1.15 Pin functions (4 of 7)

Function	Signal	I/O	Description
SPI	RSPCKA, RSPCKB	I/O	Clock input/output pin
	MOSIA, MOSIB	I/O	Input or output pins for data output from the master
	MISOA, MISOB	I/O	Input or output pins for data output from the slave
	SSLA0, SSLB0	I/O	Input or output pin for slave selection
	SSLA1 to SSLA3, SSLB1 to SSLB3	Output	Output pins for slave selection
CANFD	CRXn	Input	Receive data
	CTXn	Output	Transmit data
USBFS	VCC_USB	Input	Power supply pin
	VSS_USB	Input	Ground pin
	USB_DP	I/O	D+ pin of the USB on-chip transceiver. Connect this pin to the D+ pin of the USB bus.
	USB_DM	I/O	D- pin of the USB on-chip transceiver. Connect this pin to the D- pin of the USB bus.
	USB_VBUS	Input	USB cable connection monitor pin. Connect this pin to VBUS of the USB bus. The VBUS pin status (connected or disconnected) can be detected when the USB module is operating as a function controller.
	USB_EXICEN	Output	Low-power control signal for external power supply (OTG) chip
	USB_VBUSEN	Output	VBUS (5 V) supply enable signal for external power supply chip
	USB_OVRCURA, USB_OVRCURB	Input	Connect the external overcurrent detection signals to these pins. Connect the VBUS comparator signals to these pins when the OTG power supply chip is connected.
	USB_OVRCURA-DS, USB_OVRCURB-DS	Input	Overcurrent pins for USBFS that can also be used in Deep Software Standby mode1. Connect the external overcurrent detection signals to these pins. Connect the VBUS comparator signals to these pins when the OTG power supply chip is connected.
	USB_ID	Input	Connect the MicroAB connector ID input signal to this pin during operation in OTG mode
USBHS	VCC_USBHS	Input	Power supply pin
	VSS1_USBHS, VSS2_USBHS	Input	Ground pin
	AVCC_USBHS	Input	Analog power supply
	USBHS_RREF	I/O	Reference current source pin for the USBHS Must be connected to the VSS2_USBHS pin through a 2.2-kΩ (±1%) resistor.
	USBHS_DP	I/O	Input/output pin for the D+ data line of the USB bus
	USBHS_DM	I/O	Input/output pin for the D- data line of the USB bus
	USBHS_EXICEN	Output	Must be connected to the OTG power supply IC
	USBHS_ID	input	Must be connected to the OTG power supply IC
	USBHS_VBUSEN	Output	VBUS power supply enable pin for the USBHS
	USBHS_OVRCURA, USBHS_OVRCURB	Input	Overcurrent pin for the USBHS
	USBHS_OVRCURA-DS, USBHS_OVRCURB-DS	Input	Overcurrent pin for the USBHS that can also be used in Deep Software Standby mode1.
	USBHS_VBUS	Input	USB cable connection monitor input pin

Table 1.15 Pin functions (5 of 7)

Function	Signal	I/O	Description
OSPI	OM_SCLK	Output	Clock output (OCTACLK divided by 2)
	OM_SCLKN	Output	Inverted clock output (OCTACLK divided by 2)
	OM_CSn	Output	Chip select signal for an OctaFlash device, active-low
	OM_DQS	I/O	Read data strobe/write data mask signal
	OM_SIOn	I/O	Data input/output
	OM_RESET	Output	Reset signal for both slave devices, active-low
	OM_ECSINT1	Input	Error Correction Status and Interrupt for slave1
	OM_RSTO1	Input	Slave reset status for slave1
	OM_WP1	Output	Write Protect for slave1, active-low
SSIE	SSIBCK0, SSIBCK1	I/O	SSIE serial bit clock pins
	SSILRCK0/SSIFS0, SSILRCK1/SSIFS1	I/O	LR clock/frame synchronization pins
	SSITXD0	Output	Serial data output pin
	SSIRXD0	Input	Serial data input pin
	SSIDATA1	I/O	Serial data input/output pins
	AUDIO_CLK	Input	External clock pin for audio (input oversampling clock)
SDHI/MMC	SDnCLK	Output	SD clock output pins
	SDnCMD	I/O	Command output pin and response input signal pins
	SDnDATA0 to SDnDATA7	I/O	SD and MMC data bus pins
	SDnCD	Input	SD card detection pins
	SDnWP	Input	SD write-protect signals

Table 1.15 Pin functions (6 of 7)

Function	Signal	I/O	Description
ETHERC	REF50CK0	Input	50-MHz reference clock. This pin inputs reference signal for transmission/reception timing in RMII mode.
	RMII0_CRS_DV	Input	Indicates carrier detection signals and valid receive data on RMII0_RXD1 and RMII0_RXD0 in RMII mode.
	RMII0_TXDn	Output	2-bit transmit data in RMII mode
	RMII0_RXDn	Input	2-bit receive data in RMII mode
	RMII0_TXD_EN	Output	Output pin for data transmit enable signal in RMII mode
	RMII0_RX_ER	Input	Indicates an error occurred during reception of data in RMII mode
	ET0_CRS	Input	Carrier detection/data reception enable signal
	ET0_RX_DV	Input	Indicates valid receive data on ET0_ERXD3 to ET0_ERXD0
	ET0_EXOUT	Output	General-purpose external output pin
	ET0_LINKSTA	Input	Input link status from the PHY-LSI
	ET0_ETXDn	Output	4 bits of MII transmit data
	ET0_ERXDn	Input	4 bits of MII receive data
	ET0_TX_EN	Output	Transmit enable signal. Functions as signal indicating that transmit data is ready on ET0_ETXD3 to ET0_ETXD0.
	ET0_TX_ER	Output	Transmit error pin. Functions as signal notifying the PHY_LSI of an error during transmission.
	ET0_RX_ER	Output	Receive error pin. Functions as signal to recognize an error during reception.
	ET0_TX_CLK	Input	Transmit clock pin. This pin inputs reference signal for output timing from ET0_TX_EN, ET0_ETXD3 to ET0_ETXD0, and ET0_TX_ER.
	ET0_RX_CLK	Input	Receive clock pin. This pin inputs reference signal for input timing to ET0_RX_DV, ET0_ERXD3 to ET0_ERXD0, and ET0_RX_ER.
	ET0_COL	Input	Input collision detection signal
	ET0_WOL	Output	Receive Magic packets
	ET0_MDC	Output	Output reference clock signal for information transfer through ET0_MDIO
	ET0_MDIO	I/O	Input or output bidirectional signal for exchange of management data with PHY-LSI
Analog power supply	AVCC0	Input	Analog voltage supply pin. This is used as the analog power supply for the respective modules.
	AVSS0	Input	Analog ground pin. This is used as the analog ground for the respective modules. Supply this pin with the same voltage as the VSS pin.
	VREFH	Input	Analog reference voltage supply pin for the ADC12 (unit 1) and D/A Converter. Connect this pin to AVCC0 when not using the ADC12 (unit 1) and D/A Converter.
	VREFL	Input	Analog reference ground pin for the ADC12 and D/A Converter. Connect this pin to AVSS0 when not using the ADC12 (unit 1) and D/A Converter.
	VREFH0	Input	Analog reference voltage supply pin for the ADC12 (unit 0). Connect this pin to AVCC0 when not using the ADC12 (unit 0).
	VREFL0	Input	Analog reference ground pin for the ADC12. Connect this pin to AVSS0 when not using the ADC12 (unit 0).

Table 1.15 Pin functions (7 of 7)

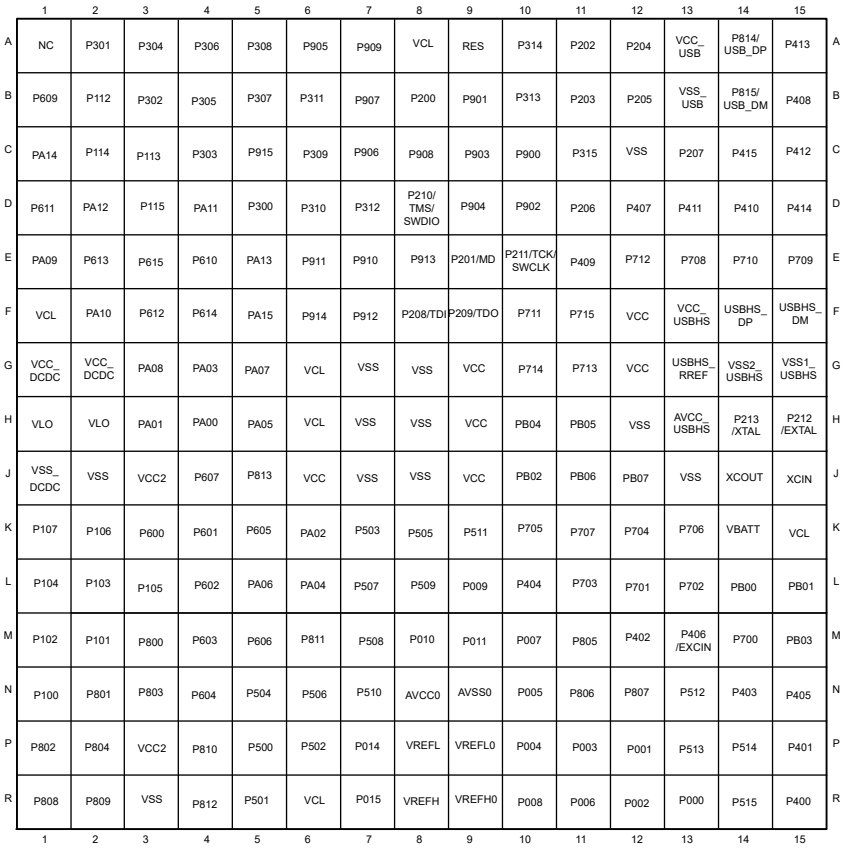
Function	Signal	I/O	Description
ADC12	ANmn	Input	Input pins for the analog signals to be processed by the A/D converter. (m: ADC unit number, n: pin number)
	ADTRGm	Input	Input pins for the external trigger signals that start the A/D conversion, active-low.
DAC12	DAn	Output	Output pins for the analog signals processed by the D/A converter.
ACMPHS	VCOUT	Output	Comparator output pin
	IVREFn	Input	Reference voltage input pins for comparator
	IVCMPn	Input	Analog voltage input pins for comparator
I/O ports	Pmn	I/O	General-purpose input/output pins (m: port number, n: pin number)
	P200	Input	General-purpose input pin
GLCDC	LCD_DATA23 to LCD_DATA00	Output	Data output pins for panel
	LCD_TCON3 to LCD_TCON0	Output	Output pins for panel timing adjustment
	LCD_CLK	Output	Panel clock output pin
	LCD_EXTCLK	Input	Panel clock source input pin
MIPI	VCC18_MIPI	Input	Power supply pin
	AVCC_MIPI	Input	Analog power supply
	VSS_MIPI	Input	Ground pin
	MIPI_CL_P	Output	DSI Clock Lane positive pin
	MIPI_CL_N	Output	DSI Clock Lane negative pin
	MIPI_DL0_P	I/O	DSI Data Lane 0 positive pin
	MIPI_DL0_N	I/O	DSI Data Lane 0 negative pin
	MIPI_DL1_P	Output	DSI Data Lane 1 positive pin
	MIPI_DL1_N	Output	DSI Data Lane 1 negative pin
	DSI_TE	Input	DSI Tearing Effect pin
CEU	VIO_D15 to VIO_D0	Input	CEU data bus pins
	VIO_CLK	Input	CEU clock pin
	VIO_VD	Input	CEU vertical sync pin
	VIO_HD	Input	CEU horizontal sync pin

1.6 Pin Assignments

The following figures show the pin assignments from the top view.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
A	NC	P301	P304	P306	P308	P905	P909	VCL	RES	MIPI_ DL1_N	MIPI_ CL_N	MIPI_ DL0_N	VCC_ USB	P814/ USB_DP	P413	A
B	P609	P112	P302	P305	P307	P311	P907	P200	AVCC_ MIPI	MIPI_ DL1_P	MIPI_ CL_P	MIPI_ DL0_P	VSS_ USB	P815/ USB_DM	P408	B
C	PA14	P114	P113	P303	P915	P309	P906	P908	P903	VSS_ MIPI	VCC18_MI PI	VSS	P207	P415	P412	C
D	P611	PA12	P115	PA11	P300	P310	P312	P210/ TMS/ SWDIO	P904	P902	P206	P407	P411	P410	P414	D
E	PA09	P613	P615	P610	PA13	P911	P910	P913	P201/MD	P211/TCK/ SWCLK	P409	P712	P708	P710	P709	E
F	VCL	PA10	P612	P614	PA15	P914	P912	P208/TDI	P209/TDO	P711	P715	VCC	VCC_ USBHS	USBHS_ DP	USBHS_ DM	F
G	VCC_ DCDC	VCC_ DCDC	PA08	PA03	PA07	VCL	VSS	VSS	VCC	P714	P713	VCC	USBHS_ RREF	VSS2_ USBHS	VSS1_ USBHS	G
H	VLO	VLO	PA01	PA00	PA05	VCL	VSS	VSS	VCC	PB04	PB05	VSS	AVCC_ USBHS	P213 /XTAL	P212 /EXTAL	H
J	VSS_ DCDC	VSS	VCC2	P607	P813	VCC	VSS	VSS	VCC	PB02	PB06	PB07	VSS	XCOUT	XCIN	J
K	P107	P106	P600	P601	P605	PA02	P503	P505	P511	P705	P707	P704	P706	VBATT	VCL	K
L	P104	P103	P105	P602	PA06	PA04	P507	P509	P009	P404	P703	P701	P702	PB00	PB01	L
M	P102	P101	P800	P603	P606	P811	P508	P010	P011	P007	P805	P402	P406 /EXCIN	P700	PB03	M
N	P100	P801	P803	P604	P504	P506	P510	AVCC0	AVSS0	P005	P806	P807	P512	P403	P405	N
P	P802	P804	VCC2	P810	P500	P502	P014	VREFL	VREFL0	P004	P003	P001	P513	P514	P401	P
R	P808	P809	VSS	P812	P501	VCL	P015	VREFH	VREFH0	P008	P006	P002	P000	P515	P400	R
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

Figure 1.3 Pin assignment for BGA 224-pin



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
A	NC	P301	P304	P306	P308	P905	P909	VCL	RES	P314	P202	P204	VCC_ USB	P814/ USB_DP	P413	A
B	P609	P112	P302	P305	P307	P311	P907	P200	P901	P313	P203	P205	VSS_ USB	P815/ USB_DM	P408	B
C	PA14	P114	P113	P303	P915	P309	P906	P908	P903	P900	P315	VSS	P207	P415	P412	C
D	P611	PA12	P115	PA11	P300	P310	P312	P210/ TMS/ SWDIO	P904	P902	P206	P407	P411	P410	P414	D
E	PA09	P613	P615	P610	PA13	P911	P910	P913	P201/MD	P211/TCK/ SWCLK	P409	P712	P708	P710	P709	E
F	VCL	PA10	P612	P614	PA15	P914	P912	P208/TDI	P209/TDO	P711	P715	VCC	VCC_ USBHS	USBHS_ DP	USBHS_ DM	F
G	VCC_ DCDC	VCC_ DCDC	PA08	PA03	PA07	VCL	VSS	VSS	VCC	P714	P713	VCC	USBHS_ RREF	VSS2_ USBHS	VSS1_ USBHS	G
H	VLO	VLO	PA01	PA00	PA05	VCL	VSS	VSS	VCC	PB04	PB05	VSS	AVCC_ USBHS	P213 /XTAL	P212 /EXTAL	H
J	VSS_ DCDC	VSS	VCC2	P607	P813	VCC	VSS	VSS	VCC	PB02	PB06	PB07	VSS	XCOUT	XCIN	J
K	P107	P106	P600	P601	P605	PA02	P503	P505	P511	P705	P707	P704	P706	VBATT	VCL	K
L	P104	P103	P105	P602	PA06	PA04	P507	P509	P009	P404	P703	P701	P702	PB00	PB01	L
M	P102	P101	P800	P603	P606	P811	P508	P010	P011	P007	P805	P402	P406 /EXCIN	P700	PB03	M
N	P100	P801	P803	P604	P504	P506	P510	AVCC0	AVSS0	P005	P806	P807	P512	P403	P405	N
P	P802	P804	VCC2	P810	P500	P502	P014	VREFL	VREFL0	P004	P003	P001	P513	P514	P401	P
R	P808	P809	VSS	P812	P501	VCL	P015	VREFH	VREFH0	P008	P006	P002	P000	P515	P400	R
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

Figure 1.4 Pin assignment for without_MIPI_BGA 224-pin

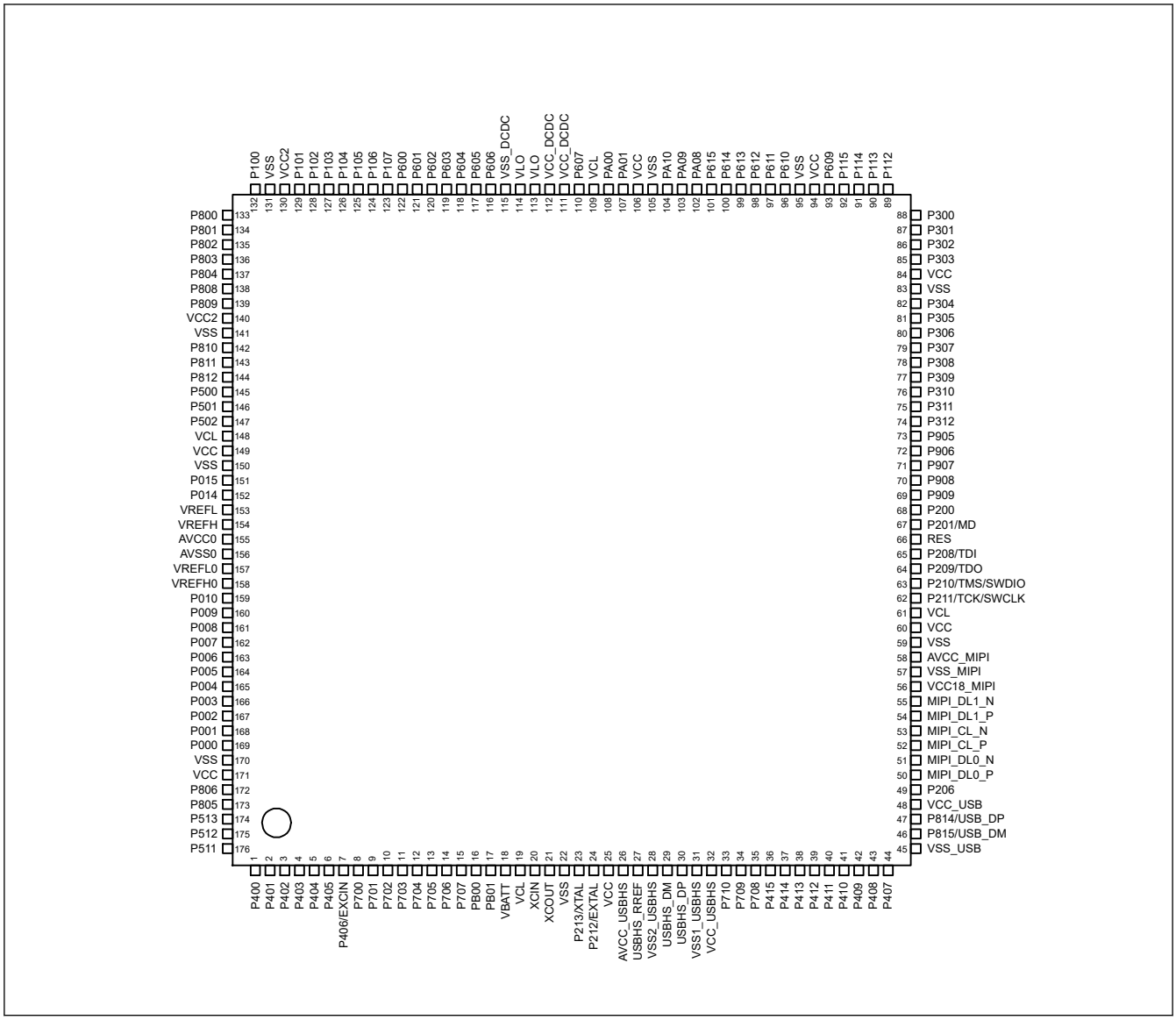


Figure 1.5 Pin assignment for LQFP 176-pin

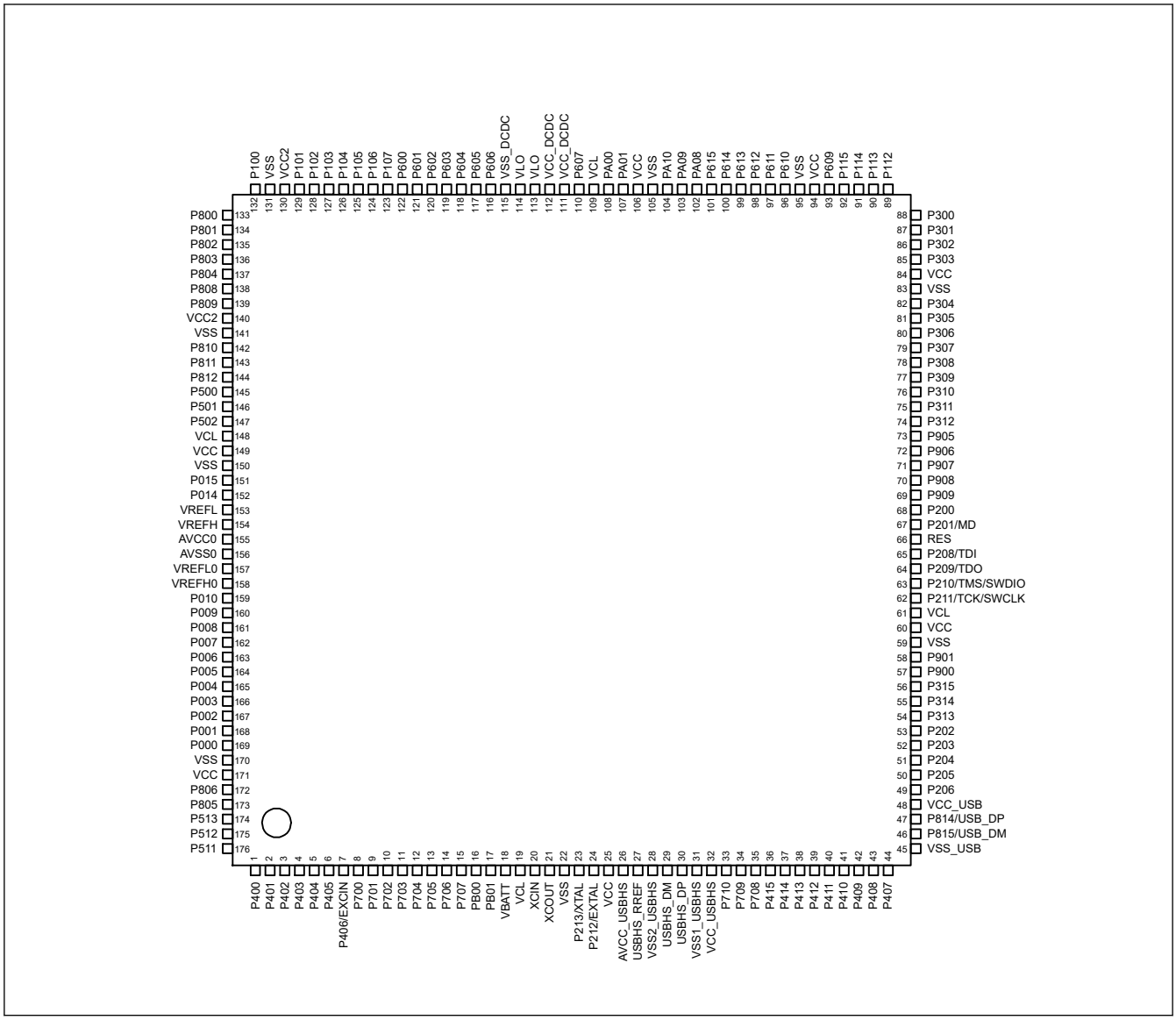


Figure 1.6 Pin assignment for without_MIPI_LQFP 176-pin

1.7 Pin Lists

Table 1.16 Pin list (1 of 7)

BGA224	BGA224 without MIPI	LQFP176	LQFP176 without MIPI	Power, System, Clock, Debug, CAC	I/O ports	Ex. Bus, SDRAM	Ex. Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/ USBHS/OSPI/ SSIE/SDHI/MMC/ EHTERC(MII,RMII)	GPT/AGT/ ULPT/RTC	ADC12/ DAC12/ ACMPHS	MIPI, GLCDC, CEU
R15	R15	1	1	-	P400	-	IRQ0	TXD1_A/MOSI1_A/SDA1_A/ I3C_SCL0/AUDIO_CLK/ SD1CLK_B/ET0_WOL/ET0_WOL	GTIOC6A/ AGTIO1	ADTRG1	VIO_D0
P15	P15	2	2	-	P401	-	IRQ5-DS	RXD1_A/MISO1_A/SCL1_A/ I3C_SDA0/CTX0/SD1CMD_B/ ET0_MDC/ET0_MDC	GTETRGA/ GTIOC6B	-	VIO_D1
M12	M12	3	3	CACREF	P402	-	IRQ4-DS	SCK1_A/DE1/CRX0/AUDIO_CLK/ SD1DAT0_B/ET0_MDIO/ ET0_MDIO	RTCIC0	-	-
N14	N14	4	4	-	P403	-	IRQ14-DS	CTS1_RTS1_A/SS4_A/DE1/ SSIBCK0_A/SD1DAT1_B/ ET0_LINKSTA/ET0_LINKSTA	GTIOC3A/ RTCIC1	-	-
L10	L10	5	5	-	P404	-	IRQ15-DS	CTS1_A/SSILRCK0/SSIFS0_A/ SD1DAT2_B/ET0_EXOUT/ ET0_EXOUT	GTIOC3B/ RTCIC2	-	VIO_D3
N15	N15	6	6	-	P405	-	-	SCK2_B/DE2/SSITXD0_A/ SD1DAT3_B/ET0_TX_EN/ RMII0_TXD_EN_B	GTIOC1A/ AGTIO1	-	VIO_D2
M13	M13	7	7	EXCIN	P406	-	-	TXD2_B/MOSI2_B/SDA2_B/ SSLA3_C/SSIRXD0_A/SD1CD/ ET0_RX_ER/RMII0_TXD1_B	GTIOC1B	-	VIO_D3
M14	M14	8	8	-	P700	-	-	RXD2_B/MISO2_B/SCL2_B/ MISOA_C/SSIDATA1_B/SD1WP/ ET0_ETXD1/RMII0_TXD0_B	GTIOC5A	-	VIO_D4
L12	L12	9	9	-	P701	-	-	CTS2_RTS2_B/SS2_B/DE2/ MOSIA_C/SSILRCK1/SSIFS1_B/ SD1DAT4_B/ET0_ETXD0/ REF50CK0_B	GTIOC5B/ ULPTO1	-	VIO_D5
L13	L13	10	10	-	P702	-	-	CTS2_B/RSPCKA_C/SSIBCK1_B/ SD1DAT5_B/ET0_ERXD1/ RMII0_RXD0_B	GTIOC6A/ ULPTO0	-	VIO_D6
L11	L11	11	11	-	P703	-	-	SSLA0_C/SD1DAT6_B/ ET0_ERXD0/RMII0_RXD1_B	GTIOC6B/AGTO1	VCOUT	VIO_D7
K12	K12	12	12	-	P704	-	-	SSLA1_C/CTX0/SD1DAT7_B/ ET0_RX_CLK/RMII0_RX_ER_B	GTADSM0/ AGTO0	-	VIO_D8
K10	K10	13	13	-	P705	-	-	CTS1_B/SSLA2_C/CRX0/ ET0_CRS/RMII0_CRS_DV_B	GTADSM1/ AGTIO0	-	VIO_D9
K13	K13	14	14	-	P706	-	IRQ7	RXD1_B/MISO1_B/SCL1_B/ USBHS_OVRCURB-DS	AGTIO0	-	VIO_D10
K11	K11	15	15	-	P707	-	IRQ8	TXD1_B/MOSI1_B/SDA1_B/ USBHS_OVRCURA-DS	-	-	LCD_DATA23_B
L14	L14	16	16	-	PB00	-	-	SCK1_B/DE1/USBHS_VBUSEN	-	-	LCD_DATA22_B
J10	J10	-	-	-	PB02	-	-	-	-	-	LCD_DATA21_B
M15	M15	-	-	-	PB03	-	-	-	-	-	LCD_DATA20_B
H10	H10	-	-	-	PB04	-	-	-	-	-	LCD_DATA19_B
L15	L15	17	17	-	PB01	ALE	-	CTS1_RTS1_B/SS1_B/DE1/ USBHS_VBUS	-	-	LCD_DATA18_B/ VIO_D11
H11	H11	-	-	-	PB05	-	-	-	-	-	LCD_DATA17_B
J11	J11	-	-	-	PB06	-	-	-	GTIOC9A	-	LCD_DATA16_B
J12	J12	-	-	-	PB07	-	-	-	GTIOC9B	-	LCD_DATA15_B
K14	K14	18	18	VBATT	-	-	-	-	-	-	-
K15	K15	19	19	VCL	-	-	-	-	-	-	-
J15	J15	20	20	XCIN	-	-	-	-	-	-	-
J14	J14	21	21	XCOUT	-	-	-	-	-	-	-
J13	J13	22	22	VSS	-	-	-	-	-	-	-
H14	H14	23	23	XTAL	P213	-	IRQ2	TXD1_C/MOSI1_C/SDA1_C	GTETRGC/ GTIOC0A/ ULPTEE0	ADTRG1	-
H15	H15	24	24	EXTAL	P212	-	IRQ3	RXD1_C/MISO1_C/SCL1_C	GTETRGD/ GTIOC0B/ AGTEE1	-	-
G12	G12	25	25	VCC	-	-	-	-	-	-	-

Table 1.16 Pin list (2 of 7)

BGA224	BGA224 without MIPI	LQFP176	LQFP176 without MIPI	Power, System, Clock, Debug, CAC	I/O ports	Ex. Bus, SDRAM	Ex. Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/ USBHS/OSPI/ SSIE/SDHI/MMC/ EHTERC(MII,RMII)	GPT/AGT/ ULPT/RTC	ADC12/ DAC12/ ACMPHS	MIPI, GLCDC, CEU
H13	H13	26	26	AVCC_USBHS	-	-	-	-	-	-	-
G13	G13	27	27	USBHS_RREF	-	-	-	-	-	-	-
G14	G14	28	28	VSS2_USBHS	-	-	-	-	-	-	-
F15	F15	29	29	-	-	-	-	USBHS_DM	-	-	-
F14	F14	30	30	-	-	-	-	USBHS_DP	-	-	-
G15	G15	31	31	VSS1_USBHS	-	-	-	-	-	-	-
F13	F13	32	32	VCC_USBHS	-	-	-	-	-	-	-
F11	F11	-	-	-	P715	-	-	RXD4_C/MISO4_C/SCL4_C	-	-	LCD_DATA14_B
G10	G10	-	-	-	P714	-	-	TXD4_C/MOSI4_C/SDA4_C	-	-	LCD_DATA13_B
H12	H12	-	-	VSS	-	-	-	-	-	-	-
G11	G11	-	-	-	P713	-	-	-	GTIOC2A/ AGTOA0	-	LCD_DATA12_B
E12	E12	-	-	-	P712	-	-	-	GTIOC2B/ AGTOB0	-	LCD_DATA11_B
F10	F10	-	-	-	P711	-	-	-	AGTEE0	-	LCD_DATA10_B
E14	E14	33	33	-	P710	CS5	-	CTS4_B	-	-	VIO_VD
E15	E15	34	34	-	P709	CS4	IRQ10	CTS4_RTS4_B/SS4_B/DE4	-	-	VIO_HD
E13	E13	35	35	CACREF	P708	WR1/BC1	IRQ11	SCK4_B/DE4/SSLB3_B/ AUDIO_CLK	-	-	VIO_CLK
C14	C14	36	36	-	P415	WAIT	IRQ8	TXD4_B/MOSI4_B/SDA4_B/ SSLB2_B/CTX1/SD0CD	GTADSM0/ GTIOC0A	-	VIO_D12
D15	D15	37	37	-	P414	A23	IRQ9	RXD4_B/MISO4_B/SCL4_B/ SSLB1_B/CRX1/SD0WP	GTADSM1/ GTIOC0B	-	VIO_D13
A15	A15	38	38	-	P413	A22	-	SSLB0_B/SD0CLK_A	GTOUUP/ ULPTEE1	-	-
C15	C15	39	39	-	P412	A21	-	CTS3_A/RSPCKB_B/ USB_EXICEN/USBHS_EXICEN/ SD0CMD_A	GTOULO/ AGTEE1	-	-
D13	D13	40	40	-	P411	A20	IRQ4	CTS3_RTS3_A/SS3_A/DE3/ MOSIB_B/USB_ID/USBHS_ID/ SD0DAT0_A	GTOVUP/ GTIOC9A/ AGTOA1	-	-
D14	D14	41	41	-	P410	A19	IRQ5	SCK3_A/DE3/SCL0_A/ MISOB_B/USB_OVRCURB-DS/ USBHS_OVRCURB/SD0DAT1_A	GTOVLO/ GTIOC9B/ AGTOB1	-	-
E11	E11	42	42	-	P409	A18	IRQ6	TXD3_A/MOSI3_A/SDA3_A/ SDA0_A/USB_OVRCURA-DS/ USBHS_OVRCURA	GTOWUP/ ULPTOA0	-	-
G6	G6	-	-	VCL	-	-	-	-	-	-	-
B15	B15	43	43	-	P408	A17	IRQ7	CTS4_A/RXD3_A/MISO3_A/ SCL3_A/SCL0_B/USB_VBUSEN/ USBHS_VBUSEN	GTOWLO/ GTIOC10A/ ULPTOB0	-	-
D12	D12	44	44	-	P407	CS6	-	CTS4_RTS4_A/SS4_A/DE4/ SDA0_B/SSLA3_A/USB_VBUS	GTIOC10B/ AGTIO0/RTCOUT	ADTRG0	-
B13	B13	45	45	VSS_USB	-	-	-	-	-	-	-
B14	B14	46	46	-	P815	-	-	CTX0/USB_DM	GTIOC8A	-	-
A14	A14	47	47	-	P814	-	-	CRX0/USB_DP	GTIOC8B	-	-
A13	A13	48	48	VCC_USB	-	-	-	-	-	-	-
C13	C13	-	-	-	P207	-	-	-	-	-	LCD_DATA9_B
D11	D11	49	49	-	P206	CS7	IRQ0-DS	RXD4_A/MISO4_A/ SCL4_A/SDA1_B/SSLA2_A/ USB_VBUSEN/SSIDATA1_A/ SD0DAT2_A	GTIU	-	DSI_TE
B12	-	50	-	MIPI_DL0_P	-	-	-	-	-	-	-
A12	-	51	-	MIPI_DL0_N	-	-	-	-	-	-	-
B11	-	52	-	MIPI_CL_P	-	-	-	-	-	-	-
A11	-	53	-	MIPI_CL_N	-	-	-	-	-	-	-
B10	-	54	-	MIPI_DL1_P	-	-	-	-	-	-	-
A10	-	55	-	MIPI_DL1_N	-	-	-	-	-	-	-
C11	-	56	-	VCC18_MIPI	-	-	-	-	-	-	-

Table 1.16 Pin list (3 of 7)

BGA224	BGA224 without MIPI	LQFP176	LQFP176 without MIPI	Power, System, Clock, Debug, CAC	I/O ports	Ex. Bus, SDRAM	Ex. Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/ USBHS/OSPI/ SSIE/SDHI/MMC/ EHTERC(MII,RMII)	GPT/AGT/ ULPT/RTC	ADC12/ DAC12/ ACMPHS	MIPI, GLCDC, CEU
C10	-	57	-	VSS_MIPI	-	-	-	-	-	-	-
B9	-	58	-	AVCC_MIPI	-	-	-	-	-	-	-
-	B12	-	50	CLKOUT	P205	-	IRQ1-DS	TXD4_A/MOSI4_A/ SDA4_A/SCL1_B/SSLA1_A/ USB_OVRCURA/SSILRCK1/ SSIFS1_A/SD0DAT3_A	GTIV/GTIOC4A/ AGTO1	-	-
-	A12	-	51	CACREF	P204	-	-	SCK4_A/DE4/SSLA0_A/ USB_OVRCURB/SSIBCK1_A/ SD0DAT4_A	GTIW/GTIOC4B/ AGTIO1	-	-
-	B11	-	52	-	P203	-	IRQ2-DS	RSPCKA_A/CTX0/SD0DAT5_A	GTIOC5A/ ULPTOA1	-	-
-	A11	-	53	-	P202	-	IRQ3-DS	MOSIA_A/CRX0/SD0DAT6_A	GTIOC5B/ ULPTOB1	-	-
-	B10	-	54	-	P313	-	-	CTS3_C/MISOA_A/SD0DAT7_A	-	-	-
-	A10	-	55	-	P314	-	-	CTS3_RTS3_C/SS3_C/DE3	-	ADTRG0	-
-	C11	-	56	-	P315	-	-	SCK3_C/DE3	-	-	-
-	C10	-	57	-	P900	-	-	TXD3_C/MOSI3_C/SDA3_C	-	-	-
-	B9	-	58	-	P901	-	-	RXD3_C/MISO3_C/SCL3_C	AGTIO1	-	-
D10	D10	-	-	-	P902	-	-	-	-	-	LCD_DATA8_B
C12	C12	59	59	VSS	-	-	-	-	-	-	-
F12	F12	60	60	VCC	-	-	-	-	-	-	-
C9	C9	-	-	-	P903	-	-	-	GTIOC11A	-	LCD_DATA7_B
D9	D9	-	-	-	P904	-	-	-	GTIOC11B	-	LCD_DATA6_B
A8	A8	61	61	VCL	-	-	-	-	-	-	-
E10	E10	62	62	TCK/SWCLK	P211	-	-	SCK9_B/DE9	GTOUUP/ GTIOC0A	-	-
D8	D8	63	63	TMS/SWDIO	P210	-	-	CTS9_RTS9_B/SS9_B/DE9	GTOULO/ GTIOC0B	-	-
F9	F9	64	64	TDO/SWO/ CLKOUT	P209	-	-	TXD9_B/MOSI9_B/SDA9_B/CTX1	GTOVUP/ GTIOC1A	-	-
F8	F8	65	65	TDI	P208	-	IRQ3	RXD9_B/MISO9_B/SCL9_B/CRX1	GTOVLO/ GTIOC1B	VCOUT	-
E8	E8	-	-	CLKOUT	P913	-	-	-	-	-	LCD_DATA5_B
A9	A9	66	66	RES	-	-	-	-	-	-	-
E9	E9	67	67	MD	P201	-	-	-	-	-	-
B8	B8	68	68	-	P200	-	NMI	-	-	-	-
F7	F7	-	-	-	P912	-	-	-	GTIOC3A	-	LCD_DATA4_B
E6	E6	-	-	-	P911	-	-	-	GTIOC3B	-	LCD_DATA3_B
E7	E7	-	-	-	P910	-	-	-	-	-	LCD_DATA2_B
A7	A7	69	69	-	P909	CS3/CAS	-	USBHS_EXICEN	GTIOC12A	-	LCD_DATA23_A
C8	C8	70	70	-	P908	CS2/RAS	IRQ11	USBHS_ID	GTIOC12B	-	LCD_DATA22_A
B7	B7	71	71	-	P907	A16/A16	IRQ10	USB_EXICEN	GTIOC13A	-	LCD_DATA21_A
C7	C7	72	72	-	P906	A15/A15	IRQ9	USB_ID	GTIOC13B	-	LCD_DATA20_A
A6	A6	73	73	-	P905	A14/A14	IRQ8	CTS3_B	-	-	LCD_DATA19_A
D7	D7	74	74	-	P312	A13/A13	-	CTS3_RTS3_B/SS3_B/DE3/ CTX0/ET0_TX_CLK	GTADSM0/ AGTOA1	-	LCD_DATA18_A
B6	B6	75	75	-	P311	A12/A12	-	SCK3_B/DE3/CRX0/ET0_TX_ER	GTADSM1/ AGTOB1	-	LCD_DATA17_A
G8	G8	-	-	VSS	-	-	-	-	-	-	-
D6	D6	76	76	-	P310	A11/A11	-	TXD3_B/MOSI3_B/SDA3_B/ ET0_ETXD2	AGTEE1	-	LCD_DATA16_A
C6	C6	77	77	-	P309	A10/A10	-	RXD3_B/MISO3_B/SCL3_B/ ET0_ETXD3	-	-	LCD_DATA15_A
A5	A5	78	78	TCLK	P308	A9/A9	-	CTS9_B/SD0CLK_B/ET0_MDC/ ET0_MDC	GTIU/ULPTOB1	-	-
B5	B5	79	79	TDATA0	P307	A8/A8	-	SD0CMD_B/ET0_MDIO/ ET0_MDIO	GTIV/ULPTOA1	-	-

Table 1.16 Pin list (4 of 7)

BGA224	BGA224 without MIPI	LQFP176	LQFP176 without MIPI	Power, System, Clock, Debug, CAC	I/O ports	Ex. Bus, SDRAM	Ex. Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/ USBHS/OSPI/ SSIE/SDHI/MMC/ EHTERC(MII,RMII)	GPT/AGT/ ULPT/RTC	ADC12/ DAC12/ ACMPHS	MIPI, GLCDC, CEU
A4	A4	80	80	TDATA1	P306	A7/A7	-	SD0CD/ET0_TX_EN/ RMII0_TXD_EN_A	GTIW/ULPTEVI1	-	-
B4	B4	81	81	TDATA2	P305	A6/A6	IRQ8	SD0WP/ET0_RX_ER/ RMII0_TXD1_A	GTOVUP/ ULPTEE1	-	-
A3	A3	82	82	TDATA3	P304	A5/A5	IRQ9	SD0DAT0_B/ET0_ETXD1/ RMII0_TXD0_A	GTOVLO/ GTIOC7A/ ULPTO1	-	-
G7	G7	83	83	VSS	-	-	-	-	-	-	-
G9	G9	84	84	VCC	-	-	-	-	-	-	-
C5	C5	-	-	-	P915	-	-	-	GTIOC5A	-	LCD_DATA1_B
F6	F6	-	-	-	P914	-	-	-	GTIOC5B	-	LCD_DATA0_B
C4	C4	85	85	-	P303	A4/A4	-	SD0DAT1_B/ET0_ETXD0/ REF50CK0_A	GTIOC7B	-	LCD_DATA14_A
B3	B3	86	86	-	P302	A3/A3	IRQ5	SD0DAT2_B/ET0_ERXD1/ RMII0_RXD0_A	GTOUUP/ GTIOC4A/ ULPTO0-DS	-	LCD_DATA13_A
A2	A2	87	87	-	P301	A2/A2	IRQ6	SD0DAT3_B/ET0_ERXD0/ RMII0_RXD1_A	GTOULO/ GTIOC4B/ AGTIO0/ ULPTEE0-DS	-	LCD_DATA12_A
D5	D5	88	88	-	P300	A1/A1/DQM3	IRQ4	SCK0_A/DE0/SSLA3_B/ ET0_RX_CLK/RMII0_RX_ER_A	GTIOC3A/ ULPTEVI0-DS	-	LCD_DATA11_A
B2	B2	89	89	-	P112	A0/BC0/A0/DQM1	-	TXD0_A/MOSI0_A/SDA0_A/ SSLA2_B/SSIBCK0_B/ET0_CRS/ RMII0_CRS_DV_A	GTIOC3B/ ULPTOB0-DS	-	LCD_DATA10_A
C3	C3	90	90	-	P113	CS1/CKE	-	RXD0_A/MISO0_A/SCL0_A/ SSLA1_B/SSILRCK0/SSIFS0_B/ ET0_EXOUT/ET0_EXOUT	GTIOC2A/ ULPTOA0-DS	-	LCD_DATA9_A
C2	C2	91	91	-	P114	CS0/WE	-	CTS0_RTS0_A/SS0_A/DE0/ SSLA0_B/SSIRXD0_B/ ET0_LINKSTA/ET0_LINKSTA	GTIOC2B	-	LCD_DATA8_A
D3	D3	92	92	-	P115	SDCS	-	CTS0_A/MOSIA_B/SSITXD0_B/ ET0_WOL/ET0_WOL	GTIOC5A	-	LCD_DATA7_A
B1	B1	93	93	-	P609	D8[A8/D8]/DQ8	-	TXD0_C/MOSI0_C/SDA0_C/ MISOA_B/CTX1/ET0_RX_DV	GTIOC5B/ ULPTOA1-DS	-	LCD_DATA6_A
D4	D4	-	-	-	PA11	WR2/BC2/DQM2	-	-	GTIOC6A	-	-
D2	D2	-	-	-	PA12	D16/DQ16	-	-	GTIOC6B	-	-
E5	E5	-	-	-	PA13	D17/DQ17	-	-	-	-	-
C1	C1	-	-	-	PA14	D18/DQ18	-	TXD9_C/MOSI9_C/SDA9_C	-	-	-
-	-	94	94	VCC	-	-	-	-	-	-	-
H7	H7	95	95	VSS	-	-	-	-	-	-	-
E4	E4	96	96	-	P610	D9[A9/D9]/DQ9	-	RXD0_C/MISO0_C/SCL0_C/ RSPCKA_B/CRX1/ET0_COL	GTIOC4A/ ULPTOB1-DS	-	LCD_DATA5_A
D1	D1	97	97	CLKOUT/CACREF	P611	D10[A10/D10]/DQ10	-	SCK0_C/DE0/MOSIA_B/ ET0_ERXD2	GTIOC4B	-	LCD_DATA4_A
F3	F3	98	98	-	P612	D11[A11/D11]/DQ11	-	CTS0_RTS0_C/SS0_C/DE0/ SSLA0_B/ET0_ERXD3	-	-	LCD_DATA3_A
E2	E2	99	99	-	P613	D12[A12/D12]/DQ12	-	CTS0_C	GTETRGA/ AGTO1	-	LCD_DATA2_A
F4	F4	100	100	-	P614	D13[A13/D13]/DQ13	-	-	GTETRGB/ AGTO0	-	LCD_DATA1_A
E3	E3	101	101	-	P615	D14[A14/D14]/DQ14	IRQ7	USB_VBUSEN	GTETRGC	-	LCD_DATA0_A
G3	G3	102	102	-	PA08	D15[A15/D15]/DQ15	IRQ6	-	GTETRGD	-	LCD_TCON3_A
E1	E1	103	103	-	PA09	EBCLK/SDCLK	IRQ5	-	-	-	LCD_TCON2_A
F2	F2	104	104	-	PA10	WR/WR0/DQM0	IRQ4	-	-	-	LCD_TCON1_A
F5	F5	-	-	-	PA15	D19/DQ19	-	RXD9_C/MISO9_C/SCL9_C	-	-	-
J5	J5	-	-	-	P813	D20/DQ20	-	-	-	-	-
G5	G5	-	-	-	PA07	D21/DQ21	-	-	GTIOC7A	-	-
L5	L5	-	-	-	PA06	D22/DQ22	-	CTS2_C	GTIOC7B	-	-
H5	H5	-	-	-	PA05	D23/DQ23	-	CTS2_RTS2_C/SS2_C/DE2	-	-	-
L6	L6	-	-	-	PA04	D24/DQ24	-	SCK2_C/DE2	-	-	-

Table 1.16 Pin list (5 of 7)

BGA224	BGA224 without MIPI	LQFP176	LQFP176 without MIPI	Power, System, Clock, Debug, CAC	I/O ports	Ex. Bus, SDRAM	Ex. Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/ USBHS/OSPI/ SSIE/SDHI/MMC/ EHTERC(MII,RMII)	GPT/AGT/ ULPT/RTC	ADC12/ DAC12/ ACMPHS	MIPI, GLCDC, CEU
G4	G4	-	-	-	PA03	D25/DQ25	-	TXD2_C/MOSI2_C/SDA2_C	-	-	-
K6	K6	-	-	-	PA02	D26/DQ26	-	RXD2_C/MISO2_C/SCL2_C	-	-	-
J7	J7	105	105	VSS	-	-	-	-	-	-	-
J6	J6	106	106	VCC	-	-	-	-	-	-	-
H3	H3	107	107	-	PA01	RD	-	-	-	-	LCD_TCON0_A
H4	H4	108	108	-	PA00	D7[A7/D7]/DQ7	-	-	-	-	LCD_CLK_A
F1	F1	109	109	VCL	-	-	-	-	-	-	-
J4	J4	110	110	-	P607	D6[A6/D6]/DQ6	-	-	-	-	LCD_EXTCLK_A
G1	G1	111	111	VCC_DCDC	-	-	-	-	-	-	-
G2	G2	112	112	VCC_DCDC	-	-	-	-	-	-	-
H1	H1	113	113	VLO	-	-	-	-	-	-	-
H2	H2	114	114	VLO	-	-	-	-	-	-	-
J1	J1	115	115	VSS_DCDC	-	-	-	-	-	-	-
M5	M5	116	116	-	P606	D5[A5/D5]/DQ5	-	-	-	-	-
K5	K5	117	117	-	P605	D4[A4/D4]/DQ4	-	CTS0_B	GTIOC8A	-	-
N4	N4	118	118	-	P604	D3[A3/D3]/DQ3	-	CTS0_RTS0_B/SS0_B/DE0	GTIOC8B	-	-
M4	M4	119	119	-	P603	D2[A2/D2]/DQ2	-	TXD0_B/MOSI0_B/SDA0_B	GTIOC7A/ ULPTO0	-	-
L4	L4	120	120	-	P602	D1[A1/D1]/DQ1	-	RXD0_B/MISO0_B/SCL0_B	GTIOC7B/ ULPTEE0	-	-
K4	K4	121	121	-	P601	D0[A0/D0]/DQ0	-	SCK0_B/DE0/OM_WP1	GTIOC6A/ ULPTEVI0/ RTCOUT	-	-
K3	K3	122	122	CACREF	P600	-	-	OM_RSTO1	GTIOC6B/ ULPTEVI1-DS	-	-
K1	K1	123	123	-	P107	-	-	OM_CS0	GTOWUP/ GTIOC8A/ AGTOA0	-	-
K2	K2	124	124	-	P106	-	-	SSLB3_A/OM_RESET	GTOWLO/ GTIOC8B/ AGTOB0/ ULPTEE1-DS	-	-
L3	L3	125	125	-	P105	-	IRQ0	SSLB2_A/OM_ECSINT1	GTIOC1A/ ULPTO1-DS	-	-
L1	L1	126	126	-	P104	-	IRQ1	CTS9_A/SSLB1_A/OM_CS1	GTETRGB/ GTIOC1B	-	-
L2	L2	127	127	-	P103	-	-	CTS9_RTS9_A/SS9_A/DE9/ SSLB0_A/CTX0/OM_SIO2	GTOWUP/ GTIOC2A	-	-
M1	M1	128	128	-	P102	-	-	TXD9_A/MOSI9_A/SDA9_A/ RSPCKB_A/CRX0/OM_SIO4	GTOWLO/ GTIOC2B/AGTO0	ADTRG0	-
M2	M2	129	129	-	P101	-	IRQ1	RXD9_A/MISO9_A/SCL9_A/ MOSIB_A/OM_SIO3	GTETRGB/ GTIOC8A/ AGTEE0	-	-
J3	J3	130	130	VCC2	-	-	-	-	-	-	-
J2	J2	131	131	VSS	-	-	-	-	-	-	-
N1	N1	132	132	-	P100	-	IRQ2	SCK9_A/DE9/MISOB_A/OM_SIO0	GTETRGA/ GTIOC8B/ AGTIO0	-	-
M3	M3	133	133	-	P800	-	IRQ11	CTS2_A/OM_SIO5	GTIU/GTIOC11A/ AGTOA0	-	-
N2	N2	134	134	-	P801	-	IRQ12	TXD2_A/MOSI2_A/SDA2_A/ OM_DQS	GTIV/GTIOC11B/ AGTOB0	-	-
P1	P1	135	135	-	P802	-	-	RXD2_A/MISO2_A/SCL2_A/ OM_SIO6	GTIW/GTIOC12A	-	-
N3	N3	136	136	-	P803	-	-	SCK2_A/DE2/OM_SIO1	GTETRGC/ GTIOC12B	-	-
P2	P2	137	137	-	P804	-	IRQ14	CTS2_RTS2_A/SS2_A/DE2/ OM_SIO7	GTETRGD/ GTIOC13A	-	-
R1	R1	138	138	-	P808	-	IRQ15	OM_SCLK	GTIOC13B	-	-
R2	R2	139	139	-	P809	-	-	OM_SCLKN	-	-	-

Table 1.16 Pin list (6 of 7)

BGA224	BGA224 without MIPI	LQFP176	LQFP176 without MIPI	Power, System, Clock, Debug, CAC	I/O ports	Ex. Bus, SDRAM	Ex. Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/ USBHS/OSPI/ SSIE/SDHI/MMC/ EHTERC(MII,RMII)	GPT/AGT/ ULPT/RTC	ADC12/ DAC12/ ACMPHS	MIPI, GLCDC, CEU
P3	P3	140	140	VCC2	-	-	-	-	-	-	-
R3	R3	141	141	VSS	-	-	-	-	-	-	-
P4	P4	142	142	-	P810	-	-	SD1CLK_A	ULPTOA0	-	-
M6	M6	143	143	-	P811	-	-	USB_ID/SD1CMD_A	ULPTOB0	-	-
R4	R4	144	144	-	P812	-	-	USB_EXICEN/SD1DAT0_A	-	AN122	-
P5	P5	145	145	CACREF	P500	-	-	USB_VBUSEN/SD1DAT1_A	-	AN121	-
R5	R5	146	146	-	P501	-	-	USB_OVRCURA/SD1DAT2_A	-	AN120	-
P6	P6	147	147	-	P502	-	-	USB_OVRCURB/SD1DAT3_A	-	AN019/AN119	-
K7	K7	-	-	-	P503	-	-	SD1CD	-	-	-
N5	N5	-	-	-	P504	-	-	SD1WP	-	-	-
K8	K8	-	-	-	P505	D27/DQ27	-	SD1DAT4_A	-	-	-
N6	N6	-	-	-	P506	D28/DQ28	-	SD1DAT5_A	-	-	-
L7	L7	-	-	-	P507	D29/DQ29	-	SD1DAT6_A	-	-	-
M7	M7	-	-	-	P508	D30/DQ30	IRQ1	SD1DAT7_A	-	-	-
L8	L8	-	-	-	P509	D31/DQ31	IRQ2	-	ULPTEVI1	-	-
N7	N7	-	-	-	P510	WR3/BC3	IRQ3	-	ULPTEVI0	-	-
R6	R6	148	148	VCL	-	-	-	-	-	-	-
J9	J9	149	149	VCC	-	-	-	-	-	-	-
J8	J8	150	150	VSS	-	-	-	-	-	-	-
R7	R7	151	151	-	P015	-	IRQ13	-	-	AN105/DA1	-
P7	P7	152	152	-	P014	-	-	-	-	AN007/DA0	-
P8	P8	153	153	VREFL	-	-	-	-	-	-	-
R8	R8	154	154	VREFH	-	-	-	-	-	-	-
N8	N8	155	155	AVCC0	-	-	-	-	-	-	-
N9	N9	156	156	AVSS0	-	-	-	-	-	-	-
P9	P9	157	157	VREFL0	-	-	-	-	-	-	-
R9	R9	158	158	VREFH0	-	-	-	-	-	-	-
M9	M9	-	-	-	P011	-	-	-	-	AN106	-
M8	M8	159	159	-	P010	-	IRQ14	-	-	AN005/ IVCMP0	-
L9	L9	160	160	-	P009	-	IRQ13-DS	-	-	AN006	-
R10	R10	161	161	-	P008	-	IRQ12-DS	-	-	AN008	-
M10	M10	162	162	-	P007	-	-	-	-	AN004	-
R11	R11	163	163	-	P006	-	IRQ11-DS	-	-	AN002/ IVCMP3	-
N10	N10	164	164	-	P005	-	IRQ10-DS	-	-	AN001	-
P10	P10	165	165	-	P004	-	IRQ9-DS	-	-	AN000/ IVCMP2	-
P11	P11	166	166	-	P003	-	-	-	-	AN104/ IVREF1	-
R12	R12	167	167	-	P002	-	IRQ8-DS	-	-	AN102/ IVCMP3	-
P12	P12	168	168	-	P001	-	IRQ7-DS	-	-	AN101/ IVREF0	-
R13	R13	169	169	-	P000	-	IRQ6-DS	-	-	AN100/ IVCMP2	-
H6	H6	-	-	VCL	-	-	-	-	-	-	-
H8	H8	170	170	VSS	-	-	-	-	-	-	-
H9	H9	171	171	VCC	-	-	-	-	-	-	-
N11	N11	172	172	-	P806	-	IRQ0	-	-	AN018/AN118	LCD_CLK_B/ VIO_D14
M11	M11	173	173	-	P805	-	-	-	-	AN017/AN117	LCD_TCON0_B/ VIO_D15

Table 1.16 Pin list (7 of 7)

BGA224	BGA224 without MIPI	LQFP176	LQFP176 without MIPI	Power, System, Clock, Debug, CAC	I/O ports	Ex. Bus, SDRAM	Ex. Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/ USBHS/OSPI/ SSIE/SDHI/MMC/ EHTERC(MII,RMII)	GPT/AGT/ ULPT/RTC	ADC12/ DAC12/ ACMPHS	MIPI, GLCDC, CEU
N12	N12	-	-	-	P807	-	-	-	-	-	LCD_TCON1_B
P13	P13	174	174	-	P513	-	-	-	-	AN016/ AN116/ IVCMP0	LCD_TCON2_B/ VIO_FLD
R14	R14	-	-	-	P515	-	-	-	-	-	LCD_TCON3_B
N13	N13	175	175	-	P512	-	IRQ14	SCL1_A/CTX1	GTIOC0A	-	-
P14	P14	-	-	-	P514	-	-	-	-	-	LCD_EXTCLK_B
K9	K9	176	176	-	P511	-	IRQ15	SDA1_A/CRX1	GTIOC0B	-	-

Note: Several pin names have the added suffix of _A, _B, and _C. These suffixes have special conditions for electrical characteristics. See section 60, Electrical Characteristics for detail.

2. CPU

The MCU is based on the Arm® Cortex®-M85 Processor.

2.1 Overview

2.1.1 CPU

- Arm® Cortex®-M85 Processor
  - Revision: r0p2-00rel0
  - ARMv8.1-M architecture profile
  - Armv8-M Security Extension
- Extension Processing Unit (EPU)
  - Floating Point Unit (FPU) compliant with the ANSI/IEEE Std 754-2008
    Scalar half, single, and double-precision floating-point operation
  - M-profile Vector Extension (MVE)
    Integer, half-precision, and single-precision floating-point MVE (MVE-F)
- Security Attribution Unit (SAU): 8 regions
- Implementation Defined Attribution Unit (IDAU)
- Memory Protection Unit (MPU)
  - Protected Memory System Architecture (PMSAv8)
  - Secure MPU (MPU_S): 8 regions
  - Non-secure MPU (MPU_NS): 8 regions
- System Timer (SysTick)
  - Reference clock:
    - CPUCLK
    - SYSTICKCLK
  - Two system timers are implemented:
    - Secure instance (SysTick_S)
    - Non-secure instance (SysTick_NS)
- Low power modes
  - CPU Sleep mode
  - CPU Deep Sleep mode
- Cache
  - Instruction cache: 16 KB with ECC
  - Data cache: 16 KB with ECC
- Tightly Coupled Memory (TCM)
  - ITCM: 64 KB with ECC (8 KB × 8 block)
  - DTCM: 64 KB with ECC (8 KB × 8 block)
- Nested Vectored Interrupt Controller (NVIC): 96 IRQs

See section 2.14. References [1], [2] for the detail.

2.1.2 Debug

CPU Debug Components

- Arm® CoreSight™ ETM-M85
  - Revision: r0p2-00rel0
  - ARM ETM Architecture version 4.5
- Instrumentation Trace Macrocell (ITM)
- Data Watchpoint and Trace unit (DWT)
  - 8 Data Watchpoint and Trace (DWT) comparators, up to 2 data value comparison
- Breakpoint Unit (BPU)
  - 8 instruction comparators
- Cortex®-M85 Trace Port Interface Unit (TPIU)
  - 4 bits TPIU Formatter output
  - Serial Wire Output (SWO)

Common Debug Components

- Debug Access Port (DAP)
  - CoreSight SoC-400 Debug Access Port
  - JTAG Debug Port and Serial Wire Debug Port (SWJ-DP)
    Serial Wire protocol version 2 (target instance is fixed value: 0b0000)
- Timestamp Generator (TSG)
  - CoreSight SoC-400 Timestamp Generator
  - Global Timestamp for ETM and ITM
  - Driven by DCLK
- Cross Trigger Interface (CTI)
  - CoreSight SoC-400 CTI (CTI)
  - Processor CTI (CTI0)
- Embedded Trace Buffer (ETB)
  - CoreSight™ Trace Memory Controller with ETB configuration
  - Buffer size: 8 KB
- Debug Register Module (DBGREG)
- On-Chip Debug Register Module (OCDREG)
- Authentication method through JTAG/SWD: Challenge response

See section 2.14. References [1], [2] for the detail.

2.2 Block Diagram

Figure 2.1 shows the block diagram of CPU and CoreSight Debug subsystem (DBGSS).

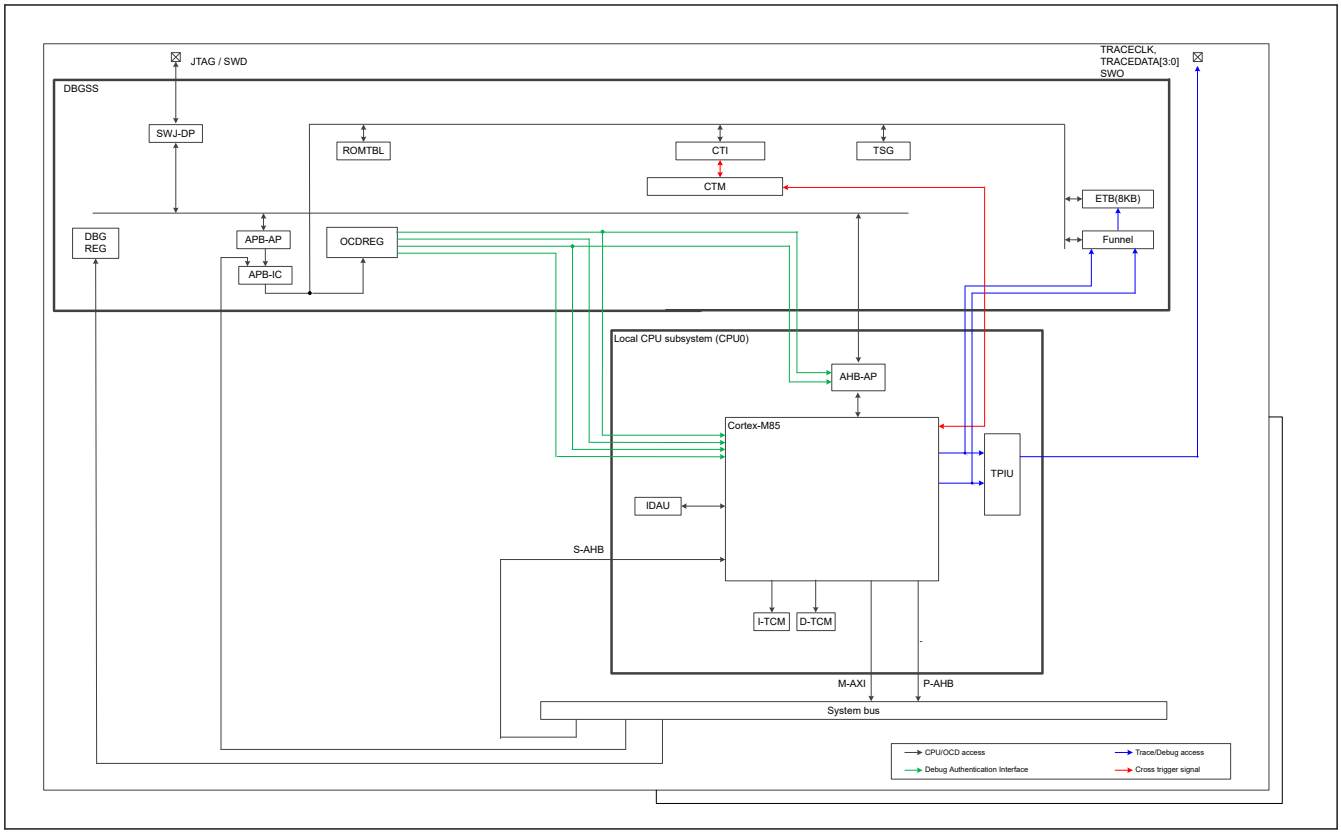


Figure 2.1 Block diagram of CPU and CoreSight Debug subsystem (DBGSS)

Note: See section 2.14. References [2] for detail of Cortex-M85 block diagram. Cortex-M85 block diagram described in this document is just for illustrative purposes only.

2.3 Implementation Options

Table 2.1 shows the implementation options of the MCU.

Table 2.1 Implementation options (1 of 2)

Option	Implementation
SAU	8 regions
IDAU	Included
MPU	MPU_S: 8 regions MPU_NS: 8 regions
MVE	Integer and half and single-precision floating-point MVE
FPU	Scalar half, single, and double-precision floating-point
Interrupts	96
Priority bits	4 bits (16 levels)
Sleep mode power-saving	CPU Sleep mode, CPU Deep Sleep mode
TCM	• ITCM : 64 KB • DTCM : 64 KB • ECC enable (determined by OFS1(_SEC).INITECCEN) • ITCMCR.EN = 1 and DTCMCR.EN = 1, cannot be rewrite
CACHE	• I-Cache: 16 KB • D-Cache: 16 KB • ECC enable (determined by OFS1(_SEC).INITECCEN) • Automatic invalidation is enabled During debug, it is controlled by CACHEDBGCR L1RSTDIS

Table 2.1 Implementation options (2 of 2)

Option	Implementation
P-AHB	• PAHBCR.SZ is 0b011 (256 MB) • PAHBCR.EN is always 1
SysTick	Included two system timers, SysTick_S and SysTick_NS External reference clock with frequency = SYSTICKCLK SYST_CALIB = 0x0000270F NOREF, bit[31] = 0: Reference clock is implemented SKEW, bit[30] = 0: TENMS calibration value is exact Bit[29:24] = 0x00: Reserved TENMS, bit[23:0] = 0x00270F: (0x270F + 1) × 1 µs = 10 ms
PACBTI	Included and Enabled
Software reset request output	AIRCR.SYSRESETREQ bit causes the Software reset
DWT	8 comparators
BPU	8 comparators
CTI	Included, CTI and CTI0
ITM	Included
ETM	Included
TPIU	Included • 4-bit TPIU formatter output • Serial Wire Output (SWO)

Table 2.2 DBGSS implementation options

Option	Implementation
TSG	Included Driven by clock with frequency = ICLK/2
ETB	8 KB
DAP	SWJ-DP AHB-AP : AP0 APB-AP : AP1

2.4 Trace Interface

A Trace Port Interface Unit (TPIU) and Serial Wire Output (SWO) provide trace output. Table 2.3 shows the MCU pins for the function. These pins are multiplexed with other functions.

Table 2.3 Trace function pins

Name	I/O	Width	Function	When not in use
TCLK	Out	1 bit	Trace clock	Open
TDATA0	Out	1 bit	Trace data output	Open
TDATA1	Out	1 bit	Trace data output	Open
TDATA2	Out	1 bit	Trace data output	Open
TDATA3	Out	1 bit	Trace data output	Open
SWO	Out	1 bit	Serial Wire Output	Open

2.5 JTAG/SWD Interface

Table 2.4 shows JTAG/SWD pins.

Table 2.4 JTAG/SWD pins (1 of 2)

Name	I/O	P/N	Width	Function	When not in use
TCK/SWCLK	In	Pos.	1 bit	JTAG Clock pin Serial Wire Data Clock pin	Pull-up

Table 2.4 JTAG/SWD pins (2 of 2)

Name	I/O	P/N	Width	Function	When not in use
TMS/SWDIO	In-Out	Neg.	1 bit	JTAG TMS pin Serial Wire Data In-Out pin	Pull-up
TDI	In	Pos.	1 bit	JTAG TDI pin	Pull-up
TDO	Out	Neg.	1 bit	JTAG TDO pin	Open

2.6 Initial Vector Table Base Address

In the MCU, the secure vector table base address is determined by the operating mode as shown in Table 2.5, and the non-secure vector table base address is fixed at 0x0000_0000.

Table 2.5 Initial vector table base address

Operating mode	CPU INITSVTOR
Single-chip mode	0x0200_0000

2.7 Debug Function

The MCU provides ability to debug software by either external debug or self-hosted debug.

The external debug function can be enabled by JTAG/SWD. See section 2.7.1. External Debugger Connectivity and Authentication Through JTAG/SWD.

Self-hosted debug function can be enabled by software programming. See section 2.7.2. Debug Protection Mechanism by Software for details.

When debug function is enabled, operation of system can be affected. See section 2.7.3. Effect of Debug Function for details.

2.7.1 External Debugger Connectivity and Authentication Through JTAG/SWD

When debug function is enabled by performing authentication through JTAG/SWD, the three debug levels that correspond to the three Authentication Level (AL) are AL0, AL1, and AL2.

- AL0: No debug functions[*1] are available.
- AL1: Only non-secure debug functions[*1] is enabled, and debugger can access only defined Non-secure debug accessible regions. It is defined as Non-secure debug in section 2.14. References [1].
- AL2: Non-secure and Secure debug functions[*1] are enabled and accessible from the debugger. It is defined as Secure debug in section 2.14. References [1].

Authentication Level is determined as follows:

- When Device Life Cycle state (DLM state) is not OEM (CM, LCK_BOOT, RMA_RET, RMA_REQ, RMA_ACK), AL is determined by DLM state. See Table 2.6.
- When Device Life Cycle state (DLM state) is equal to OEM, AL can be determined by performing authentication through JTAG/SWD. In the MCU, authentication method is the challenge response method with the authentication key installed by user.
  In this case, initial value of AL is determined by PL value (see Table 2.7).
  See section 2.13.5.2. Connecting Sequence and JTAG/SWD Authentication for authentication sequence.

In addition, using authentication key AL2 and AL1 can be prohibited by setting Protection LCKS and LCKNS respectively. Setting LCKS and LCKNS can be performed by the boot command.

Note 1. Both of invasive and non-invasive debug. See section 2.14. References [1] for details about invasive and non-invasive debug.

Description in Table 2.6 and Table 2.7 can also be used for serial Flash programmer.

Table 2.6 Authentication Level (1)

Conditions			
DLM state	**LCKS**	**LCKNS**	**Available Authentication Level**
CM	Don't care	Don't care	AL2 (default)
LCK_BOOT	Don't care	Don't care	AL0 (default)
RMA_RET	Don't care	Don't care	AL0 (default)
RMA_REQ	Don't care	Don't care	AL0 (default)
RMA_ACK	Don't care	Don't care	AL2 (default)

Table 2.7 Authentication Level (2)

Condition				
DLM state	**PL**	**LCKS**	**LCKNS**	**Available Authentication Level**
OEM	PL2	Don't care	Don't care	AL2 (default)
OEM	PL1	0b111	Don't care	AL1 (default) AL2 (authentication using AL2_KEY)
OEM	PL1	0b000 (using AL2_KEY is prohibited)	Don't care	AL1 (default)
OEM	PL0	0b111	0b111	AL0 (default) AL1 (authentication using AL1_KEY) AL2 (authentication using AL2_KEY)
OEM	PL0	0b111	0b000 (using AL1_KEY is prohibited)	AL0 (default) AL2 (authentication using AL2_KEY)
OEM	PL0	0b000 (using AL2_KEY is prohibited)	0b111	AL0 (default) AL1 (authentication using AL1_KEY)
OEM	PL0	0b000 (using AL2_KEY is prohibited)	0b000 (using AL1_KEY is prohibited)	AL0 (default)

Note: When debugging using SWJ-DP, set CDBGPWRUPREQ to 1.
Note: Debug connection from external emulator to CoreSight components through Debug APB is enabled as default setting, don't care is the authentication mechanism. However, this connection can be disabled by software programming. See DBGAUTH0.DEVICEEN for details.

Debug connection while in Boot mode

Debug capability is not available while the MCU is operating in Boot mode.

2.7.2 Debug Protection Mechanism by Software

In this mechanism, the challenge and response authentication procedure does not perform, instead software can program to directly enable debug function of the MCU. In other words, debug function is controlled completely by software.

The debug function can be enabled by setting the following bits:

- DBGAUTH0.NIDEN0 = 1: Non-invasive debug enable
- DBGAUTH0.DBGEN0 = 1: Invasive debug enable
- DAUTHCTRL.INTSPNIDEN = 1: Secure non-invasive debug enable. See Reference [1] for details
- DAUTHCTRL.INTSPIDEN = 1: Secure invasive debug enable. See Reference [1] for details.

In addition, external debug access to processor through AHB-AP can also be enabled by writing 1 to DBGAUTH0.DBGENAP and DBGAUTH1.SPIDENAP for Non-secure and Secure access respectively.

See DBGAUTH0 and DBGAUTH1 in section 2.8.5.3. DBGSTOPCR : Debug Stop Control Register and section 2.8.5.4. DBGAUTH0 : Debug Authentication Control Register 0 for more details.

Note: Authentication through software programming is controlled by OFS1(_SEC).SWDBG.

Writing 1 to OFS1(_SEC).SWDBG disables authentication through software programming, writing 0 to OFS1(_SEC).SWDBG enables authentication through software programming.

Note: Authentication through software programming is not restricted by DLM state, PL, and LCK(N)S.

Note: Except DAUTHCTRL.INTSPNIDEN and DAUTHCTRL.INTSPIDEN overrides processor external Secure invasive/ non-invasive debug authentication interface, debug function enabled by authentication through JTAG/SWD cannot be disabled by software.

2.7.3 Effect of Debug Function

The debug function affects the inside and outside of CPU. This section describes the effects of debug function excluding debug authentication.

2.7.3.1 Low power mode

All CoreSight debug components can store the register settings even when the CPU enters Deep Sleep, Software Standby, or Deep Software Standby mode if SYOCDCR.DBGEN0 is 1 and any of the following is true:

- CDBGPWRUPREQ = 1 and AL is AL2 or AL1
- OFS1(_SEC).SWDBG is 0 and DBGAUTH0.DBGEN0 or DBGAUTH0.NIDEN0 is 1.

However, AHB-AP cannot respond to On-Chip Debug (OCD) access in these low power modes. The OCD must wait for cancellation of the low power mode to access the CoreSight debug components.

The MCU does not response to On-Chip Debug (OCD) when the OCD starts to connect to the MCU while the MCU is in Software Standby mode or Deep Software Standby mode.

2.7.3.2 Resets and interrupts

During On-Chip Debug (OCD) mode, when the CPU enters the break state, IWDT and corresponding WDT are stopped. During OCD mode, the CPU is running the application and debug function AL2 or AL1 is enabled, some resets and interrupts are restrained depending on DBGSTOPCR setting such as PVDn, WDT, and IWDT. For PVDn (n = 1, 2), interrupt and reset depend on whether DBGSTOPCR is in the break state or not. Some access errors are not generated by external debugger (DAP) access. For details, see Table 2.8.

Table 2.8 Controlling of resets and interrupts generation in OCD mode

Reset and interrupt name	Break state	Running application
RES pin reset	Same as user mode	
Power-on reset	Same as user mode	
Independent watchdog timer reset/interrupt*2	Does not occur*1	Depends on DBGSTOPCR setting*2
Watchdog timer reset/interrupt	Does not occur*1	Depends on DBGSTOPCR setting*2
CPU lockup reset/interrupt	Does not occur	Same as user mode
Voltage monitor 0 reset	Same as user mode	
Voltage monitor 1 reset/interrupt	Depends on DBGSTOPCR setting*3	
Voltage monitor 2 reset/interrupt	Depends on DBGSTOPCR setting*3	
Cache/TCM ECC error BusFaults	Same as user mode	
SRAM Parity/ECC error reset/interrupt	Same as user mode*3	
Deep software standby reset	Same as user mode	
Software reset	Same as user mode	
Bus error reset/interrupt	Same as user mode*4	

Note 1. IWDT and WDT always stop in this mode if any of the following is true:
- CDBGPWRUPREQ is 1, and AL is AL2 or AL1
- OFS1(_SEC).SWDBG is 0, and DBGAUTH0.DBGEN0 is 1

Note 2. IWDT and WDT operation depends on DBGSTOPCR setting if any of the following is true:
- CDBGPWRUPREQ is 1, and AL is AL2 or AL1
- OFS1(_SEC).SWDBG is 0, and DBGAUTH0.DBGEN0 is 1

Note 3. Reset or interrupt masking depends on DBGSTOPCR setting.
Note 4. The reset request or interrupt request is not generated when an error is detected for DAP access.

2.8 Registers Description

2.8.1 CPU Control Registers

In addition to Cortex-M85 registers, the MCU also provides additional specific control and status registers that can be considered as CPU control register.

2.8.1.1 CPUSAR : CPU Security Attribution Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x170

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPUSA0	CPU Control Registers Security Attribution 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

CPUSA0 bit (CPU Control Registers Security Attribution)

The CPUSA0 bit is Security attributes for CPU Control registers. The target registers are:

- CPU.CPULCKUPCR
- CPU.CPUCRPT

2.8.1.2 CPULCKUPCR : CPU Lockup Control Register

Base address: CPU_CTRL = 0x4000_F000
CPU_CTRL_NS = 0x5000_F000

Offset address: 0x030

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OAD	Operation after detection of CPU lockup 0: Non-maskable Interrupt 1: CPU Lockup reset	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3 (CPUSAR.CPUSA0), P-TYPE-2

When CPU enters the lockup state, it can be selected to generate either a CPU Lockup reset or an interrupt[*1] by writing to CPULCKUPCR.OAD.

The CPULCKUPCR security attribution is controlled by CPUSAR. The AIRCR.BFHFNMINS and CPUSAR.CPUSA0 bits are recommended to be set to the same security attribution.

Writing to CPULCKUPCR.OAD bit is protected by CPU Control Register Protect Register (CPUCRPT).

Note 1. If lockup state occurs from a Secure HardFault when AIRCR.BFHFNMINS is set to 1 or the NMI handler, an NMI does not cause the processor exit lockup state. See section 2.14. References [4].

2.8.1.3 CPUCRPT : CPU Control Register Protection Register

Base address: CPU_CTRL = 0x4000_F000 (Secure)
CPU_CTRL_NS = 0x5000_F000 (Non-secure)

Offset address: 0x840

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Writing to CPULCKUPCR register is permitted. 1: Writing to CPULCKUPCR register is not permitted. Read access is permitted.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 in KEY[7:0] bits at the same time. When values other than 0xA5 are written to KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.	W

Note: S-TYPE-3 (CPUSAR.CPUSA0), P-TYPE-2

The CPUCRPT.PROTECT bit enables writing to CPULCKUPCR.

The security attribution of CPUCRPT is controlled by CPUSAR.

2.8.1.4 CPULOCKCR : CPU Function Lock Control Register

Base address: CPU_CTRL = 0x4000_F000 (Secure)

Offset address: 0x400

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	LCKDCAIC	LCKDTGU	LCKITGU	LCKSAU	LCKSMPU	LCKSVTAIR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LCKSVTAIR	Disable writes to the following secure registers from software or from a debug agent that is connected to the processor: **VTOR_S**, **AIRCR.PRIS**, **AIRCR.BFHFNMINS**	R/W*1
1	LCKSMPU	Disable writes to registers that are associated with the Secure MPU region from software or from a debug agent connected to the processor: **MPU_CTRL**, **MPU_RNR**, **MPU_RBAR**, **MPU_RLAR**, **MPU_RBAR_An**, **MPU_RLAR_An**	R/W*1
2	LCKSAU	Disable writes to registers that are associated with the SAU region from software or from a debug agent connected to the processor: **SAU_CTRL**, **SAU_RNR**, **SAU_RBAR**, **SAU_RLAR**	R/W*1
3	LCKITGU	Disable writes to registers that are associated with the ITCM interface security gating from software or from a debug agent connected to the processor: **ITGUCTRL**, **ITGU_LUTn**	R/W*1

Bit	Symbol	Function	R/W
4	LCKDTGU	Disable writes to registers that are associated with the DTCM interface security gating from software or from a debug agent connected to the processor: **DTGUCTRL**, **DTGU_LUTn**	R/W*1
5	LCKDCAIC	Disable access to the instruction cache direct cache access registers **DCAICLR** and **DCAICRR**	R/W*1
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6, P-TYPE-2
Note 1. Writing 0 is ignored.

This register controls some miscellaneous processor signals. Reset is the only clear condition.

2.8.1.5 CPULOCKCRNS : CPU Non-secure Function Lock Control Register

Base address CPU_CTRL_NS = 0x5000_F000 (Non-secure)

Offset address: 0x500

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	LCKNSMPU	LCKNSVTOR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LCKNSVTOR	Disable writes to the **VTOR_NS** register	R/W*1
1	LCKNSMPU	Disable writes to registers that are associated with the Non-secure MPU region from software or from a debug agent connected to the processor: **MPU_CTRL_NS**, **MPU_RNR_NS**, **MPU_RBAR_NS**, **MPU_RLAR_NS**, **MPU_RBAR_A_NSn**, **MPU_RLAR_A_NSn**	R/W*1
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-7, P-TYPE-2
Note 1. Writing 0 is ignored.

This register controls some miscellaneous processor signals. Reset is the only clear condition.

2.8.2 Address Spaces for External Debugger

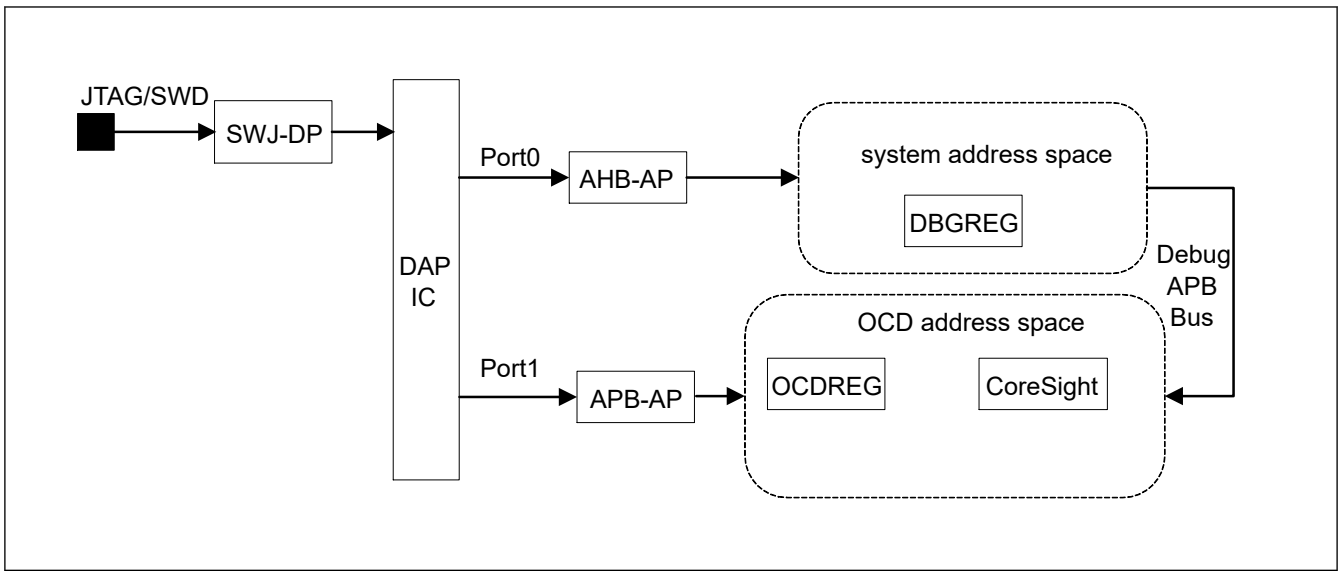


Figure 2.2 Block diagram of the AP connection and address spaces

The MCU debug system includes two CoreSight Access Ports (AP):

- AHB-AP, which is connected to the CPU bus matrix and has the same access to the system address space as the CPU
- APB-AP, which has OCD address space and is connected to the Coresight components and OCDREG registers.

Figure 2.2 shows a block diagram of the AP connection and address spaces. For debug purpose, there are two address spaces, DBGREG and OCDREG.

DBGREG is located in the system address space and can be accessed from an OCD emulator, a CPU, or other bus masters in the MCU.

OCDREG is located in the OCD address space and can be accessed from an OCD emulator, a CPU, or other bus masters in the MCU. See Table 2.10 for details.

2.8.3 CPU Peripheral

CoreSight components are accessible from either CPU or OCD emulator. Table 2.9 shows dedicated Peripheral address of CPU. Table 2.10, Table 2.11, and Table 2.12 describe CoreSight component address.

Table 2.9 CPU peripherals

Component name	Start address	End address	Note
ITM	0xE000_0000	0xE000_0FFF	See section 2.14. References [2]
DWT	0xE000_1000	0xE000_1FFF	See section 2.14. References [2]
BPU	0xE000_2000	0xE000_2FFF	See section 2.14. References [2]
PMU	0xE000_3000	0xE000_3FFF	See section 2.14. References [2]
SCS	0xE000_E000	0xE000_EFFF	See section 2.14. References [2]
SCS Non-secure alias	0xE002_E000	0xE002_EFFF	See section 2.14. References [2]
TPIU*1	0xE004_0000	0xE004_0FFF	See section 2.14. References [3]
ETM	0xE004_1000	0xE004_1FFF	See section 2.14. References [2]
CTI0	0xE004_2000	0xE004_2FFF	See section 2.14. References [5]
EPPB ROM Table	0xE00F_E000	0xE00F_EFFF	See section 2.14. References [3]
Processor ROM Table	0xE00F_F000	0xE00F_FFFF	See section 2.14. References [3]

Note 1. TPIU registers should not be accessed without connection of OCD emulator to avoid stopping bus access.

Table 2.10 CPU peripherals (Secure CPU access view)

Component name	Start address	End address	Note
System ROM Table	0x4001_0000	0x4001_0FFF	See section 2.14. References [3]
OCDREG	0x4001_1000	0x4001_1FFF	See section 2.14. References [5]
CTI	0x4001_2000	0x4001_2FFF	See section 2.14. References [5]
Funnel	0x4001_3000	0x4001_3FFF	See section 2.14. References [5]
TMC(ETB)	0x4001_4000	0x4001_4FFF	See section 2.14. References [5]
Time Stamp Generator	0x4001_5000	0x4001_5FFF	See section 2.14. References [5]

Note: These registers can only be accessed from CPU when DbgSwEnable of APB Control/Status Word register is 1. See section 2.14. References [5].

Table 2.11 CPU peripherals (Non-secure CPU access view)

Component name	Start address	End address	Note
System ROM Table	0x5001_0000	0x5001_0FFF	See section 2.14. References [3]
OCDREG	0x5001_1000	0x5001_1FFF	See section 2.14. References [5]
CTI	0x5001_2000	0x5001_2FFF	See section 2.14. References [5]
Funnel	0x5001_3000	0x5001_3FFF	See section 2.14. References [5]
TMC(ETB)	0x5001_4000	0x5001_4FFF	See section 2.14. References [5]
Time Stamp Generator	0x5001_5000	0x5001_5FFF	See section 2.14. References [5]

Note: These registers can only be accessed from CPU when DbgSwEnable of APB Control/Status Word register is 1. See section 2.14. References [5].

Table 2.12 CPU peripherals (OCD emulator access view)

Component name	Start address	End address	Note
System ROM Table	0x8001_0000	0x8001_0FFF	See section 2.14. References [3]
OCDREG	0x8001_1000	0x8001_1FFF	See section 2.14. References [5]
CTI	0x8001_2000	0x8001_2FFF	See section 2.14. References [5]
Funnel	0x8001_3000	0x8001_3FFF	See section 2.14. References [5]
TMC(ETB)	0x8001_4000	0x8001_4FFF	See section 2.14. References [5]
Time Stamp Generator	0x8001_5000	0x8001_5FFF	See section 2.14. References [5]

2.8.4 ROM Table

The MCU has three ROM Tables:

- A processor ROM Table
- An EPPB ROM Table for the Cortex-M85 CPU
- A system ROM Table

The EPPB ROM Table points to the TPIU component and processor ROM Table. The EPPB ROM Table does not point to or be pointed by system ROM Table.

The processor ROM Table contains entries that hold a list of debug components that are implemented in Cortex-M85. The system ROM Table contains entries that hold a list of debug components implemented as CoreSight debug components of the MCU.

2.8.4.1 ROM entries

ROM entries hold a list of CoreSight debug components in the system. The OCD emulator can use the ROM entries to determine which components are implemented in the system.

The entries of Debug components are described as bellow.

Table 2.13: System ROM entries

Table 2.14: EPPB ROM entries

Table 2.15: Processor ROM entries (see section 2.14. References).

Table 2.13 System ROM entries

#	Address*1	Access size	R/W	Value	Target module
0	0x4001_0000	32 bits	Read-only	0x00001003	OCDREG
1	0x4001_0004	32 bits	Read-only	0x00002003	CTI
2	0x4001_0008	32 bits	Read-only	0x00003003	Funnel
3	0x4001_000C	32 bits	Read-only	0x00004003	ETB
4	0x4001_0010	32 bits	Read-only	0x00005003	TSG
5	0x4001_0014	32 bits	Read-only	0x00000000	(End of entries)

Note 1. The base address 0x4001_0000 is dedicated for CPU secure access.
The base address for CPU non-secure access is 0x5001_0000.
The base address for OCD emulator access is 0x8001_0000.

Table 2.14 EPPB ROM entries (1 of 2)

#	Address	Access size	R/W	Value	Target module
0	0xE00F_E000	32 bits	Read-only	0x00001003	Processor ROM Table
1	0xE00F_E004	32 bits	Read-only	0xFFF42003	TPIU

Table 2.14 EPPB ROM entries (2 of 2)

#	Address	Access size	R/W	Value	Target module
2	0xE00F_E008	32 bits	Read-only	0x00000000	(End of entries)

Table 2.15 Processor ROM entries

#	Address	Access size	R/W	Value	Target module
0	0xE00F_F000	32 bits	Read-only	0xFFF0F003	SCS
1	0xE00F_F004	32 bits	Read-only	0xFFF02003	DWT
2	0xE00F_F008	32 bits	Read-only	0xFFF03003	BPU
3	0xE00F_F00C	32 bits	Read-only	0xFFF01003	ITM
5	0xE00F_F014	32 bits	Read-only	0xFFF42003	ETM
6	0xE00F_F018	32 bits	Read-only	0xFFF04003	PMU
7	0xE00F_F01C	32 bits	Read-only	0xFFF43003	CTI0
8	0xE00F_F020	32 bits	Read-only	0xFFF47002	PMC
9	0xE00F_F024	32 bits	Read-only	0x00000000	(End of entries)

2.8.4.2 CoreSight Component Registers

ROM Table has CoreSight Component Register defined in Arm CoreSight Architecture.

Table 2.16 shows the registers of system ROM Table.

Table 2.17 shows processor EPPB ROM Table.

Table 2.18 shows processor ROM Table. See section 2.14. References [2].

Table 2.16 Registers of system ROM Table

Name	Address	Access size	R/W	Initial value
PID4	0x8001_0FD0	32 bits	Read-only	0x00000004
PID5	0x8001_0FD4	32 bits	Read-only	0x00000000
PID6	0x8001_0FD8	32 bits	Read-only	0x00000000
PID7	0x8001_0FDC	32 bits	Read-only	0x00000000
PID0	0x8001_0FE0	32 bits	Read-only	0x0000003B
PID1	0x8001_0FE4	32 bits	Read-only	0x00000030
PID2	0x8001_0FE8	32 bits	Read-only	0x0000000A
PID3	0x8001_0FEC	32 bits	Read-only	0x00000000
CID0	0x8001_0FF0	32 bits	Read-only	0x0000000D
CID1	0x8001_0FF4	32 bits	Read-only	0x00000010
CID2	0x8001_0FF8	32 bits	Read-only	0x00000005
CID3	0x8001_0FFC	32 bits	Read-only	0x000000B1

Table 2.17 Processor EPPB ROM Table (1 of 2)

Name	Address	Access size	R/W	Initial value
PID4	0xE00F_EFD0	32 bits	Read-only	0x00000004
PID5	0xE00F_EFD4	32 bits	Read-only	0x00000000
PID6	0xE00F_EFD8	32 bits	Read-only	0x00000000
PID7	0xE00F_EFDC	32 bits	Read-only	0x00000000
PID0	0xE00F_EFE0	32 bits	Read-only	0x0000003B
PID1	0xE00F_EFE4	32 bits	Read-only	0x00000030

Table 2.17 Processor EPPB ROM Table (2 of 2)

Name	Address	Access size	R/W	Initial value
PID2	0xE00F_EFE8	32 bits	Read-only	0x0000000A
PID3	0xE00F_EFEC	32 bits	Read-only	0x00000000
CID0	0xE00F_EFF0	32 bits	Read-only	0x0000000D
CID1	0xE00F_EFF4	32 bits	Read-only	0x00000010
CID2	0xE00F_EFF8	32 bits	Read-only	0x00000005
CID3	0xE00F_EFFC	32 bits	Read-only	0x000000B1

Table 2.18 Processor ROM Table

Name	Address	Access size	R/W	Initial value
PID4	0xE00F_FFD0	32 bits	Read-only	0x00000004
PID5	0xE00F_FFD4	32 bits	Read-only	0x00000000
PID6	0xE00F_FFD8	32 bits	Read-only	0x00000000
PID7	0xE00F_FFDC	32 bits	Read-only	0x00000000
PID0	0xE00F_FFE0	32 bits	Read-only	0x000000D4
PID1	0xE00F_FFE4	32 bits	Read-only	0x000000B4
PID2	0xE00F_FFE8	32 bits	Read-only	0x0000000B
PID3	0xE00F_FFEC	32 bits	Read-only	0x00000000
CID0	0xE00F_FFF0	32 bits	Read-only	0x0000000D
CID1	0xE00F_FFF4	32 bits	Read-only	0x00000010
CID2	0xE00F_FFF8	32 bits	Read-only	0x00000005
CID3	0xE00F_FFFC	32 bits	Read-only	0x000000B1

2.8.5 DBGREG

Base_Address: CPU_DBG = 0x4001_B000

The DBGREG module controls the debug functionality.

Table 2.19 shows a list of DBGREG registers.

Table 2.19 DBGREG registers

Name	Symbol	DAP port	Address	Access size	R/W
Debug Status Register	DBGSTR	Port 0	0x4001_B000	32 bits	Read-only
Debug Stop Control Register	DBGSTOPCR	Port 0	0x4001_B010	32 bits	Read/Write
Debug Authentication Control Register0	DBGAUTH0	Port 0	0x4001_B020	32 bits	Read/Write
Debug Authentication Control Register1	DBGAUTH1	Port 0	0x4001_B024	32 bits	Read/Write
Trace Port Control Register	TRPORTCR	Port 0	0x4001_B030	32 bits	Read/Write
Cache Debug Control Register	CACHEDBGCR	Port 0	0x4001_B040	32 bits	Read/Write
MOCO Enable Register for Debug	DBGMOCOEN	Port 0	0x4001_B300	32 bits	Read/Write
Flash Sequencer Clock Select Register for Debug	DBGFCLKSEL	Port 0	0x4001_B310	32 bits	Read/Write

2.8.5.1 DEBUGSAR : Debug Security Attribution Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x180

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DBGSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DBGSA0	Debug Resources Security Attribution 0 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

DBGSA0 bit (Debug Resources Security Attribution 0)

Security attributes of registers for Debug Control. The target registers are as follow:

- DBGREG.DBGAUTH0
- The CoreSight registers connected to APB-AP and mapped to the system address space

Note: This does not affect access from external to APB-AP.

2.8.5.2 DBGSTR : Debug Status Register

Base address: CPU_DBG = 0x4001_B000
CPU_DBG_NS = 0x5001_B000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	CDBGPWRUPACK	CDBGPWRUPREQ	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	x	x	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
27:0	—	These bits are read as 0.	R
28	CDBGPWRUPREQ	Debug power-up request 0: OCD is not requesting debug power-up 1: OCD is requesting debug power-up	R
29	CDBGPWRUPACK	Debug power-up acknowledge 0: Debug power-up request is not acknowledged 1: Debug power-up request is acknowledged	R
31:30	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-2

The DBGSTR register is a debug status register to notify OCD status to programs. This register is monitoring Debug connecting state.

2.8.5.3 DBGSTOPCR : Debug Stop Control Register

Base address: CPU_DBG = 0x4001_B000
CPU_DBG_NS = 0x5001_B000

Offset address: 0x010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	DBGSTOP_RER	—	—	—	—	—	—	DBGSTOP_PVD	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DBGSTOP_WDT0	DBGSTOP_IWDT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	DBGSTOP_IWDT	Mask bit for IWDT reset/interrupt in the OCD run mode In the OCD break mode, the reset/interrupt is masked and IWDT counter is stopped, regardless of this bit value. 0: Enable IWDT reset/interrupt 1: Mask IWDT reset/interrupt and stop IWDT counter	R/W
1	DBGSTOP_WDT0	Mask bit for WDT reset/interrupt in the OCD run mode In the OCD break mode, the reset/interrupt is masked and WDT counter is stopped, regardless of this bit value. 0: Enable WDT reset/interrupt 1: Mask WDT reset/interrupt and stop WDT counter	R/W
16:2	—	These bits are read as 0. The write value should be 0.	R/W
17	DBGSTOP_PVD	Mask bit for PVDn (n = 1, 2) reset/interrupt 0: Enable PVDn (n = 1, 2) reset/interrupt 1: Mask PVDn (n = 1, 2) reset/interrupt	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
24	DBGSTOP_RER	Mask bit for SRAM parity/ECC error reset/interrupt 0: Enable SRAM parity/ECC error reset/interrupt 1: Mask SRAM parity/ECC error reset/interrupt	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5, P-TYPE-2

Debug Stop Control Register (DBGSTOPCR) specifies functional stop during debugging.

This register is valid only when any of the following conditions are true:

- CDBGPWRUPREQ is 1, and AL is AL2 or AL1
- OFS1(_SEC).SWDBG is 0, and DBGAUTH0.DBGEN0 is 1

When none of the above conditions are true, the MCU ignores this register setting and considers all bits to be 0.

2.8.5.4 DBGAUTH0 : Debug Authentication Control Register 0

Base address: CPU_DBG = 0x4001_B000
CPU_DBG_NS = 0x5001_B000

Offset address: 0x020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SWDBG	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEVICEEN
Value after reset:	OFS1(_SEC).SWDBG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	DBGENAP	—	—	—	NIDEN0	—	—	—	DBGEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DBGEN0*1	CPU invasive debug enable 0: Disabled 1: Enabled	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	NIDEN0	CPU non-invasive debug enable 0: Disabled 1: Enabled	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	DBGENAP*1	CPU AHB-AP (AP0) debug enable 0: Disabled 1: Enabled	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	DEVICEEN*2	APB-AP (AP1) authentication 0: Disabled 1: Enabled	R/W
30:17	—	These bits are read as 0. The write value should be 0.	R/W
31	SWDBG	Software control of debug function 0: Disabled 1: Enabled	R/W

Note: S-TYPE-3 (DEBUGSAR.DBGSA0), P-TYPE-2
Note 1. When AL2 or AL1, the MCU ignores this bit and considers as if it is set to 1.
Note 2. When AL2, the MCU ignores this bit and considers as if it is set to 1.

When software control of debug function (OFS1(_SEC).SWDBG=0) is enabled, non-secure invasive and non-invasive debug function of processor enabling can be controlled by DBGEN0 and NIDEN0 bits of the register.

Non-secure AHB-AP access can be enabled by DBGENAP bit of the register.

Access from APB-AP to Coresight components through Debug APB can be enabled by DEVICEEN bit of the register.

When OFS1(_SEC).SWDBG is 1, the MCU ignores this register and considers as if it is set to its initial value.

Note: MCU behaves according to DBGSTOPCR, SYOCDCR, MCUCTRL when DBGEN0 = 1, and also according to TRPORTCR when (DBGEN0 = 1 or NIDEN0 = 1) is true.

2.8.5.5 DBGAUTH1 : Debug Authentication Control Register 1

Base address: CPU_DBG = 0x4001_B000

Offset address: 0x024

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SPIDENAP	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	SPIDENAP*1	CPU AHB-AP (AP0) debug enable 0: Disabled 1: Enabled	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6, P-TYPE-2

Note 1. When AL2, the MCU ignores this bit and considers as if it is set to 1.

When software control of debug function (OFS1(_SEC).SWDBG = 0) is enabled, secure access from AHB-AP port to D-AHB can be enabled by SPIDENAP bit of the register.

When OFS1(_SEC).SWDBG is 1, the MCU ignores this register and considers as if it is set to its initial value.

2.8.5.6 CACHEDBGCR : Cache Debug Control Register

Base address: CPU_DBG = 0x4001_B000

Offset address: 0x040

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	L1RSTDIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	L1RSTDIS	Disable L1 cache automatic invalidation 0: Enable automatic invalidation of the L1 cache 1: Disable automatic invalidation of the L1 cache	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6, P-TYPE-2

The CACHEDBGCR register can control Disable L1 cache invalidation out of reset during debugging.

Note: The setting of the register is valid only if any of the following conditions apply:

- CDBGPWRUPREQ = 1 and AL is AL2 or AL1
- OFS1(_SEC).SWDBG is 0 and DBGAUTH0.DBGEN0 or DBGAUTH0.NIDEN0 is 1

Note: See section 4.6 Cache Initialization Signal in section 2.14. References [2] before writing to this bit.

2.8.5.7 TRPORTCR : Trace Port Control Register

Base address: CPU_DBG = 0x4001_B000
CPU_DBG_NS = 0x5001_B000

Offset address: 0x030

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	DRV[1:0]		—	OE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
0	OE	Data Out Enable bit indicates whether Trace Clock, Trace Data, and SWO outputs are enabled. 0: Output disabled 1: Output enabled	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
3:2	DRV[1:0]	Port Drive Capability Control indicates trace port buffer speed: 0 0: Low-drive 0 1: Middle-drive 1 0: High-speed high-drive 1 1: High-drive	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5, P-TYPE-2

The TRPORTCR register controls trace port of tracing data.

Note: The setting of the register is valid only if any of the following conditions apply:

- CDBGPWRUPREQ = 1, and AL is AL2 or AL1.
- OFS1(_SEC).SWDBG is 0, and DBGAUTH0.DBGEN0 or DBGAUTH0.NIDEN0 is 1.

2.8.5.8 DBGMOCOEN : MOCO Enable Request Register for Debug

Base address: CPU_DBG = 0x4001_B000
CPU_DBG_NS = 0x5001_B000

Offset address: 0x300

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOCOEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MOCOEN	MOCO enable request 0: No request MOCO enable 1: Request MOCO enable	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5, P-TYPE-2

The DBGMOCOEN register enables MOCO operation. When external debugger is not connected, the register does not have any effect. During P/E mode, the value of this register cannot be updated.

Note: Usage of the register:

When only non-secure debug function is enabled, the debugger cannot set clock-related registers that are protected by TrustZone. Therefore, writing to flash memory (by setting software breakpoint) is not available if the FCLK frequency is not at minimum required speed for programming flash operation.

In this case, by setting DBGMOCOEN and DBGFCLKSEL, the flash sequency clock can be changed to MOCO to enable writes to flash memory.

2.8.5.9 DBGFCLKSEL : Flash Sequencer Clock Select Register for Debug

Base address: CPU_DBG = 0x4001_B000
CPU_DBG_NS = 0x5001_B000

Offset address: 0x310

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FCLK SEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FCLKSEL	Flash sequencer clock select 0: FCLK 1: MOCO	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5, P-TYPE-2

The DBGFCLKSEL register switches the flash P/E sequencer clock. When external debugger is not connected, the register does not have any effect. During P/E mode, the value of this register cannot be updated.

Flash P/E is available with MOCO instead of FCLK even when FCLK is not in the range to support flash P/E. The following flow shows how to use MOCO instead of FCLK.

1. Wait until CPU is in debug state (DHCSR.S_HALT = 1).
2. Check that the flash is not in P/E mode (when flash is in P/E mode, flash P/E by the external host cannot be performed.)
3. Set DBGMOCOEN.MOCOEN to 1.
4. Wait for MOCO oscillation settling time.
5. Set DBGFCLKSEL.FCLKSEL to 1.
6. Perform flash P/E.
7. Clear DBGFCLKSEL.FCLKSEL to 0.
8. Clear DBGMOCOEN.MOCOEN to 0.
9. Restart the CPU.

Note: Usage of the register:

When only non-secure debug function is enabled, the debugger cannot set clock-related registers that are protected by TrustZone. Therefore, writing to flash memory (by setting software breakpoint) is not available if the FCLK frequency is not at minimum required speed for programming flash operation.

In this case, by setting DBGMOCOEN and DBGFCLKSEL, the flash sequency clock can be changed to MOCO to enable writes to flash memory.

2.8.5.10 CoreSight Component Registers

DBGREG has CoreSight Component Registers defined in ARM CoreSight Architecture. Table 2.20 shows the list of the registers.

Table 2.20 CoreSight component registers

Name	Address	Access size	R/W	Initial value
PID4	0x4001_BFD0	32 bits	Read-only	0x00000004
PID5	0x4001_BFD4	32 bits	Read-only	0x00000000
PID6	0x4001_BFD8	32 bits	Read-only	0x00000000
PID7	0x4001_BFDC	32 bits	Read-only	0x00000000
PID0	0x4001_BFE0	32 bits	Read-only	0x00000005
PID1	0x4001_BFE4	32 bits	Read-only	0x00000030
PID2	0x4001_BFE8	32 bits	Read-only	0x0000000A
PID3	0x4001_BFEC	32 bits	Read-only	0x00000000
CID0	0x4001_BFF0	32 bits	Read-only	0x0000000D
CID1	0x4001_BFF4	32 bits	Read-only	0x000000F0
CID2	0x4001_BFF8	32 bits	Read-only	0x00000005
CID3	0x4001_BFFC	32 bits	Read-only	0x000000B1

Note: S-TYPE-5, P-TYPE-2

2.8.6 OCDREG

OCDREG is a register module provided for OCD (On-Chip Debug) Emulator functionalities. OCDREG is implemented as a CoreSight compliant component. Table 2.21 shows register list of OCDREG.

Table 2.21 Register list of OCDREG (1 of 2)

Name	Symbol	DAP port	OCD Address	CPU Address	Access size	R/W
MCU Status Register	MCUSTAT	Port 1	0x8000_0400	Cannot be accessed from CPU	32 bits	Read only
MCU Control Register	MCUCTRL	Port 1	0x8001_1004	0x4001_1004 (Secure) 0x5001_1004 (Non-secure)	32 bits	Read/Write
JTAG Boot Mode Entry Register	JBMDR	Port 1	0x8001_1100	0x4001_1100 (Secure) 0x5001_1100 (Non-secure)	32 bits	Read/Write
JTAG Boot Receive Data Register	JBRDR	Port 1	0x8001_1120	0x4001_1120 (Secure) 0x5001_1120 (Non-secure)	32 bits	Read/Write
JTAG Boot Transmit Data Register	JBTDR	Port 1	0x8001_1130	0x4001_1130 (Secure) 0x5001_1130 (Non-secure)	32 bits	Read/Write
JTAG Boot Status Register	JBSTR	Port 1	0x8001_1140	0x4001_1140 (Secure) 0x5001_1140 (Non-secure)	32 bits	Read/Write
JTAG Boot Interrupt Control Register	JBICR	Port 1	0x8001_1150	0x4001_1150 (Secure) 0x5001_1150 (Non-secure)	32 bits	Read/Write

Table 2.21 Register list of OCDREG (2 of 2)

Name	Symbol	DAP port	OCD Address	CPU Address	Access size	R/W
First Stage Boot Loader Status Monitor Register	FSBLSTATM	Port 1	0x8001_1300	0x4001_1300 (Secure) 0x5001_1300 (Non-secure)	32 bits	Read only

Note: Accesses to OCDREG from CPU is only permitted by privileged access.
Accesses to OCDREG from external debug through APB-AP is permitted by either Privileged access or Unprivileged access.
Note: Do not write to MCUCTRL using JTAG/SWD while the CPU is accessing MCUCTRL, JBMDR, JBRDR, JBTDR or JBICR.
Do not write to JBMDR using JTAG/SWD except during RES pin reset.
Do not write to JBRDR using JTAG/SWD while the CPU is accessing MCUCTRL, JBMDR, JBRDR, JBTDR or JBICR.
Do not write to JBTDT using JTAG/SWD.
Do not write to JBSTR using JTAG/SWD.
Do not write to JBICR using JTAG/SWD.

2.8.6.1 MCUSTAT : MCU Status Register

Base address: 0x8000_0400*1

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	DSTBY3	DSTBY2	DSTBY1	—	—	—	STBY1
Value after reset:	0	0	0	0	0	0	0	0	0	x	x	x	0	0	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	AL[1:0]		—	—	—	—	—	—	—	—	—	CPUSTOPCLK	SLEEP	—
Value after reset:	1	0	0	0	0	0	0	1	0	0	0	0	0	x	x	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0.	R
1	SLEEP	Indicate CPU is in CPU Sleep mode 0: CPU is not in CPU Sleep mode 1: CPU is in CPU Sleep mode	R
2	CPUSTOPCLK	Indicate CPU clock is stopped 0: CPU clock is not stopped. It indicates that the MCU is in Normal or CPU Sleep mode. 1: CPU clock is stopped. It indicates that the MCU is in CPU Deep Sleep mode, Software Standby mode, or Deep Software Standby mode.	R
7:3	—	These bits are read as 0.	R
8	—	This bit is read as 1.	R
11:9	—	These bits are read as 0.	R
13:12	AL[1:0]	AL monitor 0x0: AL0 0x1: AL1 0x3: AL2	R
14	—	This bit is read as 0.	R
15	—	This bit is read as 1.	R
16	STBY1	Indicate the MCU is in Software Standby mode 0: MCU is not in Software Standby mode 1: MCU is in Software Standby mode	R
19:17	—	These bits are read as 0.	R
20	DSTBY1	Indicate the MCU is in Deep Software Standby mode 1 0: MCU is not in Deep Software Standby mode 1 1: MCU is in Deep Software Standby mode 1	R

Bit	Symbol	Function	R/W
21	DSTBY2	Indicate the MCU is in Deep Software Standby mode 2 0: MCU is not in Deep Software Standby mode 2 1: MCU is in Deep Software Standby mode 2	R
22	DSTBY3	Indicate the MCU is in Deep Software Standby mode 3 0: MCU is not in Deep Software Standby mode 3 1: MCU is in Deep Software Standby mode 3	R
31:23	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-2
Note 1. The address of MCUSTAT register can only be accessed by external debugger.

The MCUSTAT register indicates the MCU status and authentication status. All bits in the MCUSTAT register are monitoring bits, so they cannot be reset. The register has no security protection.

2.8.6.2 MCUCTRL : MCU Control Register

Base address: OCD_CPU = 0x4001_1000
OCD_CPU_NS = 0x5001_1000
OCD_DAP = 0x8001_1000*1

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUWAIT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EDBGRQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EDBGRQ	External Debug Request. Writing 1 to the bit causes a CPU Halt or Debug Monitor exception request. 0: Debug event is not requested 1: Debug event is requested	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
16	CPUWAIT	CPU Wait Setting 0: Deassert CPUWAIT CPU starts boot-up sequence and instruction execution out of Reset. 1: Assert CPUWAIT CPU is forced into a quiescent state out of Reset.	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5, P-TYPE-2
Note: This register is valid only when any of the following conditions are true:

- CDBGPWRUPREQ is 1, and AL is AL2 or AL1.
- OFS1(_SEC).SWDBG is 0, and DBGAUTH0.DBGEN0 is 1.

When none of the above conditions are true, the MCU ignores this register setting and considers all bits to be 0.
Note 1. This address is accessed from external debugger.

The MCUCTRL register provides the external debugger ability to enter debug state by external event.

2.8.6.3 Authentication Control Registers

Authentication registers accessible from debugger are placed in OCDREG.

2.8.6.3.1 JBMDR : JTAG Boot Mode Entry Register

Base address: OCD_CPU = 0x4001_1000
OCD_CPU_NS = 0x5001_1000
OCD_DAP = 0x8001_1000*1

Offset address: 0x100

Bit position:	31																								7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	KEY[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	KEY[7:0]	Mode entry key Pin reset releases after 0xA5 is set, then system enters the JTAG boot mode. MDSR.JBOTS = 1 when system transitions to JTAG boot mode.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5, P-TYPE-2
Note 1. This address is accessed from external debugger.

The JBMDR register sets command from the debugger.

This register is initialized by the factors listed Table 5.2 in section 5, Resets, but also initialized by debugger disconnection.

2.8.6.3.2 JBRDR : JTAG Boot Receive Data Register

Base address: OCD_CPU = 0x4001_1000
OCD_CPU_NS = 0x5001_1000
OCD_DAP = 0x8001_1000*1

Offset address: 0x120

Bit position:	31																															0
Bit field:	RDAT[31:0]																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RDAT[31:0]	Received data register R/W is possible for both the external host and CPU, but the following usage is recommended: W: External host R: CPU (Boot firmware) When JBSTR.RDF = 1, write is not possible and an error occurs.	R/W

Note: S-TYPE-5, P-TYPE-2
Note 1. This address is accessed from external debugger.

The JBRDR register is used for receiving data from the debugger.

2.8.6.3.3 JBTDR : JTAG Boot Transmit Data Register

Base address: OCD_CPU = 0x4001_1000
OCD_CPU_NS = 0x5001_1000
OCD_DAP = 0x8001_1000*1

Offset address: 0x130

Bit position:	31																															0
Bit field:	TDAT[31:0]																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TDAT[31:0]	Transmitted data register R/W is possible for both the external host and CPU, but the following usage is recommended: W: CPU (Boot firmware) R: External host When JBSTR.TDE = 1, read is not possible and an error occurs.	R/W

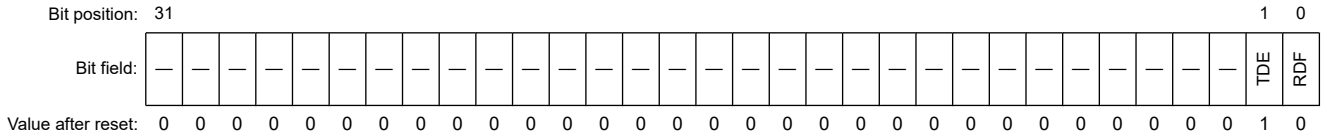
Note: S-TYPE-5, P-TYPE-2
Note 1. This address is accessed from external debugger.

The JBTDR register is used for transmitting data to the debugger.

2.8.6.3.4 JBSTR : JTAG Boot Status Register

Base address: OCD_CPU = 0x4001_1000
OCD_CPU_NS = 0x5001_1000
OCD_DAP = 0x8001_1000*1

Offset address: 0x140



Bit	Symbol	Function	R/W
0	RDF	Receive buffer full 0: No receiving data 1: There is receiving data	R/W
1	TDE	Transmit data empty 0: There is data transmission 1: No data transmission	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5, P-TYPE-2
Note 1. This address is accessed from external debugger.

The JBSTR register is used for monitoring booting status.

RDF flag (Receive buffer full

The RDF bit indicates receive buffer is full or not.

[Set conditions]

- Read access to JBTDR
- Write 1 to JBSTR.RDF

[Clear conditions]

- Write access to JBTDR
- Write 0 to JBSTR.RDF

TDE flag (Transmit data empty

The TDE bit indicates transmit data is empty or not.

[Set conditions]

- Read access to JBTDR
- Write 1 to JBSTR.TDE

[Clear conditions]

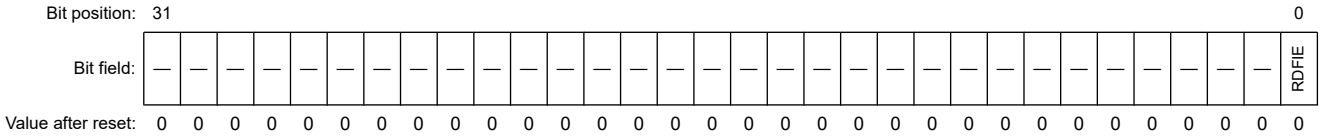
- Write access to JBTDR

- Write 0 to JBSTR.TDE

2.8.6.3.5 JBICR : JTAG Boot Interrupt Control Register

Base address: OCD_CPU = 0x4001_1000
OCD_CPU_NS = 0x5001_1000
OCD_DAP = 0x8001_1000*1

Offset address: 0x150



Bit	Symbol	Function	R/W
0	RDFIE	Receive buffer full interrupt enabled 0: Interrupt request disabled by RDF = 1 1: Enable interrupt request by RDF = 1	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

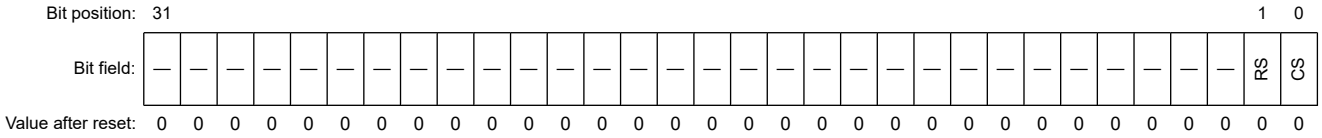
Note: S-TYPE-5, P-TYPE-2
Note 1. This address is accessed from external debugger.

The JBICR register is used for controlling interrupt during JTAG boot.

2.8.6.3.6 FSBLSTATM : First Stage Boot Loader Status Monitor Register

Base address: OCD_CPU = 0x4001_1000
OCD_CPU_NS = 0x5001_1000
OCD_DAP = 0x8001_1000*1

Offset address: 0x300



Bit	Symbol	Function	R/W
0	CS	FSBL completion status 0: FSBL is not complete 1: FSBL is complete	R
1	RS	FSBL result status 0: FSBL failed 1: FSBL passed	R
31:2	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-2
Note 1. This address is accessed from external debugger.

The FSBLSTATM indicates first stage boot loader status.

Deep Software Standby reset and software reset do not initialize the CS and RS bits of the FSBLSTATM register when FSBL execution is skipped by those resets.

2.8.6.4 CoreSight Component Registers

OCDREG has CoreSight Component Registers defined in ARM CoreSight Architecture. Table 2.22 shows the list of the registers.

See section 2.14. References [7] for the detail of each register

Table 2.22 CoreSight component registers

Name	Address	Access size	R/W	Initial value
PID4	0x8001_1FD0	32 bits	Read-only	0x00000004
PID5	0x8001_1FD4	32 bits	Read-only	0x00000000
PID6	0x8001_1FD8	32 bits	Read-only	0x00000000
PID7	0x8001_1FDC	32 bits	Read-only	0x00000000
PID0	0x8001_1FE0	32 bits	Read-only	0x00000004
PID1	0x8001_1FE4	32 bits	Read-only	0x00000030
PID2	0x8001_1FE8	32 bits	Read-only	0x0000000A
PID3	0x8001_1FEC	32 bits	Read-only	0x00000000
CID0	0x8001_1FF0	32 bits	Read-only	0x0000000D
CID1	0x8001_1FF4	32 bits	Read-only	0x000000F0
CID2	0x8001_1FF8	32 bits	Read-only	0x00000005
CID3	0x8001_1FFC	32 bits	Read-only	0x000000B1

Note: S-TYPE-5, P-TYPE-2

2.9 CoreSight Cross Trigger Interface (CTI)

Table 2.23 CTI channel

Number of CTI channel	CTITRIGIN		CTITRIGOUT	
CTI	0	ACQCOMP	0	—
	1	FULL	1	—
	2	—	2	ETB FLUSHIN
	3	—	3	ETB TRIGIN
	4	—	4	—
	5	—	5	—
	6	—	6	—
	7	—	7	—
CTI0	0	Processor Halted	0	Processor debug request
	1	DWT Comparator Output 0	1	Processor Restart
	2	DWT Comparator Output 1	2	CTIIRQ[0] (connected to ICU0)
	3	DWT Comparator Output 2	3	CTIIRQ[1] (connected to ICU0)
	4	ETM Event Output 0	4	ETM Event Input 0
	5	ETM Event Output 1	5	ETM Event Input 1
	6	—	6	ETM Event Input 2
	7	—	7	ETM Event Input 3

The MCU supports processor CTI0 and CoreSight Common CTI which are implemented in Debug module. The ICU setting is necessary when CTIIRQ is used.

2.10 CoreSight ATB Funnel

The MCU has one CoreSight ATB funnel. The funnel has two ATB slaves and one ATB master, and it is used to select the debug trace sources from ETM and ITM to ETB. Figure 2.3 shows the CoreSight ATB connection in the MCU.

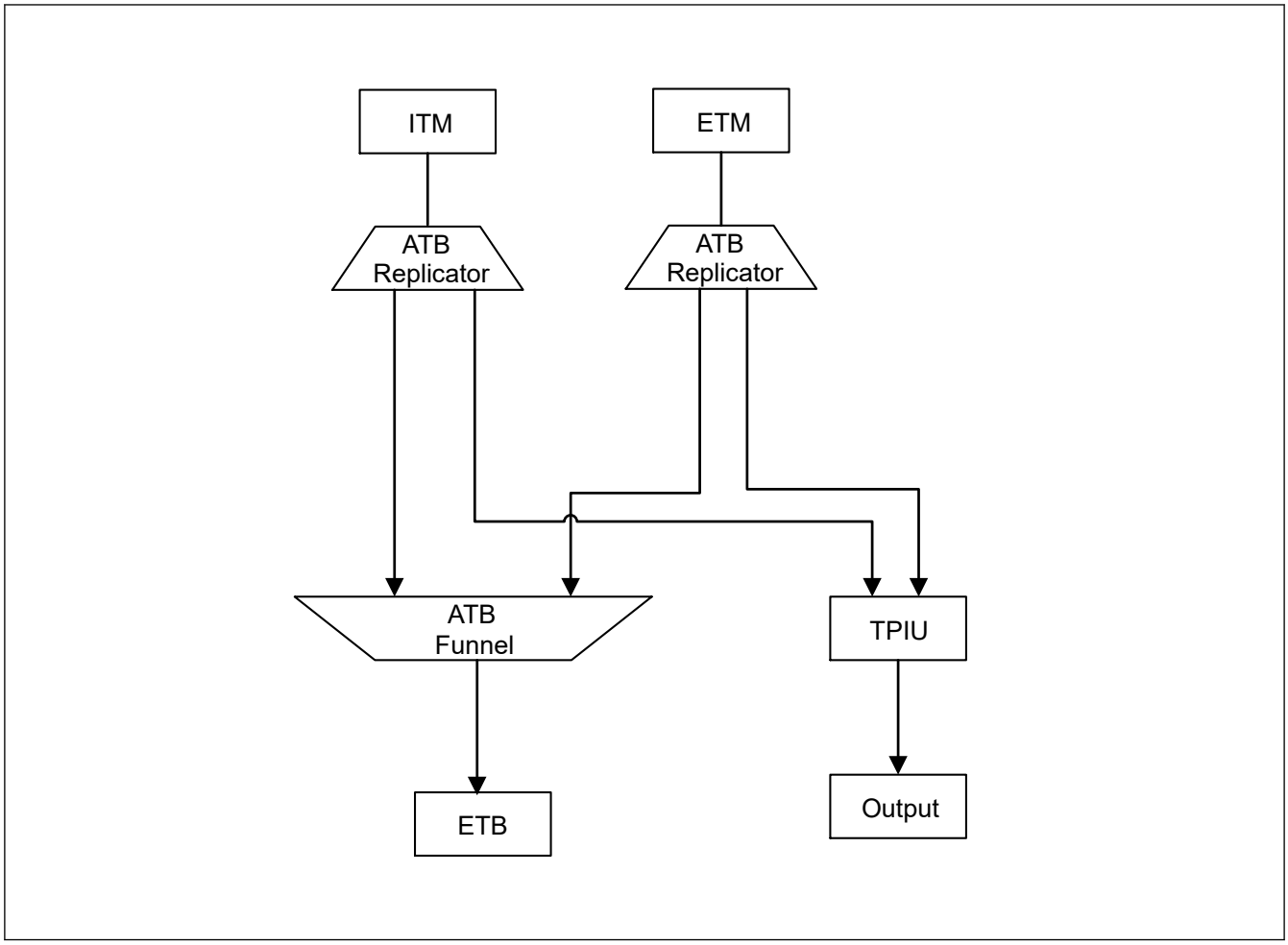


Figure 2.3 CoreSight ATB connection in the MCU

Table 2.24 shows funnel ATB slave connection.

Table 2.24 Funnel ATB slave connection.

ATB slave number	Connected trace source
#0	CPU-ITM
#1	CPU-ETM

See section 2.14. References [5] for detail of ATB and Funnel.

2.11 SysTick System Timer

The MCU has two SysTick system timers, secure SysTick and non-secure SysTick.

The SysTick source is CPUCLK or SYSTICKCLK (MOCO/8). When the SysTick source is SYSTICKCLK (MOCO/8), the CPUCLK frequency must be 2 MHz or higher.

Figure 2.4 shows the SysTick Timer block diagram.

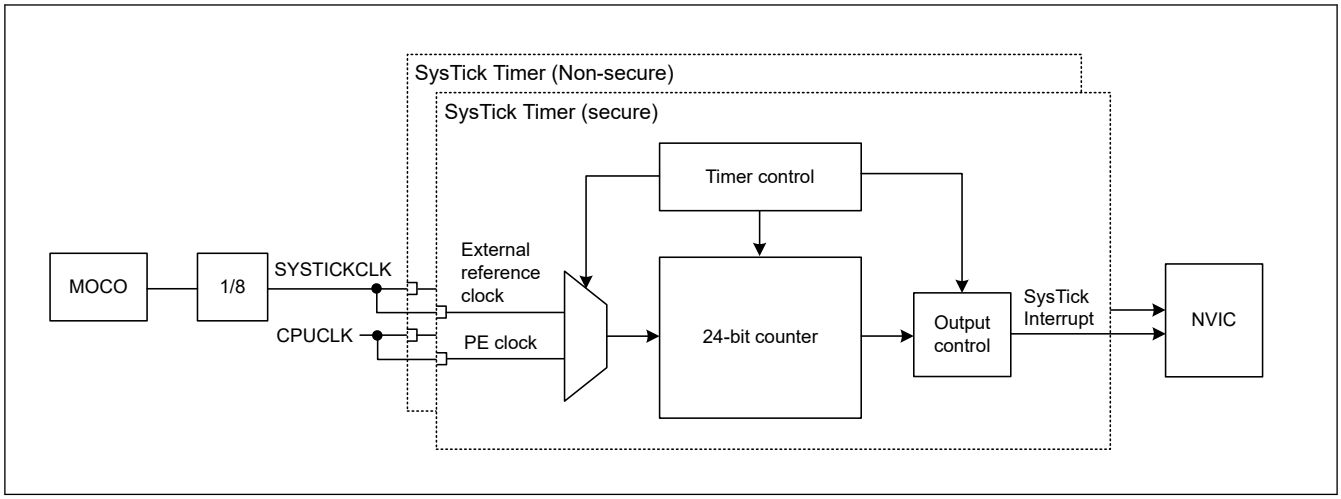


Figure 2.4 SysTick Timer block diagram

2.12 CoreSight Timestamp Generator

The MCU has a CoreSight Timestamp Generator to provide timestamp to ITM and ETM.

The timestamp is generated by a 64-bit counter operating with ICLK/2.

See section 2.14. References [5] for detail.

2.13 OCD Emulator Connection

In this product, the MCU confirms access permission for MCU resources by checking the current authentication level stored in OCDREG.

There are two ways to control authentication level:

- Through JTAG/SWD. See Figure 2.5.
- Using software. See Figure 2.6.

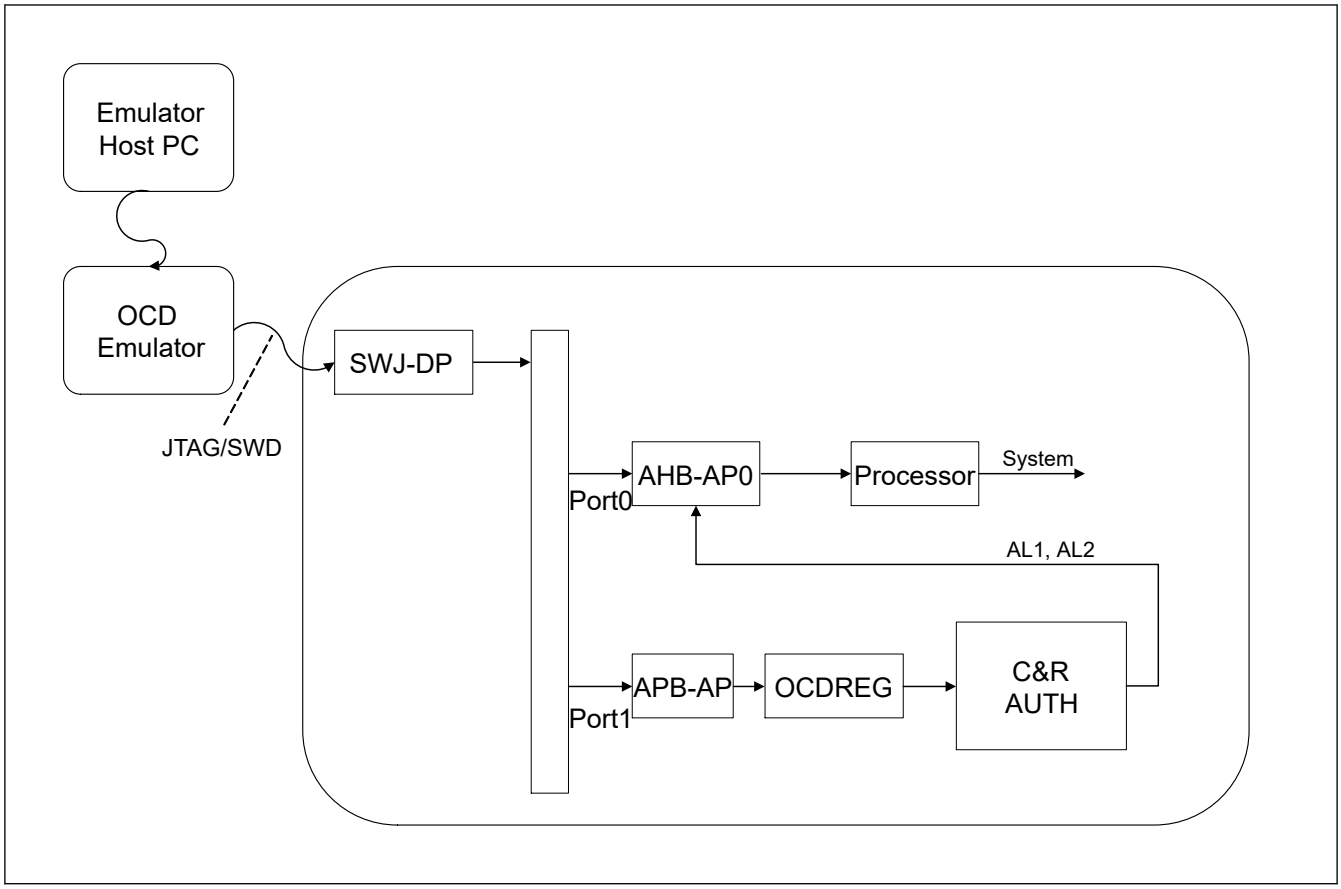


Figure 2.5 Control authentication level through JTAG/SWD

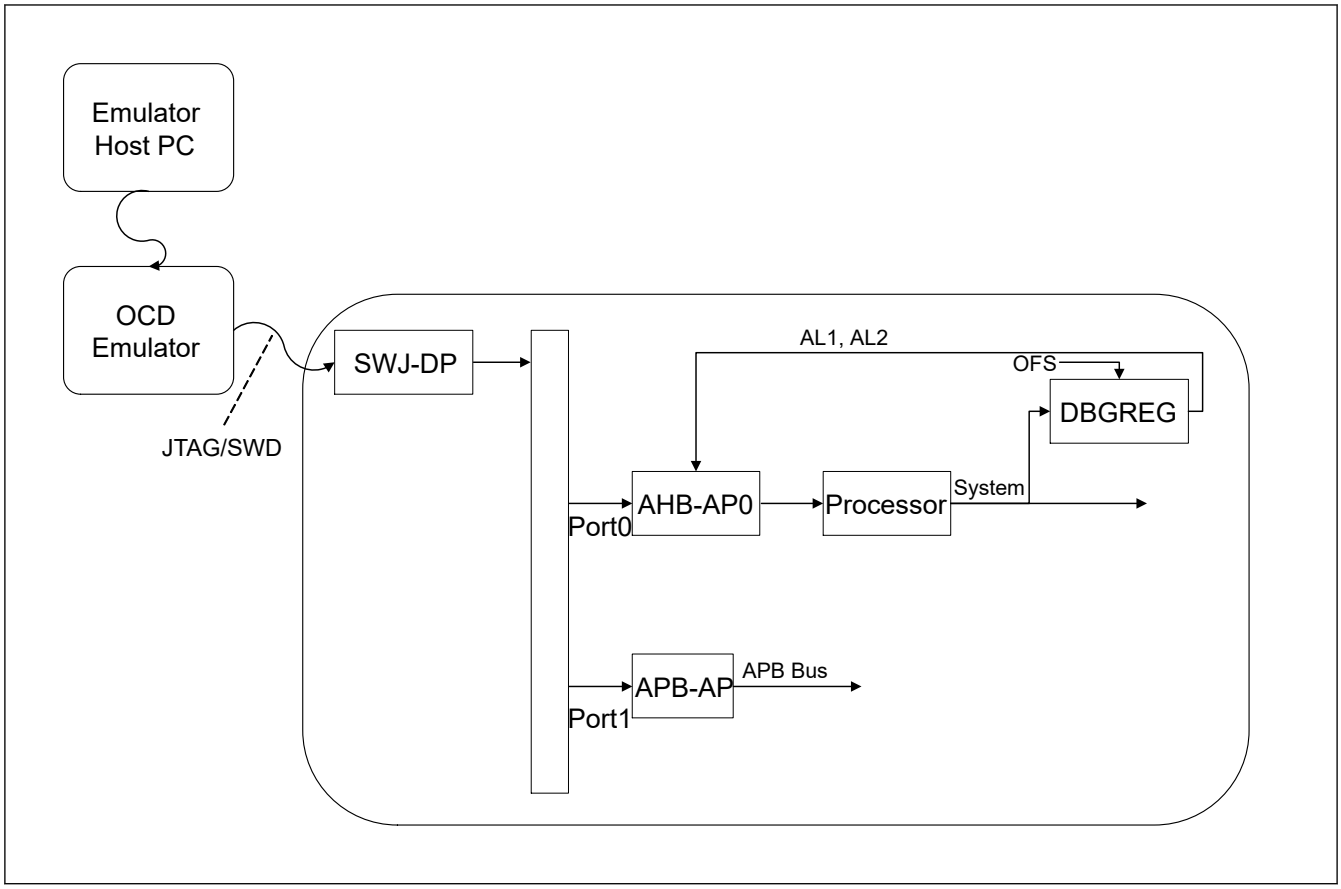


Figure 2.6 Control authentication level using software

2.13.1 SYOCDCR.DBGEN

See section 2.7.3.1. Low power mode for details.

2.13.2 Restriction in OCD Emulator Connecting

There are some restrictions regarding emulator access. This section describes those restrictions.

2.13.3 Starting Connection While in Low Power Mode

When starting a JTAG/SWD connection from an OCD emulator, the MCU must be in Normal, CPU Sleep, or CPU Deep Sleep mode. If the MCU is in Software Standby or Deep Software Standby mode, the OCD emulator can cause the MCU to hang.

2.13.4 Low Power Mode Change During Debugging

When the MCU is in OCD mode, the low power mode can be changed. However, system bus access from AHB-AP is prohibited in Deep Sleep, Software Standby, or Deep Software Standby mode. Only SWJ-DP, APB-AP, and OCDREG can be accessed from OCD emulator in these modes. Table 2.25 shows the limitations.

Table 2.25 Low power mode change during debugging (1 of 2)

Current mode	Start OCD emulator connection	Change low power mode	Access AHB-AP and system bus	Access APB-AP and OCDREG
Normal	Available	Available	Available	Available
Sleep	Available	Available	Available	Available
Deep Sleep	Available	Available	N/A	Available
Software Standby	N/A	Available	N/A	Available

Table 2.25 Low power mode change during debugging (2 of 2)

Current mode	Start OCD emulator connection	Change low power mode	Access AHB-AP and system bus	Access APB-AP and OCDREG
Deep Software Standby	N/A	Available	N/A	Available

2.13.5 Connecting Sequence and JTAG/SWD Authentication

This section provides steps on how to authenticate the MCU. First, the section introduces which mechanism is used to authenticate the MCU. Next, using the previously described mechanism, the section describes steps for connecting sequence and authenticating the MCU using JTAG/SWD. See section 2.8.5. DBGREG and section 2.8.6. OCDREG for details about the registers control JTAG/SWD authentication.

2.13.5.1 JTAG/SWD Authentication Mechanism

The common platform of the MCU supports challenge-response authentication.

For challenge-response authentication, Boot FW authenticates.

2.13.5.2 Connecting Sequence and JTAG/SWD Authentication

For challenge-response authentication:

1. Connect the OCD debugger to the MCU through the JTAG or SWD interface.
2. Set up SWJ-DP to access DAP bus.
   In the setup, the OCD emulator must assert CDBGPWRUPREQ in the SWJDP Control Status Register, and then wait until CDBGPWRUPACK in the same register is asserted.
3. Set the APB-AP to access OCDREG. This APB-AP is connected to DAP bus port 1.
4. Set boot mode request.
5. Negate RES.
6. Check MCUSTAT.AL.
   If MCUSTAT.AL is higher than or equal to request AL, there is no need to authenticate.
7. Set authentication data.
8. Check MCUSTAT.AL.
   (a) If MCUSTAT.AL does not request AL, authentication failed.
   (b) If retry authentication: go to step 7.
9. Assert RES to move to single chip mode.
10. Set up the AHB-AP to access the system address space. The AHB-AP is connected to DAP bus port 0.
11. Start accessing the CPU debug resources using the AHB-AP.

2.14 References

[1] Arm Limited., *Arm®v8-M Architecture Reference Manual* (ARM DDI 0553B)

[2] Arm Limited., *Arm® Cortex®-M85 Processor Technical Reference Manual* (ARM 101924)

[3] Arm Limited., *Arm® Cortex®-M85 Processor Integration and Implementation Manual* (ARM 101925)

[4] Arm Limited., *Arm® Cortex®-M85 Processor Devices Generic User Guide* (ARM 101928)

[5] Arm Limited., *Arm® CoreSight™ SoC-400 Technical Reference Manual* (ARM DDI 0480G)

[6] Arm Limited., *Arm® CoreSight™ ETM-M85 Technical Reference Manual* (ARM 101926)

[7] Arm Limited., *Arm® CoreSight™ Architecture Specification* (ARM IHI 0029F)

[8] Arm Limited., *Arm® Embedded Trace Macrocell Architecture Specification* (ARM IHI 0064H.a)

[9] Arm Limited., *Arm® CoreSight SoC-400 User Guide* (ARM 100490)

2.15 Usage Notes

2.15.1 Restrictions for OSPI

There are restrictions to using OSPI, see section 37.4.3. Memory Write Combination Mode and section 37.4.5. Restriction in 8D-8D-8D profile 1.0 formatt.

3. Operating Modes

3.1 Overview

Table 3.1 shows the selection of operating modes by the mode-setting pin(MD pin) and JTAG accessible command. For details, see section 3.3. Details of Operating Modes.

Table 3.1 Selection of operating modes by the mode-setting pin

Mode-setting pin (MD)	Operating mode	On-chip Flash	External bus
1	Single chip mode/ JTAG Boot Mode	Enable	Disable
0	SCI / USB boot mode	Enable	Disable

3.2 Operating Mode Types and Selection

Table 3.1 show the relationship between levels on the mode-setting pins (MD) on release from the reset state and the operating mode selected at that time. For details on each of the operating modes, see section 3.3. Details of Operating Modes. Operation starts with the on-chip Flash enabled and the external bus disabled, regardless of the mode in which operation started.

3.3 Details of Operating Modes

3.3.1 Single-Chip Mode

In single-chip mode, all I/O pins are available for use as input or output port, inputs or outputs for peripheral functions, or as interrupt inputs. When a reset is released while the MD pin is high, the MCU starts in single-chip mode and the on-chip flash is enabled.

3.3.2 JTAG Boot Mode

In this mode, the on-chip flash memory programming routine (JTAG boot program), stored in the boot area within the MCU, is used. The on-chip flash, including code flash memory and data flash memory, can be modified from outside the MCU by using the JTAG and SWD interface. CSW.DbgSwEnable bit in APB-AP Control/Status Word register must be 1 in this mode.

To enter this mode, it is necessary to input the request from the JTAG and SWD-I/F during RES-pin reset.

3.3.3 SCI Boot Mode

In this mode, the on-chip flash memory programming routine (SCI boot program), stored in a dedicated area within the MCU, is used. The on-chip flash including the code flash memory and data flash memory can be modified from outside the MCU by using a universal asynchronous receiver/transmitter (UART) SCI. For details, see section 52, Flash Memory. The MCU starts in SCI boot mode if the MD pin is held low on release from the reset state.

3.3.4 USB Boot Mode

In this mode, the on-chip flash memory programming routine (USB boot program), stored in the boot area within the MCU, is used. The on-chip flash, including the code flash memory and data flash memory, can be modified from outside the MCU by using the USB. For details, see section 52, Flash Memory. The MCU starts in USB boot mode if the MD pin is held low on release from the reset state.

3.4 Operating Modes Transitions

3.4.1 Operation modes and the relationship of Mode Transition

Figure 3.1 shows that the operation modes and the relationship of mode transition. Operating mode can be transitioned in the direction of the arrow at the figure.

JTAG Boot mode is not possible using POR.

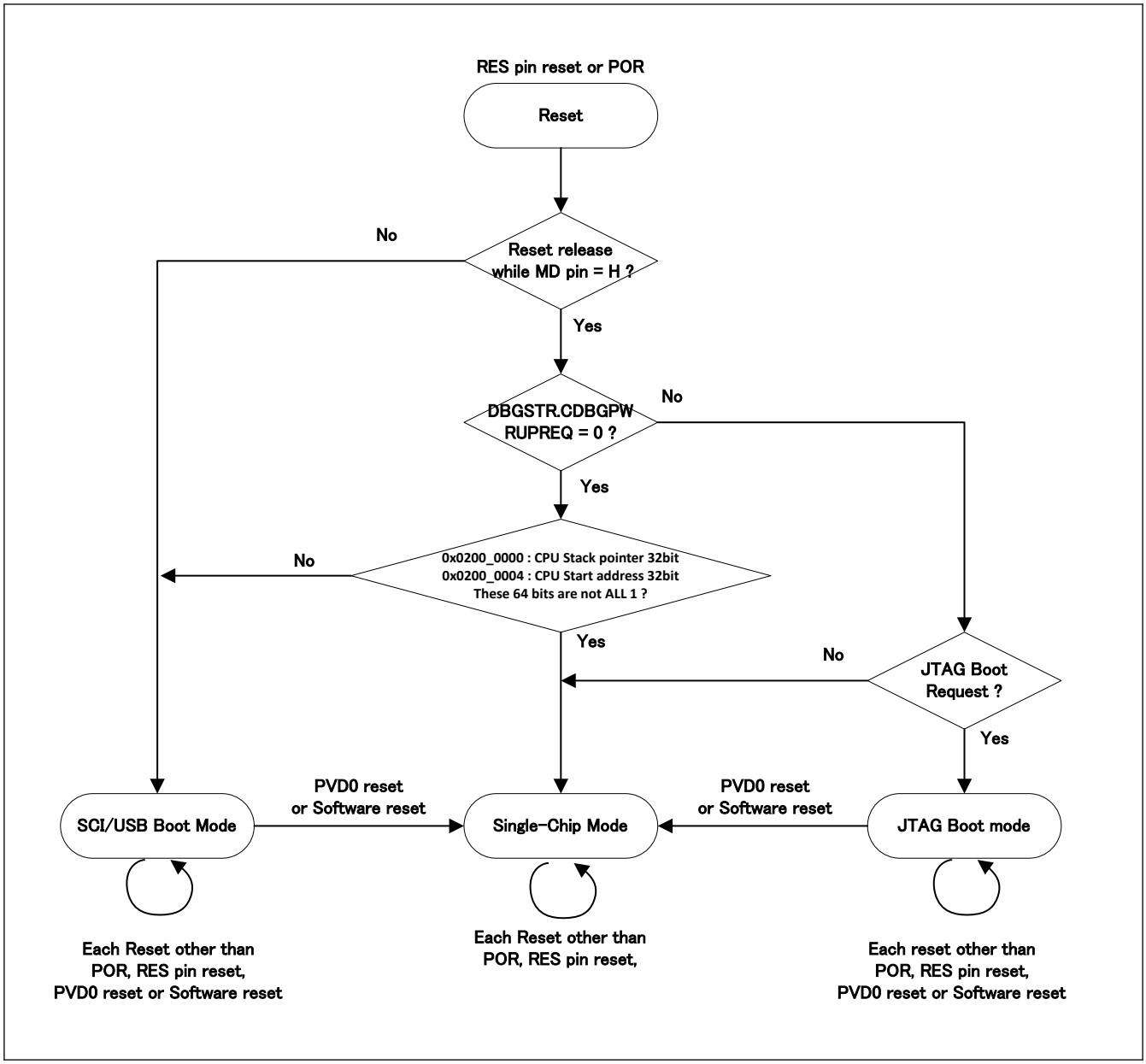
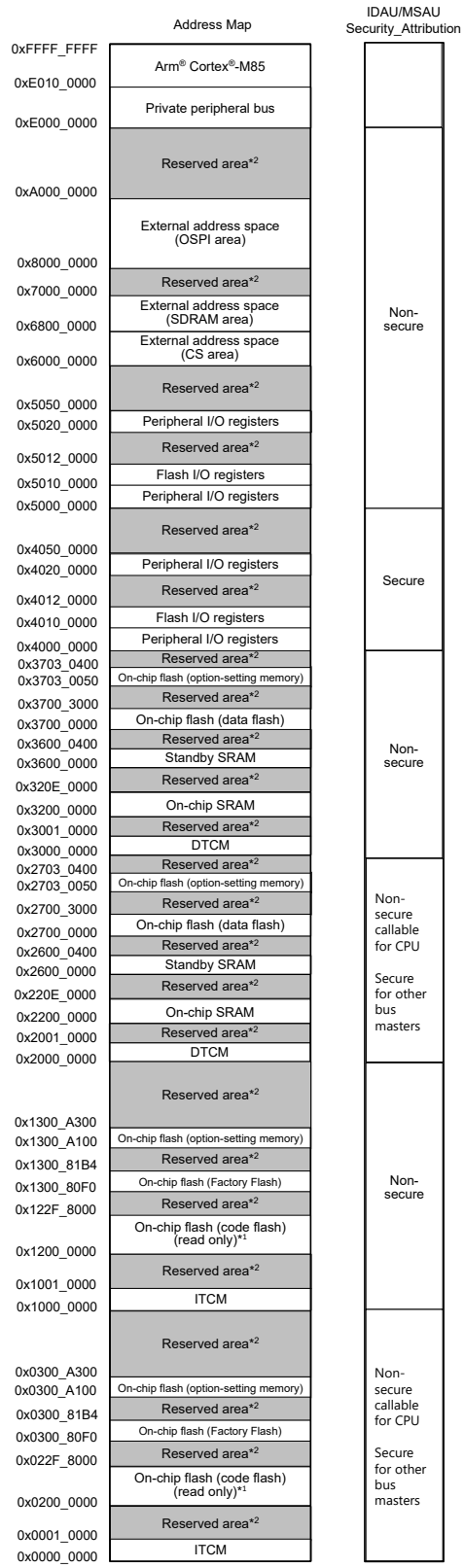


Figure 3.1 Operation modes and the relationship of mode transition

4. Address Space

4.1 Address Space

The MCU supports a 4 GB linear address space ranging from 0x0000_0000 to 0xFFFF_FFFF that can contain both program and data. In the address from 0x0000_0000 to 0x5FFF_FFFF, secure and non-secure region are isolated by using bit 28 of the address. For detail on the security attributes, see section 43.3. Arm Security Features. Figure 4.1 shows the memory map example of 2 MB flash product.



Note 1. See Table 4.1 and Table 4.2. The capacity of the flash differs depending on the product.
Note 2. Do not access reserved areas.

Figure 4.1 Memory map example of 2 MB flash product

Table 4.1 Capacity of the code flash memory, data flash memory, and SRAM0 in Secure alias

Code flash memory			Data flash memory		On-chip SRAM	
Capacity	Address		Capacity	Address	Capacity	Address
	Linear mode	Dual mode (BANKSEL.BANKSWP[2:0] = 111b)				
2 MB	0x0200_0000 - 0x021F_7FFF	upper side bank 0x0220_0000 - 0x022F_7FFF	12 KB	0x2700_0000 - 0x2700_2FFF	896 KB	0x2200_0000 - 0x220D_FFFF
		lower side bank 0x0200_0000 - 0x020F_7FFF				
1 MB	0x0200_0000 - 0x020F_FFFF	upper side bank 0x0220_0000 - 0x0227_FFFF				
		lower side bank 0x0200_0000 - 0x0207_FFFF				

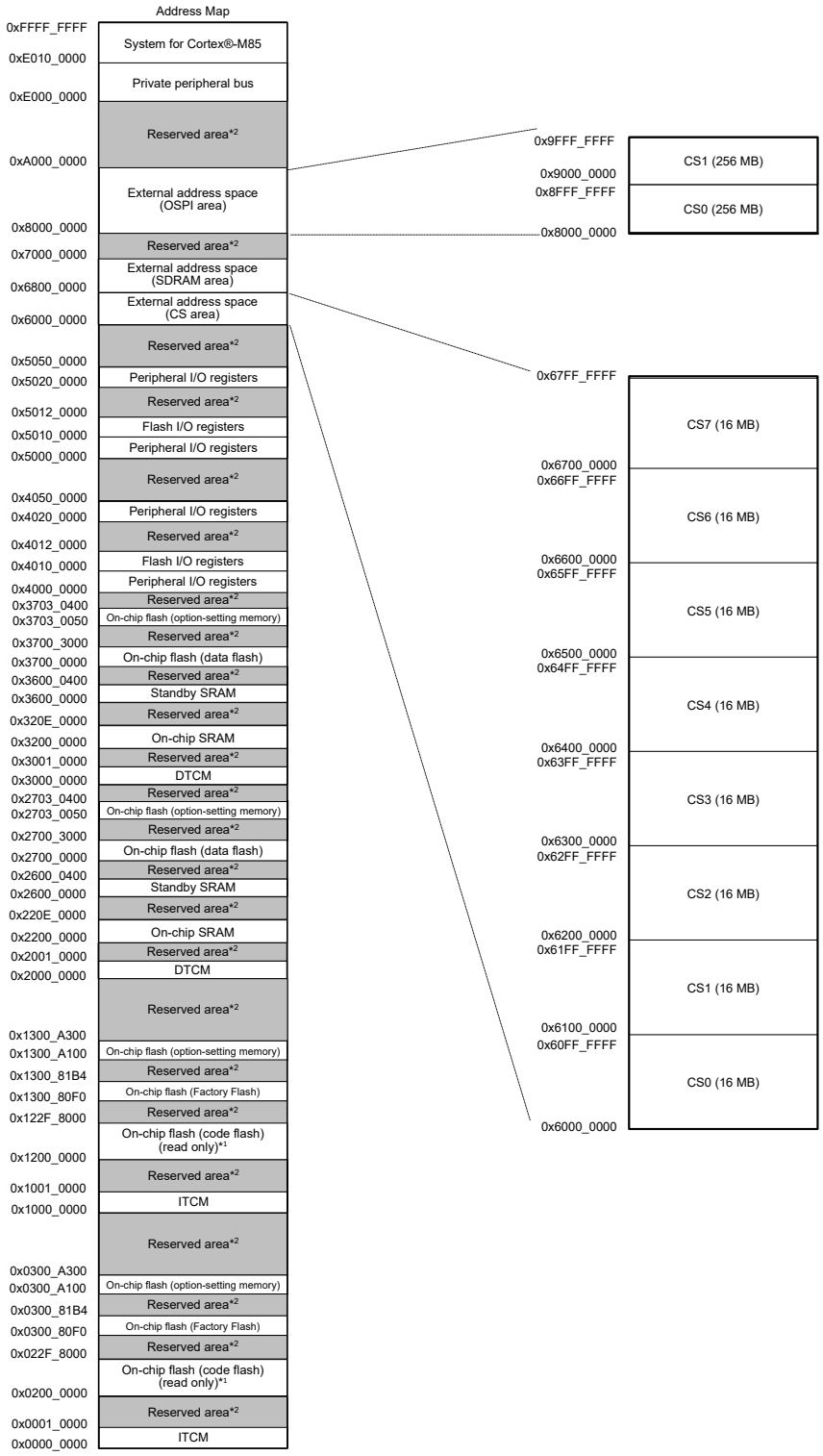
Table 4.2 Capacity of the code flash memory, data flash memory, and On-chip SRAM in Non-secure alias

Code flash memory			Data flash memory		On-chip SRAM	
Capacity	Address		Capacity	Address	Capacity	Address
	Linear mode	Dual mode (BANKSEL.BANKSWP[2:0] = 111b)				
2 MB	0x1200_0000 - 0x121F_7FFF	upper side bank 0x1220_0000 - 0x122F_7FFF	12 KB	0x3700_0000 - 0x3700_2FFF	896 KB	0x3200_0000 - 0x320D_FFFF
		lower side bank 0x1200_0000 - 0x120F_7FFF				
1 MB	0x1200_0000 - 0x120F_FFFF	upper side bank 0x1220_0000 - 0x1227_FFFF				
		lower side bank 0x1200_0000 - 0x1207_FFFF				

4.2 External Address Space

The external address space is divided into CS areas (CS0 to CS7), SDRAM area (SDCS), and OSPI areas (CS0 and CS1). The eight CS areas (CS0 to CS7) each correspond to the CSn signal output from a CSn (n = 0 to 7) pin. The two OSPI areas (CS0 and CS1) each correspond to the OM_CSn signal output from a OM_CSn (n = 0, 1) pin.

Figure 4.2 shows the address ranges associated with the individual CS areas (CS0 to CS7), SDRAM area (SDCS), and OSPI areas (CS0 and CS1).



Note 1. See Table 4.1. The capacity of the flash differs depending on the product.
Note 2. Do not access reserved areas.

Figure 4.2 Detailed address map for CS area and OSPI area

5. Resets

5.1 Overview

This MCU provides the following 13 types of reset.

It also has a function to control the reset generation by only Secure program access.

System reset is all factors such that can initialize CPU core. All reset factors except VBATT_POR are system reset for this product.

On the other hand, CPU has a reset that initializes itself individually. That reset is not a system reset. See section 2, CPU for details.

Table 5.1 lists the reset names and sources.

Table 5.1 Reset names and sources

Reset Name		Source
System reset	RES Pin reset	The RES pin voltage input is driven low
	Power-on reset (POR)	VCC falls (voltage detection: V_{POR})*1
	Voltage Monitor 0 reset	VCC falls (voltage detection: V_{det0})*1
	Voltage Monitor 1 reset	VCC falls (voltage detection: V_{det1})*1
	Voltage Monitor 2 reset	VCC falls (voltage detection: V_{det2})*1
	Independent Watchdog Timer reset	IWDT underflow or refresh error
	Watchdog Timer reset	WDT underflow or refresh error for CPU
	CPU Lockup reset	This reset is generated when CPU encounters lockup
	Bus Error reset	BUS error (MSAU error, MMPU error, Illegal address error, STZF error, Slave Bus error, Bufferable write error)
	Common Memory Error reset	RAM error (ECC error or Parity error of SRAM and Standby SRAM)
	Deep Software Standby reset	Deep Software Standby mode is canceled by an interrupt
	Software reset	Register setting (use the software reset bit AIRCR.SYSRESETREQ)
VBATT_POR reset		VBATT_R falls (voltage detection: $V_{PDR\ (BATR)}$)*1

Note 1. For details on the voltages to be monitored (V_{POR}, V_{det0}, V_{det1}, V_{det2}, and $V_{PDR\ (BATR)}$), see section 7, Programmable Voltage Detection (PVD), section 11, Battery Backup Function, and section 60, Electrical Characteristics.

The internal state and pins are initialized by a reset. Table 5.2 and Table 5.3 list the targets initialized by resets.

Some of these registers are initialized when returning from Software Standby mode. See section 10, Low Power Modes.

In this table, registers of modules marked as "Undefined" in Software Standby mode are initialized.

Table 5.2 Reset detect flags initialized by each reset source (1 of 4)

Flag to be initialized	Reset source						
	RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer reset	CPU Lockup reset	Voltage monitor 1 reset
Power-On Reset Detect Flag (RSTSR0.PORF)	✓	—	—	—	—	—	—
Voltage Monitor 0 Reset Detect Flag (RSTSR0.PVD0RF)	✓	✓	—	—	—	—	—
Independent Watchdog Timer Reset Detect Flag (RSTSR1.IWDTRF)	✓	✓	✓	—	—	—	—
Watchdog Timer Reset Detect Flag (RSTSR1.WDT0RF)	✓	✓	✓	—	—	—	—
CPU Lockup Reset Detect Flag (RSTSR1.CLU0RF)	✓	✓	✓	—	—	—	—

Table 5.2 Reset detect flags initialized by each reset source (2 of 4)

Flag to be initialized	Reset source						
	RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer reset	CPU Lockup reset	Voltage monitor 1 reset
Voltage Monitor 1 Reset Detect Flag (RSTSR0.PVD1RF)	✓	✓	✓	—	—	—	—
Voltage Monitor 2 Reset Detect Flag (RSTSR0.PVD2RF)	✓	✓	✓	—	—	—	—
Software Reset Detect Flag (RSTSR1.SWRF)	✓	✓	✓	—	—	—	—
Bus error Reset Detect Flag (RSTSR1.BUSRF)	✓	✓	✓	—	—	—	—
Common Memory Error Reset Detect Flag (RSTSR1.CMRF)	✓	✓	✓	—	—	—	—
Deep Software Standby Reset Detect Flag (RSTSR0.DPSRSTF)	✓	✓	✓	—	—	—	—
Cold Start/Warm Start Determination Flag (RSTSR2.CWSF)	—	✓	—	—	—	—	—

Table 5.2 Reset detect flags initialized by each reset source (3 of 4)

Flag to be initialized	Reset source							
	Voltage monitor 2 reset	Software reset	Bus error reset	Common memory error reset	Deep software standby reset			VBATT_POR*1
					Deep Software Standby mode 1	Deep Software Standby mode 2	Deep Software Standby mode 3	
Power-On Reset Detect Flag (RSTSR0.PORF)	—	—	—	—	—	—	—	—
Voltage Monitor 0 Reset Detect Flag (RSTSR0.PVD0RF)	—	—	—	—	—	—	—	—
Independent Watchdog Timer Reset Detect Flag (RSTSR1.IWDTRF)	—	—	—	—	—	✓	✓	—
Watchdog Timer Reset Detect Flag (RSTSR1.WDT0RF)	—	—	—	—	—	✓	✓	—
CPU Lockup Reset Detect Flag (RSTSR1.CLU0RF)	—	—	—	—	—	✓	✓	—
Voltage Monitor 1 Reset Detect Flag (RSTSR0.PVD1RF)	—	—	—	—	—	—	—	—
Voltage Monitor 2 Reset Detect Flag (RSTSR0.PVD2RF)	—	—	—	—	—	—	—	—
Software Reset Detect Flag (RSTSR1.SWRF)	—	—	—	—	—	✓	✓	—
Bus error Reset Detect Flag (RSTSR1.BUSRF)	—	—	—	—	—	✓	✓	—
Common Memory Error Reset Detect Flag (RSTSR1.CMRF)	—	—	—	—	—	✓	✓	—

Table 5.2 Reset detect flags initialized by each reset source (4 of 4)

Flag to be initialized	Reset source							
	Voltage monitor 2 reset	Software reset	Bus error reset	Common memory error reset	Deep software standby reset			VBATT_POR*1
					Deep Software Standby mode 1	Deep Software Standby mode 2	Deep Software Standby mode 3	
Deep Software Standby Reset Detect Flag (RSTSR0.DPSRSTF)	—	—	—	—	—	—	—	—
Cold Start/Warm Start Determination Flag (RSTSR2.CWSF)	—	—	—	—	—	—	—	—

Note: ✓ : Initialized to 0
— : Not initialized
Note 1. For VBATT_POR details, See section 11, Battery Backup Function.

Table 5.3 Module-related registers initialized by each reset source (1 of 4)

Registers to be initialized		Reset source						
		RES# pin reset	Power-on reset	Voltage monitoring 0 reset	Independent watchdog timer reset	Watchdog timer reset	CPU Lockup reset	Voltage monitoring 1 reset
Voltage Monitor Function 1 registers	PVD1CR0, PVD1CMPCR, PVD1FCR	✓	✓	✓	✓	✓	—	—
	PVD1CR1. PVD1SR	✓	✓	✓	✓	✓	—	—
Voltage Monitor Function 2 registers	PVD2CR0, PVD2CMPCR, PVD2FCR	✓	✓	✓	✓	✓	—	—
	PVD2CR1. PVD2SR	✓	✓	✓	✓	✓	—	—
SOSC registers	SOSCCR, SOMCR	—	—	—	—	—	—	—
LOCO registers	LOCOUTCR	—	✓	✓	—	—	—	—
MOSC registers	MOMCR	✓	✓	✓	✓	✓	✓	✓
HOCO control register 2	HOCOCR2	—	✓	✓	—	—	—	—
Pin states (except XCIN / XCOUT)		✓	✓	✓	✓	✓	✓	✓
Pin states (XCIN / XCOUT)		—	—	—	—	—	—	—
IO capture and tamper detection such as VBAT (sampling timing for RTC) (RTCIC0-2)	VBTICTLR,VBTICTLR2,VBTADSR, VBTADCR1,VBTADCR2	—	—	—	—	—	—	—
VBATT Battery power supply switch control register 1	VBTBPCR1	✓	✓	✓	✓	✓	✓	✓
VBATT Battery Power Supply Switch Control register 2	VBTBPCR2	—	—	—	—	—	—	—

Table 5.3 Module-related registers initialized by each reset source (2 of 4)

Registers to be initialized		Reset source						
		RES# pin reset	Power-on reset	Voltage monitoring 0 reset	Independent watchdog timer reset	Watchdog timer reset	CPU Lockup reset	Voltage monitoring 1 reset
VBATT Backup Enable register	VBTBER	—	✓	—	—	—	—	—
Battery Backup register	VBTBKR[n]	—	—	—	—	—	—	—
VBATT Input Monitor Register	VBTIMONR	—	—	—	—	—	—	—
Independent Watchdog Timer registers	IWDTRR, IWDTCR, IWDTSR, IWDTRCR, IWDTCSTPR	✓	✓	✓	✓	✓	✓	✓
Realtime Clock register		—	—	—	—	—	—	—
Ultra-low-power timer registers	ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR, ULPTMR1, ULPTMR2, ULPTMR3, ULPTIOC, ULPTISR, ULPTCMSR	✓	✓	✓	✓	✓	✓	✓
USBFS registers	DPUSR0R, DPUSR1R	✓	✓	✓	✓	✓	✓	✓
USBHS registers	DPUSR0R, DPUSR1R, DPUSR2R, DPUSRCR	✓	✓	✓	✓	✓	✓	✓
Reset Flag	BUSnERRADD (n = 4 to 9) BUSnERRRW (n = 4 to 9) BMSAnERRADD (n = 4 to 9) BMSAnERRRW (n = 4 to 9) BUSnERRSTAT (n = 1 to 12) MBWERRSTAT, SBWERRSTAT, SRAMESR, SRAMEARn (n = 0 to 2) STBRAMEAR	✓	✓	✓	✓	✓	✓	✓
Reset Flag		See Table 5.2						
Low Power Function registers	DPSBYCR, DPSWCR, DPSIER0 to DPSIER3, DPSIFR0 to DPSIFR3, DPSIEGR0 to DPSIEGR2 LPSCR FWEPROR SSCR1 LVOCR	✓	✓	✓	✓	✓	✓	✓
Low power Function registers	SYOCDCR	—	✓	✓	—	—	—	—
System Reset Mask Control Register *7	SYRSTMSK0, SYRSTMSK2	✓	✓	✓	—	—	—	—
ARM Debug function	MCUSTAT,MCUCTRL, JBMDR,FSBLSTATM, DBGSTR,DBGSTOPCR, DBGAUTH0,DBGAUTH1, CACHEDBGCR,TRPORTCR	—	✓	✓	—	—	—	—
TRCLK Control registers	TRCKCR	—	✓	✓	—	—	—	—
Power Gating Control registers	PDRAMSCR0, PDRAMSCR1	—	✓	✓	—	—	—	—
other than specified		✓	✓	✓	✓	✓	✓	✓

Table 5.3 Module-related registers initialized by each reset source (3 of 4)

Registers to be initialized		Reset source							
		Voltage monitoring 2 reset	Software reset	Bus error reset	Common memory error reset	Deep software standby reset			VBATT-selected voltage power-on reset
						Deep Software Standby mode 1 reset	Deep Software Standby mode 2 reset	Deep Software Standby mode 3 reset	
Voltage Monitor Function 1 registers	PVD1CR0, PVD1CMPCR, PVD1FCR	—	—	—	—	—	—	—	—
	PVD1CR1. PVD1SR	—	—	—	—	✓	✓	✓	—
Voltage Monitor Function 2 registers	PVD2CR0, PVD2CMPCR, PVD2FCR	—	—	—	—	—	—	—	—
	PVD2CR1. PVD2SR	—	—	—	—	✓	✓	✓	—
SOSC registers	SOSCCR, SOMCR	—	—	—	—	—	—	—	✓
LOCO registers	LOCOUTCR	—	—	—	—	—	✓	✓	—
MOSC registers	MOMCR	✓	✓	✓	✓	—	—	—	—
HOCO control register 2	HOCOCR2	—	—	—	—	✓*3	✓*3	✓*3	—
Pin states (except XCIN / XCOUT)		✓	✓	✓	✓	✓*2	✓*2	✓*2	—
Pin states (XCIN / XCOUT)		—	—	—	—	—	—	—	✓
IO capture and tamper detection such as VBAT (sampling timing for RTC) (RTCIC0-2)	VBTICTLR,VBTICTLR2,VBTADSR, VBTADCR1,VBTADCR2	—	—	—	—	—	—	—	✓
VBATT Battery Power Supply Switch Control register 1	VBTBPCR1	✓	✓	✓	✓	—	—	—	—
VBATT Battery Power Supply Switch Control register 2	VBTBPCR2	—	—	—	—	—	—	—	✓
VBATT Backup Enable register	VBTBER	—	—	—	—	—	—	—	—
Battery Backup register	VBTBKR[n]	—	—	—	—	—	—	—	✓
VBATT Input Monitor Register	VBTIMONR	—	—	—	—	—	—	—	—

Table 5.3 Module-related registers initialized by each reset source (4 of 4)

Registers to be initialized		Reset source							
		Voltage monitoring 2 reset	Software reset	Bus error reset	Common memory error reset	Deep software standby reset			VBATT-selected voltage power-on reset
						Deep Software Standby mode 1 reset	Deep Software Standby mode 2 reset	Deep Software Standby mode 3 reset	
Independent Watchdog Timer registers	IWDTRR, IWDTCR, IWDTSR, IWDTRCR, IWDTCSTPR	✓	✓	✓	✓	—	✓	✓	—
Realtime Clock*1 register		—	—	—	—	—	—	—	—
Ultra-low-power timer registers	ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR, ULPTMR1, ULPTMR2, ULPTMR3, ULPTIOC, ULPTISR, ULPTCMSR	✓	✓	✓	✓	—	✓	✓	—
USBFS registers	DPUSR0R, DPUSR1R	✓	✓	✓	✓	—	✓	✓	—
USBHS registers	DPUSR0R, DPUSR1R, DPUSR2R, DPUSRCR	✓	✓	✓	✓	—	✓	✓	—
Reset Flag	BUSnERRADD (n = 4 to 9) BUSnERRRW (n = 4 to 9) BMSAnERRADD (n = 4 to 9) BMSAnERRRW (n = 4 to 9) BUSnERRSTAT (n = 1 to 12) MBWERRSTAT, SBWERRSTAT, SRAMESR, SRAMEARn (n = 0 to 2) STBRAMEAR	✓	✓	—	—	✓	✓	✓	—
Reset Flag		See Table 5.2							
Low Power Function registers	DPSBYCR, DPSWCR, DPSIER0 to DPSIER3, DPSIFR0 to DPSIFR3, DPSIEGR0 to DPSIEGR2 LPSCR FWEPROR SSCR1 LVOCR	✓	✓	✓	✓	—	—	—	—
Low Power Function registers	SYOCDCR	—	—	—	—	—	—	—	—
System Reset Mask Control Register *7	SYRSTMSK0, SYRSTMSK2	—	—	—	—	—	—	—	—
ARM Debug function	MCUSTAT,MCUCTRL, JBMDR,FSBLSTATM, DBGSTR,DBGSTOPCR, DBGAUTH0,DBGAUTH1, CACHEDBGCR,TRPORTCR	—	—	—	—	✓*3	✓*3	✓*3	—
TRCLK Control registers	TRCKCR	—	—	—	—	✓*3	✓*3	✓*3	—
Power Gating Control registers	PDRAMSCR0, PDRAMSCR1	—	—	—	—	✓	✓	✓	—
Other than specified		✓	✓	✓	✓	✓	✓	✓	—

Note: ✓ : Initialized
— : Not initialized

Note 1. RTC has Software reset for RTC. Some control bits are not initialized by all types of resets. For details on the target bits, section 24, Realtime Clock (RTC).

Note 2. Depend on setting of DPSBYCR.IOKEEP.

Note 3. After the debugger authentication passes, setting SYOCDCR.DBGEN to 1 does not initialize this register or the function. For details, see section 2, CPU.

SOSC and LOCO can be selected as the clock sources of the RTC.

Table 5.4 and Table 5.5 show the states of SOSC and LOCO when a reset occurs.

Table 5.4 States of SOSC when a reset occurs

		Reset source	
		VBATT_POR reset	Other
SOSC	Enable or disable	Initialized to enable	Continue with the state that was selected before the reset occurred
	Drive capability	Initialized to standard drive capability	

Table 5.5 States of LOCO when a reset occurs

		Reset source	
		Power-on reset, Voltage-Monitoring0 reset, Deep software standby 2,3 reset	Other
LOCO	Enable or disable	Initialized to enable	
	Oscillation accuracy*1	Initialized to accuracy before trimming by LOCOUTCR (accuracy: ± 15%)	Continue with the accuracy that was trimmed by LOCOUTCR

Note 1. If the LOCO that is trimmed by LOCOUTCR is selected as the RTC source clock, please be careful that LOCO oscillation accuracy will be initialized by Power-on reset, Voltage-Monitor 0 reset and Deep software standby 2, 3 reset.

When a reset is released, reset exception handling starts. For details on the reset exception handling, see section 5.3.12. Determination of Reset Generation Source.

Table 5.6 lists the pin related to the reset function.

For details on the reset exception handling, see section 5.3.12. Determination of Reset Generation Source.

Table 5.6 Pin related to reset

Pin name	I/O	Function
RES	Input	Reset pin

5.2 Register Descriptions

5.2.1 RSTSAR : Reset Security Attribution Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x3C4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	NONSEC2	NONSEC1	NONSEC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NONSEC0	Non-secure Attribute bit 0 Target register: Reset Status Register 0 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
1	NONSEC1	Non-secure Attribute bit 1 Target register: Reset Status Register 1 0: Secure 1: Non-secure	R/W
2	NONSEC2	Non-secure Attribute bit 2 Target register: Reset Status Register 2 0: Secure 1: Non-secure	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

NONSEC0 bit (Non-secure Attribute bit 0)

This bit controls the security attribute of RSTSR0.

NONSEC1 bit (Non-secure Attribute bit 1)

This bit controls the security attribute of RSTSR1.

NONSEC2 bit (Non-secure Attribute bit 2)

This bit controls the security attribute of RSTSR2.

5.2.2 RSTSR0 : Reset Status Register 0

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA40

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DPSR STF	—	—	—	PVD2 RF	PVD1 RF	PVD0 RF	PORF
Value after reset:	x*1	0	0	0	x*1	x*1	x*1	x*1

Bit	Symbol	Function	R/W
0	PORF	Power-On Reset Detect Flag 0: Power-on reset not detected 1: Power-on reset detected	R/W*2
1	PVD0RF	Voltage Monitor 0 Reset Detect Flag 0: Voltage monitor 0 reset not detected 1: Voltage monitor 0 reset detected	R/W*2
2	PVD1RF	Voltage Monitor 1 Reset Detect Flag 0: Voltage monitor 1 reset not detected 1: Voltage monitor 1 reset detected	R/W*2
3	PVD2RF	Voltage Monitor 2 Reset Detect Flag 0: Voltage monitor 2 reset not detected 1: Voltage monitor 2 reset detected	R/W*2
6:4	—	These bits are read as 0. The write value should be 0.	R/W
7	DPSRSTF	Deep Software Standby Reset Flag 0: Deep Software Standby mode cancellation not requested by an interrupt or a reset*3. 1: Deep Software Standby mode cancellation requested by an interrupt or a reset*3.	R/W*2

Note: S-TYPE-3, P-TYPE-2
Note 1. The value after reset depends on the reset source.
Note 2. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 after 1 is read.
Note 3. Independent watchdog timer reset, Voltage Monitor 1 reset, Voltage Monitor 2 reset.

PORF flag (Power-On Reset Detect Flag)

The PORF flag indicates that a power-on reset occurred.

[Setting condition]

- When a power-on reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When 0 is written after 1 is read from PORF flag.

PVD0RF flag (Voltage Monitor 0 Reset Detect Flag)

The PVD0RF flag indicates that the Voltage Monitor 0 reset has occurred.

[Setting condition]

- When a voltage monitor 0 reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When 0 is written after 1 is read from PVD0RF flag.

PVD1RF flag (Voltage Monitor 1 Reset Detect Flag)

The PVD1RF flag indicates that the Voltage Monitor 1 reset has occurred.

[Setting condition]

- When a voltage monitor 1 reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When 0 is written after 1 is read from PVD1RF flag.

PVD2RF flag (Voltage Monitor 2 Reset Detect Flag)

The PVD2RF flag indicates that the Voltage Monitor 2 reset has occurred.

[Setting condition]

- When a voltage monitor 2 reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When 0 is written after 1 is read from PVD2RF flag.

DPSRSTF flag (Deep Software Standby Reset Flag)

The DPSRSTF flag indicates that Deep Software Standby mode is canceled by an external or internal interrupt, and that an internal reset (Deep Software Standby reset) occurs when an exception from Deep Software Standby mode occurs.

[Setting condition]

- When Deep Software Standby mode is cancelled by an external or an internal interrupt. For details, see section 10, Low Power Modes.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs.
- When 0 is written after 1 is read from DPSRSTF flag.

5.2.3 RSTSR1 : Reset Status Register 1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CMRF	—	—	—	BUSRF	—	—	—	—	—	CLU0RF	—	SWRF	WDT0RF	IWDTRF
Value after reset:	0	x*1	0	0	0	x*1	0	0	0	0	0	x*1	0	x*1	x*1	x*1

Bit	Symbol	Function	R/W
0	IWDTRF	Independent Watchdog Timer Reset Detect Flag 0: Independent watchdog timer reset not detected 1: Independent watchdog timer reset detected	R/W*2
1	WDT0RF	Watchdog Timer Reset Detect Flag 0: Watchdog timer reset not detected 1: Watchdog timer reset detected	R/W*2
2	SWRF	Software Reset Detect Flag 0: Software reset not detected 1: Software reset detected	R/W*2
3	—	This bit is read as 0. The write value should be 0.	
4	CLU0RF	CPU Lockup Reset Detect Flag 0: CPU Lockup reset not detected 1: CPU Lockup reset detected	R/W*2
9:5	—	These bits are read as 0. The write value should be 0.	R/W
10	BUSRF	Bus Error Reset Detect Flag 0: Bus error reset not detected 1: Bus error reset detected	R/W*2
13:11	—	These bits are read as 0. The write value should be 0.	R/W
14	CMRF	Common Memory Error Reset Detect Flag 0: Common memory error reset not detected 1: Common memory error reset detected	R/W*2
31:15	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note 1. The value after reset depends on the reset source.
Note 2. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 after 1 is read.

IWDTRF flag (Independent Watchdog Timer Reset Detect Flag)

The IWDTRF flag indicates that an Independent watchdog timer reset occurred.

[Setting condition]

- When an independent watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When 0 is written after 1 is read from IWDTRF flag.

WDT0RF flag (Watchdog Timer Reset Detect Flag)

The WDT0RF flag indicates that a Watchdog timer reset occurred.

[Setting condition]

- When a watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When 0 is written after 1 is read from WDT0RF flag.

SWRF flag (Software Reset Detect Flag)

The SWRF flag indicates that a Software reset occurred.

[Setting condition]

- When a software reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When 0 is written after 1 is read from SWRF flag.

CLU0RF flag (CPU Lockup Reset Detect Flag)

The CLU0RF flag indicates that a CPU Lockup reset occurred.

[Setting condition]

- When a CPU Lockup reset occurs

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When 0 is written after 1 is read from CLU0RF flag.

BUSRF flag (Bus Error Reset Detect Flag)

The BUSRF flag indicates that BUS error reset (MSAU error, MMPU error, Illegal address error, Slave TrustZone Filter error, Slave Bus error, Bufferable write error) occurred.

[Setting condition]

- When a bus error reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When 0 is written after 1 is read from BUSRF flag.

CMRF flag (Common Memory Error Reset Detect Flag)

The CMRF flag indicates that a Common memory error reset (ECC error or Parity error of SRAM and Standby SRAM) occurred.

[Setting condition]

- When a Common memory error reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When 0 is written after 1 is read from CMRF flag.

There are several causes for setting this reset flag.

If you want to identify the reset factor, see section 13, Interrupt Controller Unit (ICU).

5.2.4 RSTSR2 : Reset Status Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA44

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CWSF
Value after reset:	0	0	0	0	0	0	0	x*1

Bit	Symbol	Function	R/W
0	CWSF	Cold/Warm Start Determination Flag 0: Cold start 1: Warm start	R/W*2
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note 1. The value after reset depends on the reset source.
Note 2. Only 1 can be written to set the flag.

RSTSR2 determines whether a power-on reset caused the reset processing (cold start) or a reset signal input during operation caused the reset processing (warm start).

CWSF flag (Cold/Warm Start Determination Flag)

The CWSF flag indicates the type of reset processing, either cold start or warm start. CWSF flag is initialized by a power-on reset. It is not initialized by a reset signal generated by the RES pin.

[Setting condition]

- When 1 is written by software. Writing 0 to CWSF does not set it to 0.

[Clearing condition]

- When a reset listed in Table 5.2 occurs.

5.2.5 SYRSTMSK0 : System Reset Mask Control Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0xAD0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	BUSMASK	CMMASK	—	CLU0MASK	—	SWMASK	WDT0MASK	IWDTMASK
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTMASK	Independent Watchdog Timer Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
1	WDT0MASK	Watchdog Timer Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
2	SWMASK	Software Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	CLU0MASK	CPU Lockup Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W

Bit	Symbol	Function	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	CMMASK	Common Memory Error Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
7	BUSMASK	Bus Error Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W

Note: S-TYPE-6, P-TYPE-2
Note: Set the PRCR.PRC5 bit to 1 (write enabled) before rewriting this register.

The SYRSTMSK0 is a register that controls the occurrence of reset.

The IWDTMASK bit cannot be rewritten while the independent watchdog timer is operating. The WDT0MASK bit cannot be rewritten while the watchdog timer is operating.

5.2.6 SYRSTMSK2 : System Reset Mask Control Register 2

Base address: SYSC = 0x4001_E000

Offset address: 0xAD8

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PVD2 MASK	PVD1 MASK
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PVD1MASK	Voltage Monitor 1 Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
1	PVD2MASK	Voltage Monitor 2 Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6, P-TYPE-2
Note: Set the PRCR.PRC5 bit to 1 (write enabled) before rewriting this register.

The SYRSTMSK2 is a register that controls the occurrence of reset.

5.3 Operation

5.3.1 RES Pin Reset

The RES pin generates this reset. When the RES pin is driven low, all the processing in progress is aborted and the MCU enters a reset state. To successfully reset the MCU, the RES pin must be held low for the power supply stabilization time specified at power-on.

When the RES pin is driven high from low, the internal reset is canceled after the post-RES cancellation wait time (t_{RESWT}) elapses. The CPU then starts the reset exception handling.

For details, see section 60, Electrical Characteristics.

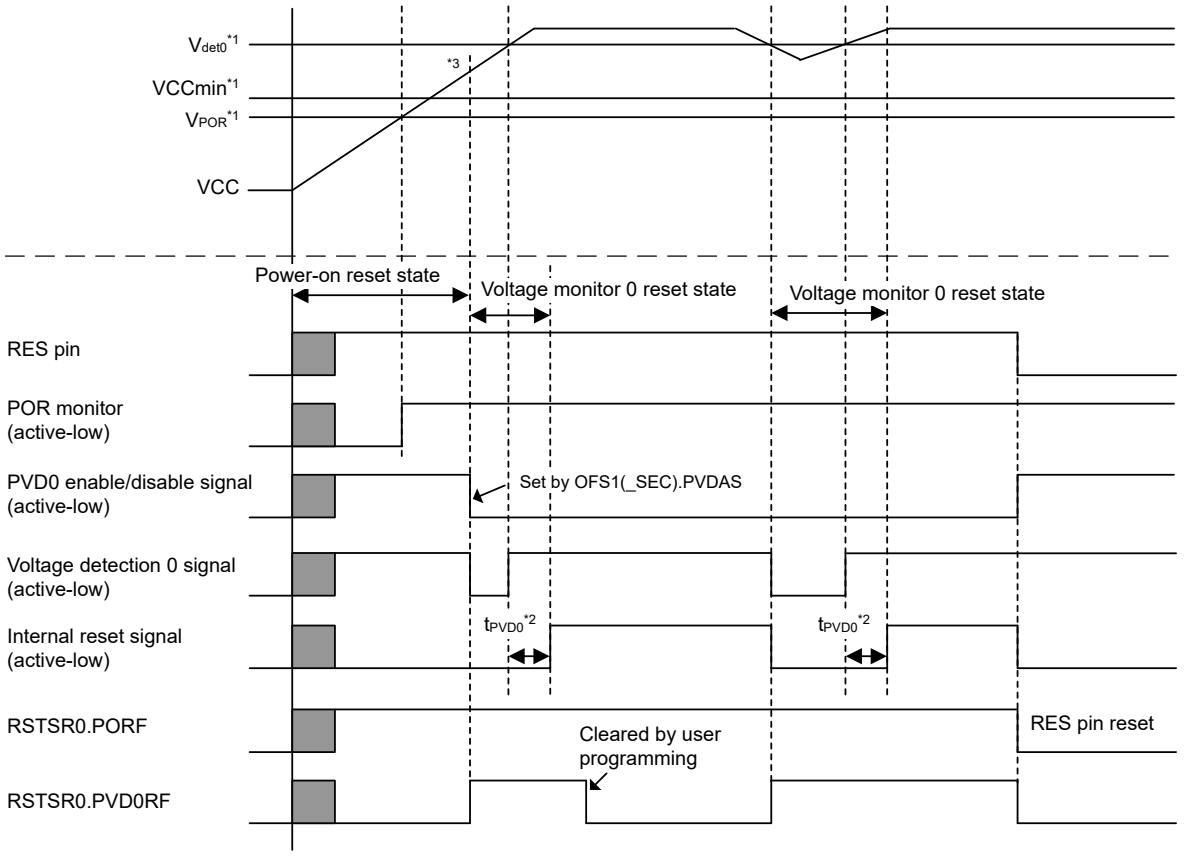
5.3.2 Power-On Reset

The power-on reset circuit generates this internal reset. If the RES pin is in a high-level state when power is supplied, a power-on reset is generated.

After VCC exceeds V_{POR} and the specified period (power-on reset time) elapses, the internal reset is canceled, and the CPU starts the reset exception handling. The power-on reset time is the stabilization period for the external power supply and the MCU circuit.

After a power-on reset is generated, the PORF flag in the RSTSR0 is set to 1. The PORF flag is initialized by the RES pin reset.

Figure 5.1 shows examples of operations during a power-on reset and voltage monitor 0 reset.



Note: For details on the electrical characteristics, see section 60, Electrical Characteristics.
Note 1. V_{det0} shows a voltage monitor 0 reset detection level, V_{POR} shows a power-on reset detection level, and VCCmin shows minimum guaranteed voltage of MCU.
Note 2. tPVD0 shows the reset time of Voltage Monitor 0.
Note 3. At power-on, VCC should rise to the minimum guaranteed voltage before the power-on reset is released.

Figure 5.1 Example of operations during a power-on reset and Voltage Monitor 0 Reset

5.3.3 Voltage Monitor Reset

The voltage monitor 0 reset is an internal reset generated by the voltage monitor circuit. If the Voltage Detection 0 Circuit Start bit (PVDAS) in Option Function Select Register 1 (OFS1) is 0 (voltage monitor 0 reset is enabled after a reset) and VCC falls below V_{det0}, the RSTSR0.PVD0RF flag is set to 1 and the voltage detection circuit generates a voltage monitor 0 reset. Clear the OFS1(_SEC).PVDAS bit to 0 if the voltage monitor 0 reset is to be used. After VCC exceeds V_{det0} and the voltage monitor 0 reset time (t_{PVD0}) elapses, the internal reset is canceled, and the CPU starts the reset exception handling. The V_{det0} voltage detection level can be changed by the setting in the VDSEL[1:0] bits in Option Function Select Register 1 (OFS1).

- Case of PVDmFCR.RHSEL = 0

When the Voltage Monitor 1 Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the Voltage Monitor 1 Circuit Mode Select bit (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 1 Circuit Control Register 0 (PVD1CR0), the RSTSR0.PVD1RF flag is set to 1 and the voltage detection circuit generates a Voltage Monitor 1 reset if VCC falls to or below Vdet1.

Similarly, timing for release from the Voltage Monitor 1 reset state is selectable in the Voltage Monitor 1 Reset Negate Select bit (RN) in PVD1CR0. When the RN bit is 0 and VCC falls to or below Vdet1, the CPU is released from the internal reset state and starts reset exception handling when the PVD1 reset time (t_{PVD1}) elapses after VCC rises above Vdet1. When the PVD1CR0.RN bit is 1 and VCC falls to or below Vdet1, the CPU is released from the internal reset state and starts reset exception handling when the PVD1 reset time ($_{tPVD1}$) elapses.

Likewise, when the Voltage Monitor 2 Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the Voltage Monitor 2 Circuit Mode Select bit (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 2 Circuit Control Register 0 (PVD2CR0), the RSTSR0.PVD2RF flag is set to 1 and the voltage detection circuit generates a Voltage Monitor 2 reset if VCC falls to or below Vdet2.

Similarly, timing for release from the Voltage Monitor 2 reset state is selectable in the Voltage Monitor 2 Reset Negate Select bit (RN) in PVD2CR0. When the RN bit is 0 and VCC falls to or below Vdet2, the CPU is released from the internal reset state and starts reset exception handling when the PVD2 reset time (t_{PVD2}) elapses after VCC rises above Vdet2. When the PVD2CR0.RN bit is 1 and VCC falls to or below Vdet2, the CPU is released from the internal reset state and starts reset exception handling when the PVD2 reset time (t_{PVD2}) elapses.

- Case of PVDmFCR.RHSEL = 1.

When the Voltage Monitor 1 Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the Voltage Monitor 1 Circuit Mode Select bit (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 1 Circuit Control Register 0 (PVD1CR0), the RSTSR0.PVD1RF flag is set to 1 and the voltage detection circuit generates a Voltage Monitor 1 reset if VCC rises above Vdet1.

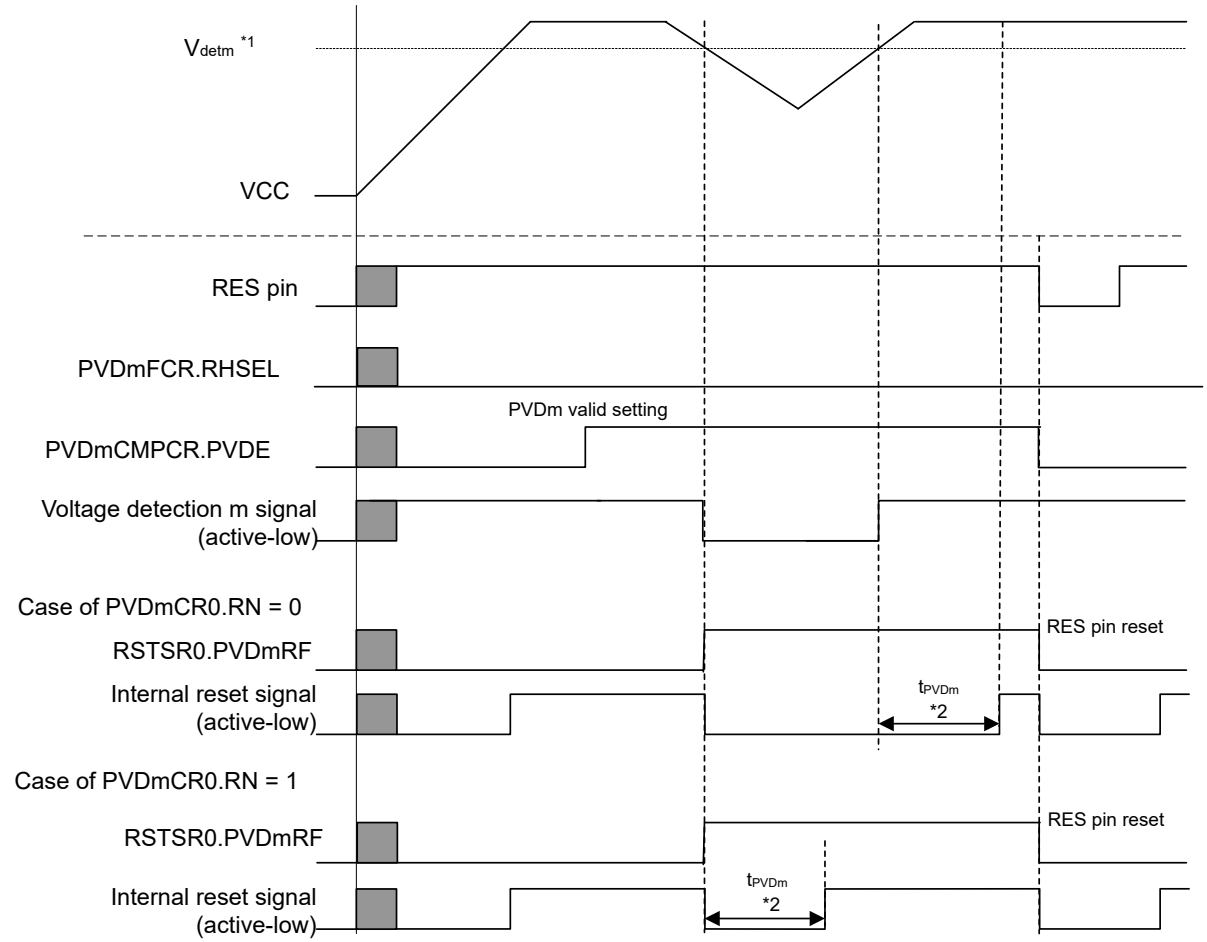
Then the CPU is released from the internal reset state and starts reset exception handling when the PVD1 reset time (t_{PVD1}) elapses after VCC falls to or below Vdet1.

Likewise, when the Voltage Monitor 2 Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the Voltage Monitor 2 Circuit Mode Select bit (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 2 Circuit Control Register 0 (PVD2CR0), the RSTSR0.PVD2RF flag is set to 1 and the voltage detection circuit generates a Voltage Monitor 2 reset if VCC rises above Vdet2.

Then the CPU is released from the internal reset state and starts reset exception handling when the PVD2 reset time (t_{PVD2}) elapses after VCC falls to or below Vdet2.

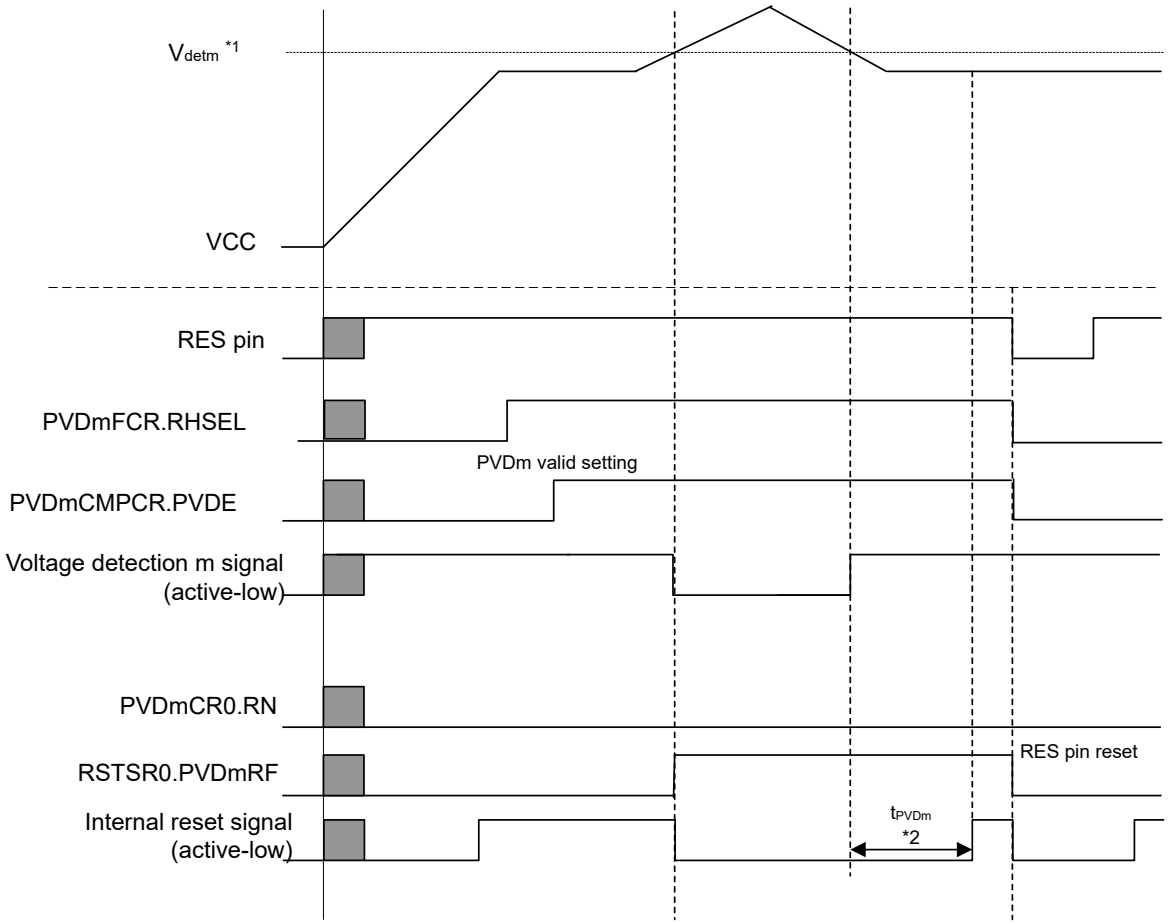
Figure 5.2 shows examples of operations during the voltage monitor 1 and 2 reset when PVDmFCR.RHSEL=0.

Figure 5.3 shows examples of operations during the voltage monitor 1 and 2 reset when PVDmFCR.RHSEL=1.



Note: For details on the electrical characteristics, see section 60, Electrical Characteristics.
Note 1. V_{detm} indicates the detection level of the voltage monitor m reset. (m = 1, 2)
Note 2. t_{PVDm} indicates the time for the voltage monitor m reset. (m = 1, 2)

Figure 5.2 Example of operations during voltage monitor m (m = 1, 2) reset when PVDmFCR.RHSEL = 0



Note: For details on the electrical characteristics, see section 60, Electrical Characteristics.
Note 1. V_{detm} indicates the detection level of the voltage monitor m reset. (m = 1, 2)
Note 2. t_{PVDm} indicates the time for the voltage monitor m reset. (m = 1, 2)

Figure 5.3 Example of operations during voltage monitor m (m = 1, 2) reset when PVDmFCR.RHSEL = 1

5.3.4 Deep Software Standby Reset

These internal resets are generated when either of the Deep Software Standby mode are canceled by an associated interrupt.

The Deep Software Standby reset is canceled after t_{DSBY} (return time after Deep Software Standby mode cancellation) elapses. At the same time, Deep Software Standby mode is also canceled.

When t_{DSBYWT} (wait time after Deep Software Standby mode cancellation) elapses after Deep Software Standby mode has been canceled, the internal reset is canceled, and the CPU starts the reset exception handling.

For details of the Deep Software Standby reset, see section 10, Low Power Modes.

5.3.5 Independent Watchdog Timer Reset

The independent watchdog timer reset is an internal reset generated from the Independent Watchdog Timer (IWDT). Output of the reset from the IWDT can be selected in the Option Function Select Register 0 (OFS0).

When output of the independent watchdog timer reset is selected, the reset is generated if the IWDT underflows, or if data is written when refresh operation is disabled. When the internal reset time (t_{RESW2}) elapses after the independent watchdog timer reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

For details on the independent watchdog timer reset, see section 26, Independent Watchdog Timer (IWDT).

5.3.6 Watchdog Timer Reset

The watchdog timer reset is an internal reset generated from the Watchdog Timer (WDT). Output of the reset from the WDT can be selected in the WDT Reset Control Register (WDTRCR) or Option Function Select register 0 (OFS0).

When output of the watchdog timer reset is selected, the reset is generated if the WDT underflows, or if data is written when refresh operation is disabled. When the internal reset time (t_{RESW2}) elapses after the watchdog timer reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

For details on the watchdog timer reset, see section 25, Watchdog Timer (WDT).

5.3.7 CPU Lockup Reset

The CPU Lockup reset is an internal reset generated by Arm core. Output of the reset from Arm core can be selected by CPULCKUPCR.OAD.

When output of the CPU Lockup reset is selected, CPU generated if Arm core is in the lockup state. When the internal reset time (t_{RESW2}) elapses after the CPU Lockup reset is generated, the internal reset is cancelled, and the CPU starts the reset exception handling.

For details on the CPU Lockup reset, see section 2, CPU.

5.3.8 Software Reset

This internal reset is generated by a software setting of the SYSRESETREQ bit in the AIRCR register in the Arm core. When the SYSRESETREQ bit is set to 1, a software reset is generated. When the internal reset time (t_{RESW2}) elapses after the software reset is generated, the internal reset is canceled, and the CPU starts the reset exception handling.

5.3.9 Bus Error Reset

Bus error reset is an internal reset generated by buses.

It is an integrated reset generated by buses, and it consists of the following:

- Bus error reset (MSAU error, MMPU error, Illegal address error, Slave TrustZone Filter error, Slave Bus error, Bufferable write error)

Output of Bus error reset can be selected by OADCFG.OAD.

When the internal reset time (t_{RESW2}) elapses after the Bus error reset is generated, the internal reset is cancelled, and the CPU starts the reset exception handling.

For details on the Bus error reset, see section 14, Buses. Determination of reset generation source in Bus error reset is similar to determination of interrupt generation source, see section 13, Interrupt Controller Unit (ICU).

5.3.10 Common Memory Error Reset

Common memory error reset is an internal reset generated by SRAMs.

It is an integrated reset generated by SRAMs and consists of the following:

- SRAM error reset (ECC error, Parity error)
- Standby SRAM error reset (Parity error)

Output of SRAM error reset can be selected by SRAMCR0.OAD or SRAMCR1.OAD. Output of Standby SRAM error reset can be selected by STBRAMCR.OAD.

When the internal reset time (t_{RESW2}) elapses after the Common memory error reset is generated, the internal reset is cancelled, and the CPU starts the reset exception handling.

For details on the Common memory error reset, see section 14, Buses. Determination of reset generation source in Common memory error reset is similar to determination of interrupt generation source, see section 13, Interrupt Controller Unit (ICU).

5.3.11 Determination of Cold/Warm Start

Read the CWSF flag in RSTSR2 to determine the cause of reset processing. This flag indicates whether a power-on reset caused the reset processing (cold start) or a reset signal input during operation caused the reset processing (warm start).

The CWSF flag is set to 0 when a power-on reset occurs (cold start), otherwise the flag is not set to 0. The flag is set to 1 when 1 is written to it through software. It is not set to 0 even on writing 0 to it.

Figure 5.4 shows an example of cold/warm start determination operation.

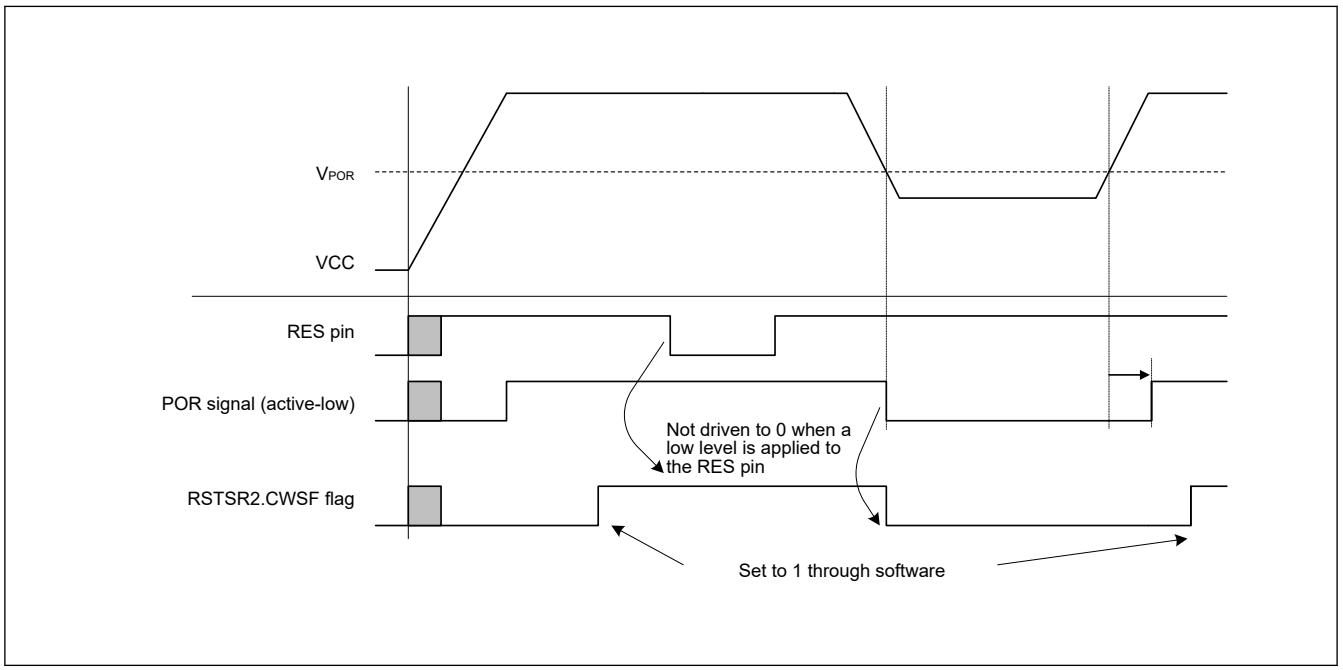
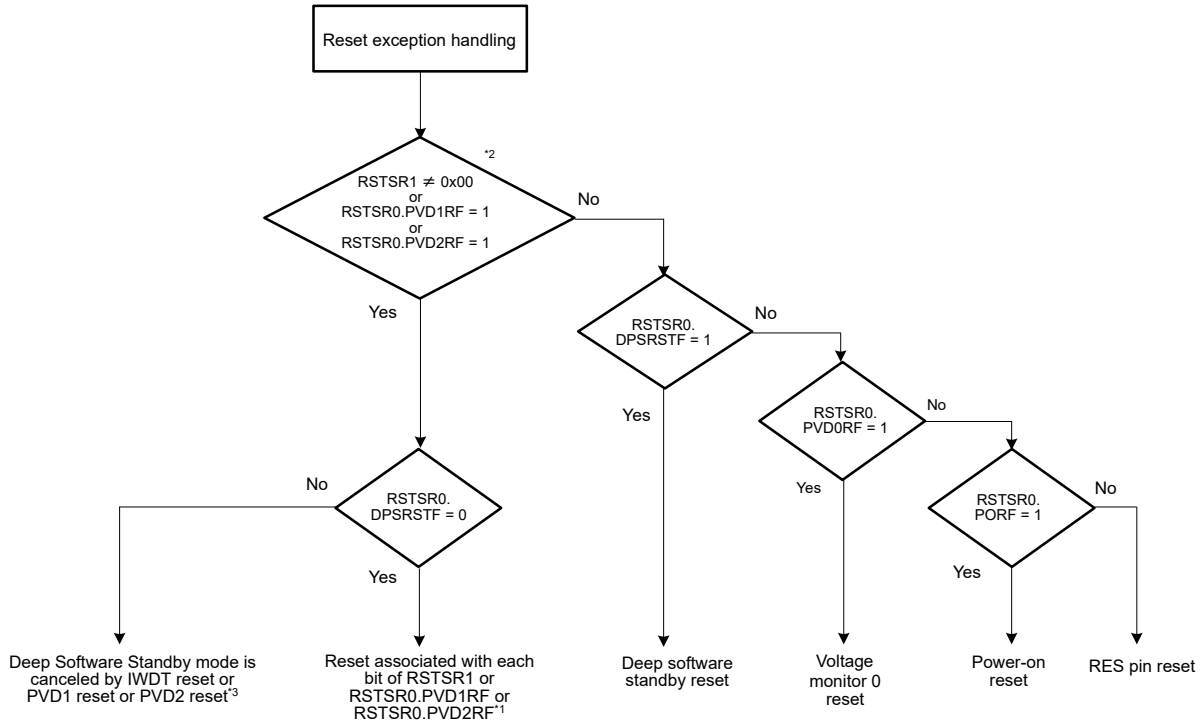


Figure 5.4 Example of cold/warm start determination operation

5.3.12 Determination of Reset Generation Source

Read RSTSR0 and RSTSR1 to determine which reset executes the reset exception handling.

Figure 5.5 shows an example of the flow to identify a reset generation source. The reset flag must be written with 0 after 1 is read.



Note 1. If a reset associated with each bit of RSTSR1, RSTSR0.PVD1RF, or RSTSR0.PVD2RF occurred at the same time, all occurrences of the reset flags are set to 1.
Note 2. After checking RSTSR1 ≠ 0x00, RSTSR0.PVD1RF = 1 or RSTSR0.PVD2RF = 1, check RSTSR0.DPSRSTF to determine whether it caused the cancellation of Deep Software Standby mode.
Note 3. The Reset Status registers (RSTSR0, RSTSR1 and RSTSR2) should be cleared before the Deep Software Standby transition. If the reset flag has not been cleared, the return factor cannot be correctly identified.

Figure 5.5 Example of reset generation source determination flow

5.4 Reset Protection from Non-secure Programs

The reset occurrence can be controlled. A reset is generated according to the setting of SYRSTMSK0 and SYRSTMSK2 that can only be accessed from a Secure program.

The reset source is generated but is not propagated to the MCU. The reset flags of RSTSR0 and RSTSR1 bits are not set.

The resets that can be controlled are as follows:

- Independent watchdog timer reset
- Watchdog timer reset
- CPU Lockup reset
- Common memory error reset
- Bus error reset
- Voltage monitor 1 reset
- Voltage monitor 2 reset
- Software reset

6. Option-Setting Memory

6.1 Overview

The option-setting memory determines the state of the MCU after a reset. The option-setting memory is allocated to the configuration setting area of the flash memory.

Figure 6.1 shows the option-setting memory area. The option-setting memory area has Secure region and Non-secure region. Table 6.1 shows the programming condition of the option-setting memory area.

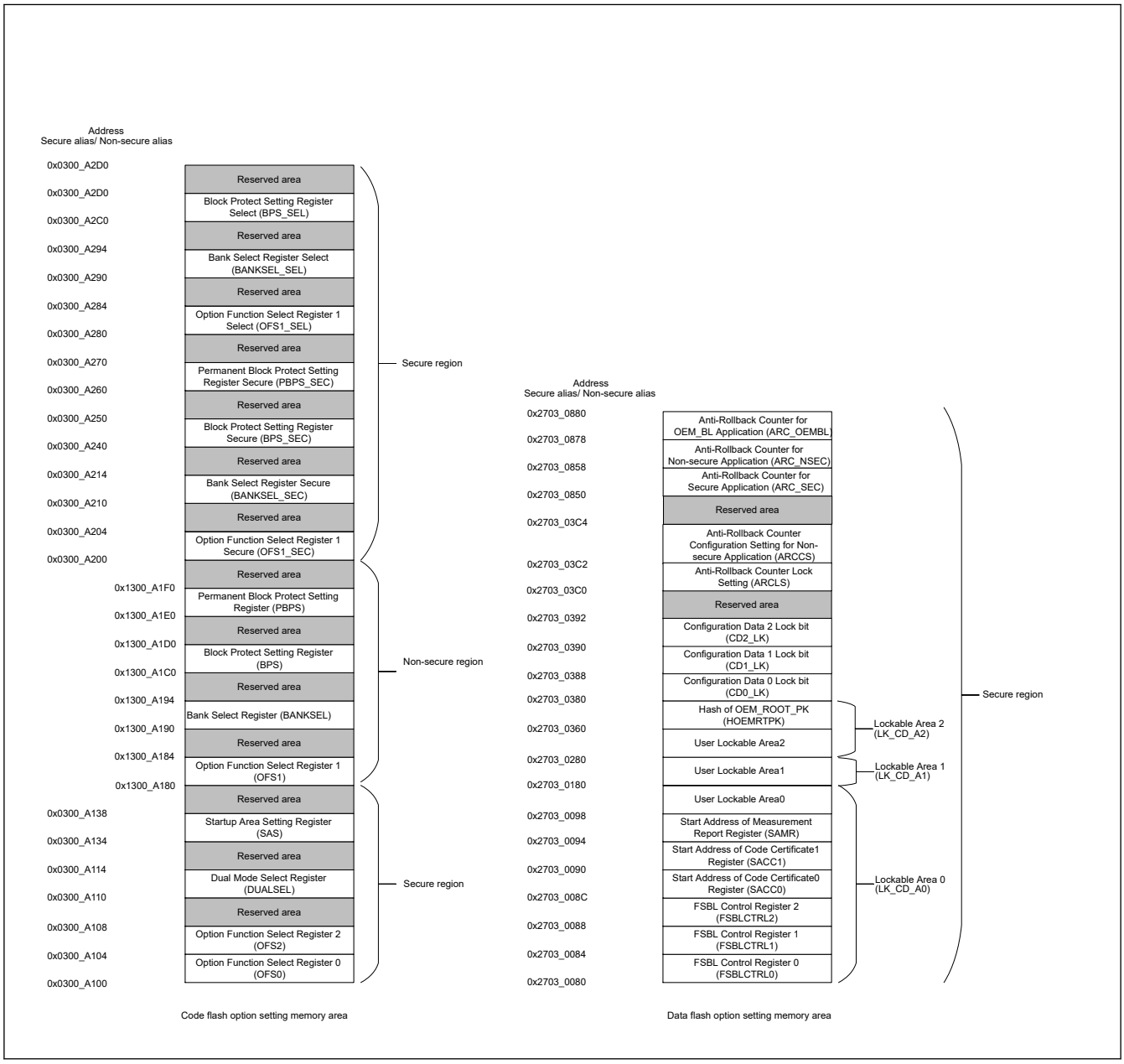


Figure 6.1 Option-setting memory area

Table 6.1 The programming condition of the option-setting memory area (1 of 2)

	Self programming	Serial programming	Programming by the on-chip debugger
Secure region	Programming commands issued via Secure alias access.	Programming commands issued when the authentication level is AL2.	Programming commands issued when the authentication level is AL2.

Table 6.1 The programming condition of the option-setting memory area (2 of 2)

	Self programming	Serial programming	Programming by the on-chip debugger
Non-secure region	Programming commands issued via Non-secure alias access.	Programming commands issued when the authentication level is AL2 or AL1.	Programming commands issued when the authentication level is AL2 or AL1.

6.2 Register Descriptions

6.2.1 OFS0 : Option Function Select Register 0

Address: 0x0300_A100

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	WDT0 STPC TL	—	WDT0 RSTIR QS	WDT0RPSS [1:0]		WDT0RPES [1:0]		WDT0CKS[3:0]				WDT0TOPS [1:0]		WDT0 STRT	—
Value after reset:	User setting*1															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	IWDT STPC TL	—	IWDT RSTIR QS	IWDTRPSS[1:0]		IWDTRPES[1:0]		IWDTCKS[3:0]				IWDTTOPS[1:0]		IWDT STRT	—
Value after reset:	User setting*1															

Bit	Symbol	Function	R/W
0	—	The program value is read from this bit.*2	R
1	IWDTSTRT	IWDT Start Mode Select 0: Automatically activate IWDT after a reset (auto start mode) 1: Stop IWDT after a reset (register start mode)	R
3:2	IWDTTOPS[1:0]	IWDT Timeout Period Select 0 0: 128 cycles (0x007F) 0 1: 512 cycles (0x01FF) 1 0: 1024 cycles (0x03FF) 1 1: 2048 cycles (0x07FF)	R
7:4	IWDTCKS[3:0]	IWDT-Dedicated Clock Frequency Division Ratio Select 0x0: × 1 0x2: × 1/16 0x3: × 1/32 0x4: × 1/64 0xF: × 1/128 0x5: × 1/256 Others: Reserved	R
9:8	IWDTRPES[1:0]	IWDT Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (no window end position setting)	R
11:10	IWDTRPSS[1:0]	IWDT Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (no window start position setting)	R
12	IWDTRSTIRQS	IWDT Reset Interrupt Request Select 0: Interrupt 1: Reset	R
13	—	The program value is read from this bit.*2	R

Bit	Symbol	Function	R/W
14	IWDTSTPCTL	IWDT Stop Control 0: Continue counting 1: Stop counting when in CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode or Deep Software Standby mode1	R
16:15	—	The program value is read from these bits.*2	R
17	WDT0STRT	WDT Start Mode Select 0: Automatically activate WDT after a reset (auto start mode) 1: Stop WDT after a reset (register start mode)	R
19:18	WDT0TOPS[1:0]	WDT Timeout Period Select 0 0: 1024 cycles (0x03FF) 0 1: 4096 cycles (0x0FFF) 1 0: 8192 cycles (0x1FFF) 1 1: 16384 cycles (0x3FFF)	R
23:20	WDT0CKS[3:0]	WDT Clock Frequency Division Ratio Select 0x1: PCLKB divided by 4 0x4: PCLKB divided by 64 0xF: PCLKB divided by 128 0x6: PCLKB divided by 512 0x7: PCLKB divided by 2048 0x8: PCLKB divided by 8192 Others: Reserved	R
25:24	WDT0RPES[1:0]	WDT Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (no window end position setting)	R
27:26	WDT0RPSS[1:0]	WDT Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (no window start position setting)	R
28	WDT0RSTIRQS	WDT Reset Interrupt Request Select 0: Interrupt 1: Reset	R
29	—	The program value is read from this bit.*2	R
30	WDT0STPCTL	WDT Stop Control 0: Continue counting 1: Stop counting when entering CPU Sleep mode or CPU Deep Sleep mode	R
31	—	The program value is read from this bit.*2	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.
Note 2. This register can only be programmed by FACI command. When programming, the set value should be 1.

IWDTSTRT bit (IWDT Start Mode Select)

The IWDTSTRT bit selects the mode in which the IWDT is activated after a reset (stopped state or activated state). When IWDT is activated in auto start mode, the OFS0 register setting for the IWDT is valid.

IWDTTOPS[1:0] bits (IWDT Timeout Period Select)

The IWDTTOPS[1:0] bits specify the timeout period, that is, the time it takes for the down counter to underflow, as 128, 512, 1024, or 2048 cycles of the frequency-divided clock set in the IWDTCKS[3:0] bits. The time it takes for the counter to underflow after a refresh operation is determined by the combination of the IWDTCKS[3:0] and IWDTTOPS[1:0] bits.

For details, see section 26, Independent Watchdog Timer (IWDT).

IWDTCKS[3:0] bits (IWDT-Dedicated Clock Frequency Division Ratio Select)

The IWDTCKS[3:0] bits specify the division ratio of the prescaler for dividing the frequency of the clock for the IWDT as 1/1, 1/16, 1/32, 1/64, 1/128, and 1/256. Using this setting combined with the IWDTTOPS[1:0] bits setting, the IWDT counting period can be set from 128 to 524,288 IWDT clock cycles.

For details, see section 26, Independent Watchdog Timer (IWDT).

IWDTRPES[1:0] bits (IWDT Window End Position Select)

The IWDTRPES[1:0] bits specify the position where the window for the down counter ends as 0%, 25%, 50%, or 75% of the count value. The value of the window end position must be smaller than the value of the window start position, otherwise only the value for the window start position is valid.

The counter values associated with the settings for the start and end positions of the window in the IWDTRPSS[1:0] and IWDTRPES[1:0] bits vary with the setting in the IWDTTOPS[1:0] bits.

For details, see section 26, Independent Watchdog Timer (IWDT).

IWDTRPSS[1:0] bits (IWDT Window Start Position Select)

The IWDTRPSS[1:0] bits specify the position where the window for the down counter starts as 25%, 50%, 75%, or 100% of the counted value. The point at which counting starts is 100% and the point at which an underflow occurs is 0%. The interval between the window starts and ends positions becomes the period in which a refresh is possible. Refresh is not possible outside this period.

For details, see section 26, Independent Watchdog Timer (IWDT).

IWDTRSTIRQS bit (IWDT Reset Interrupt Request Select)

The IWDTRSTIRQS bit selects the operation on an underflow of the down counter or generation of a refresh error. The operation is selectable to an independent watchdog timer reset, a non-maskable interrupt request, or an interrupt request.

For details, see section 26, Independent Watchdog Timer (IWDT).

IWDTSTPCTL bit (IWDT Stop Control)

The IWDTSTPCTL bit specifies whether to stop counting when entering CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode or Deep Software Standby mode1.

For details, see section 26, Independent Watchdog Timer (IWDT).

WDT0STRT bit (WDT Start Mode Select)

The WDT0STRT bit selects the mode in which the WDT is activated after a reset (stopped state or activated in auto start mode). When WDT is activated in auto start mode, the OFS0 register setting for the WDT is valid.

WDT0TOPS[1:0] bits (WDT Timeout Period Select)

The WDT0TOPS[1:0] bits specify the timeout period, that is, the time it takes for the down counter to underflow as 1024, 4096, 8192, or 16384 cycles of the frequency-divided clock set in the WDT0CKS[3:0] bits. The number of PCLKB cycles that takes to underflow after a refresh operation is determined by a combination of the WDT0CKS[3:0] and WDT0TOPS[1:0] bits.

For details, see section 25, Watchdog Timer (WDT).

WDT0CKS[3:0] bits (WDT Clock Frequency Division Ratio Select)

The WDT0CKS[3:0] bits specify the division ratio of the prescaler for dividing the frequency of PCLKB as 1/4, 1/64, 1/128, 1/512, 1/2048, and 1/8192. Using this setting combined with the WDT0TOPS[1:0] bits setting, the WDT counting period can be set from 4,096 to 134,217,728 PCLKB cycles.

For details, see section 25, Watchdog Timer (WDT).

WDT0RPES[1:0] bits (WDT Window End Position Select)

The WDT0RPES[1:0] bits specify the position where the window on the down counter ends as 0%, 25%, 50%, or 75% of the counted value. The value of the window end position must be smaller than the value of the window start position, otherwise only the value for the window start position is valid.

The counter values associated with the settings for the start and end positions of the window in the WDT0RPSS[1:0] and WDT0RPES[1:0] bits vary with the setting of the WDT0TOPS[1:0] bits.

For details, see section 25, Watchdog Timer (WDT).

WDT0RPSS[1:0] bits (WDT Window Start Position Select)

The WDT0RPSS[1:0] bits specify the position where the window for the down counter starts as 25%, 50%, 75%, or 100% of the counted value. The point at which counting starts is 100% and the point at which an underflow occurs is 0%. The interval between the positions where the window starts and ends becomes the period in which a refresh is possible.

Refresh is not possible outside this period.

For details, see section 25, Watchdog Timer (WDT).

WDT0RSTIRQS bit (WDT Reset Interrupt Request Select)

The WDT0RSTIRQS bit selects the operation on an underflow of the down-counter or generation of a refresh error. The operation is selectable to a watchdog timer reset, a non-maskable interrupt request, or an interrupt request.

For details, see section 25, Watchdog Timer (WDT).

WDT0STPCTL bit (WDT Stop Control)

The WDT0STPCTL bit specifies whether to stop counting when entering CPU Sleep mode or CPU Deep Sleep mode.

For details, see section 25, Watchdog Timer (WDT).

6.2.2 OFS2 : Option Function Select Register 2

address: 0x0300_A104

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	User setting*1															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DCDCEN
Value after reset:	User setting*1															

Bit	Symbol	Function	R/W
0	DCDCEN	DCDC enable 0: Disable DCDC 1: Enable DCDC	R
31:1	—	The program value is read from these bits.*2	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.
Note 2. This register can only be programmed by FACI command. When programming, the set value should be 1.

DCDCEN bit (DCDC enable)

The DCDCEN bit selects whether the DCDC is enabled or disabled.

6.2.3 DUALSEL : Dual Mode Select Register

address: 0x0300_A110

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	User setting*1															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	BANKMD[2:0]		
Value after reset:	User setting*1															

Bit	Symbol	Function	R/W
2:0	BANKMD[2:0]	Bank Mode Select 0 0 0: Dual mode 1 1 1: Linear mode Others: Reserved	R
31:3	—	The program value is read from these bits.*2	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application
Note 2. This register can only be programmed by FACI command. When programming, the set value should be 1.

BANKMD[2:0] bit (Bank Mode Select)

The BANKMD[2:0] bits select bank mode of the dual bank function of the code flash memory

6.2.4 SAS : Startup Area Setting Register

Address: 0x0300_A134

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BTFLG	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	User setting*1															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FSPR	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	User setting*1															

Bit	Symbol	Function	R/W
14:0	—	The program value is read from these bits.*2	R
15	FSPR	Protection of Startup Area Select Function This bit controls the programming of the write/erase protection for the Startup Area Select flag (SAS.BTFLG), and the temporary boot swap control. When this bit is set to 0, it cannot be changed to 1. 0: Executing the configuration setting command for programming the Startup Area Select flag (SAS.BTFLG) is invalid. 1: Executing the configuration setting command for programming the Startup Area Select flag (SAS.BTFLG) is valid.	R
30:16	—	The program value is read from these bits.*2	R
31	BTFLG	Startup Area Select Flag This bit specifies whether the address of the startup area is exchanged for the boot swap function or not. In dual mode (the DUALSEL.BANKMD[2:0] bits are 000b), set 1 to this bit. 0: First 8 KB area (0x0200_0000 to 0x0200_1FFF) and second 8 KB area (0x0200_2000 to 0x0200_3FFF) are exchanged. 1: First 8 KB area (0x0200_0000 to 0x0200_1FFF) and second 8 KB area (0x0200_2000 to 0x0200_3FFF) are not exchanged.	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application
Note 2. This register can only be programmed by FACI command. When programming, the set value should be 1.

6.2.5 OFS1, OFS1_SEC : Option Function Select Register 1 for Non-secure and Secure

Address: OFS1: 0x1300_A180 (Non-secure)
OFS1_SEC: 0x0300_A200 (Secure)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	INITECCEN	SWDBG	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	HOCOFRQ0[2:0]			HOCOEN	—	—	PVDLPSEL	—	PVDAS	VDSEL[2:0]		
Value after reset:	The value set by the user*1															

Bit	Symbol	Function	R/W
2:0	VDSEL[2:0]	Voltage Detection 0 Level Select 0 0 0: Select 2.85 V 0 0 1: Select 2.58 V 0 1 0: Select 2.15 V 0 1 1: Select 2.00 V 1 0 0: Select 1.90 V 1 0 1: Select 1.80 V 1 1 0: Select 1.70 V 1 1 1: Select 1.60 V*3	R
3	PVDAS	Voltage Detection 0 Circuit Start 0: Enable voltage monitor 0 reset after a reset 1: Disable voltage monitor 0 reset after a reset	R
4	—	The program value is read from this bit.*2	R
5	PVDLPSEL	Low Power Consumption Function of PVD0 Select at Deep Software Standby mode 0: Enable low power consumption function of PVD0 during DSTBY1 and DSTBY2. 1: Disable low power consumption function of PVD0 during DSTBY1 and DSTBY2.	R
7:6	—	The program value is read from these bits.*2	R
8	HOCOEN	HOCO Oscillation Enable *4 0: Enable HOCO oscillation after a reset 1: Disable HOCO oscillation after a reset	R
11:9	HOCOFRQ0[2:0]	HOCO Frequency Setting 0 0 0 0: 16 MHz 0 0 1: 18 MHz 0 1 0: 20 MHz 1 0 0: 32 MHz 1 1 1: 48 MHz Others: Reserved.	R
23:12	—	The program value is read from these bits.*2	R
24	SWDBG	Software Debug Control 0: Enable software debug control. The MCU behaves according to DBGAUTH0 and DBGAUTH1, and IWDT and WDT automatically stops when CPU is in the debug state. 1: Disable software debug control. The MCU ignores DBGAUTH0 and DBGAUTH1, and IWDT and WDT continue the operation even when CPU is in the debug state. However, when CDBGPWRUPREQ is 1 and AL is not AL0, IWDT and WDT automatically stops when CPU is in the debug state.	R
25	INITECCEN	Initial ECC Enable 0: Disable ECC function of TCM and CACHE. 1: Enable ECC function of TCM and CACHE.	R
31:26	—	The program value is read from these bits.*2	R

Note 1. The value of OFS1 and OFS1_SEC in a blank product is 0xFFFFFFFF. It is set to the value written by your application.
Note 2. This register can only be programmed by FACI command. When programming, the set value should be 1.
Note 3. Setting prohibited when VBATT function is enabled, OFS1(_SEC).PVDAS and PVDLPSEL bits are 0 and at least one of Deep Software Standby mode 1 and 2 is used.
Note 4. When FSBL is executed, the OFS1.HOCOEN register value must be 1. OFS1 address = 0x1200_A180 (Non-secure).

OFS1 register is for Non-secure developer, and OFS1_SEC register is for Secure developer.

VDSEL[2:0] bits (Voltage Detection 0 Level Select)

The VDSEL[2:0] bits select the voltage detection level of the voltage detection 0 circuit.

PVDAS bits (Voltage Detection 0 Circuit Start)

The PVDAS bit selects whether the voltage monitor 0 reset is enabled or disabled after a reset.

PVDLPSEL bits (Low Power Consumption Function of PVD0 Select at Deep Software Standby mode)

The PVDLPSEL selects whether the low power consumption function of PVD0 is enabled or disabled during DSTBY1 and DSTBY2 mode.

When disable this function, the response time is faster instead of increasing current consumption. When enable this function, the current consumption is smaller instead of response time delay.

See section 60, Electrical Characteristics

HOCOEN bit (HOCO Oscillation Enable)

The HOCOEN bit selects whether the HOCO oscillation is enabled or disabled after a reset. Setting this bit to 0 allows the HOCO oscillation to start before the CPU starts operation, which reduces the wait time for oscillation stabilization.

Note: When the HOCOEN bit is set to 0, the system clock source is not switched to HOCO. The system clock source is only switched to HOCO by setting the Clock Source Select bits (SCKSCR.CKSEL[2:0]). To use the HOCO clock, Set the OFS1(_SEC).HOCOFRQ0[2:0] bits*1 to an optimum value.

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits when OFS1(_SEC).HOCOEN=1.

HOCOFRQ0[2:0] bits (HOCO Frequency Setting 0)

The HOCOFRQ0[2:0] bits specify the HOCO frequency after a reset as 16 to 48 MHz.

SWDBG bit (Software Debug Control)

The SWDBG bit selects whether software debug control is enabled or disabled.

INITECCEN bit (Initial ECC Enable)

The INITECCEN bit selects whether ECC function of TCM and CACHE is enabled or disabled.

When the INITECCEN bit is changed from 1 to 0, be sure to perform a power-on reset after changing it.

6.2.6 OFS1_SEL : Option Function Select Register 1 for Security Attribution

Address: OFS1_SEL: 0x0300_A280

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	INITECCEN	SWDBG	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	HOCOFRQ0[2:0]			HOCOEN	—	—	PVDLPSEL	—	PVDAS	VDSEL[2:0]		
Value after reset:	The value set by the user*1															

Bit	Symbol	Function	R/W
2:0	VDSEL[2:0]	Security attributes of Voltage Detection 0 Level Select 0 0 0: Select OFS1_SEC.VDSEL[2:0] 1 1 1: Select OFS1.VDSEL[2:0] Others: Select Reserved.	R
3	PVDAS	Security attributes of Voltage Detection 0 Circuit Start 0: Select OFS1_SEC.PVDAS 1: Select OFS1.PVDAS	R
4	—	The program value is read from this bit.	R
5	PVDLPSEL	Security attributes of Low Power Consumption Function of PVD0 Select at Deep Software standby mode 0: Select OFS1_SEC.PVDLPSEL 1: Select OFS1.PVDLPSEL	R
7:6	—	The program value is read from these bits.	R
8	HOCOEN	Security attributes of HOCO Oscillation Enable*2 0: Select OFS1_SEC.HOCOEN 1: Select OFS1.HOCOEN	R
11:9	HOCOFRQ0[2:0]	Security attributes of HOCO Frequency Setting 0*3 0 0 0: Select OFS1_SEC.HOCOFRQ0[2:0] 1 1 1: Select OFS1.HOCOFRQ0[2:0] Others: Reserved	R
23:12	—	The program value is read from these bits.	R
24	SWDBG	Security attributes of Software Debug Control 0: Select OFS1_SEC.SWDBG 1: Select OFS1.SWDBG	R
25	INITECCEN	Security attributes of Initial ECC Enable 0: Select OFS1_SEC.INITECCEN 1: Select OFS1.INITECCEN	R
31:26	—	The program value is read from these bits.	R

Note 1. The value of OFS1_SEL in a blank product is 0x00000000. It is set to the value written by your application
Note 2. If FSBL is executed in reset, OFS1.HOCOEN is always selected regardless of this bit value.
Note 3. If FSBL is executed in reset, OFS1.HOCOFRQ0 is always selected regardless of this bit-field value.

The OFS1_SEL register selects the security attribute. Which security attribute is applied, OFS or OFS_SEC, is determined by the setting value of the corresponding bit in the OFS1_SEL register. For details, see section 6.3.3. Security Attribution of Code Flash Option-setting Memory.

6.2.7 BANKSEL, BANKSEL_SEC : Bank Select Register for Non-secure and Secure

Address: BANKSEL: 0x1300_A190 (Non-secure)
BANKSEL_SEC: 0x0300_A210 (Secure)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	BLCKSWP[6:0]						
Value after reset:	User setting*1															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	BANKSWP[2:0]		
Value after reset:	User setting*1															

Bit	Symbol	Function	R/W
2:0	BANKSWP[2:0]	Startup Bank Switch This setting is valid in dual mode. 0 0 0: Start address of Bank0 is code flash base address + 0x0020_0000 and Bank is code flash base address + 0x0000_0000 in dual mode 1 1 1: Start address of Bank0 is code flash base address + 0x0000_0000 and Bank is code flash base address + 0x0020_0000 in dual mode Others: Reserved	R
15:3	—	The program value is read from these bits. *1	R
22:16	BLCKSWP[6:0]	Block Swap Select When all bits are set to 1, the block swap is disabled. When at least one bit is set to 0, block swap is enabled and the corresponding blocks of code flash memory are swapped.	R
31:23	—	The program value is read from these bits.*1	R

Note 1. The value of BANKSEL and BANKSEL_SEC in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

BANKSEL register is for Non-secure developer, and BANKSEL_SEC register is for Secure developer.

BANKSWP[2:0] bits (Startup Bank Switch)

The BANKSWP[2:0] bits select the start address of the bank0 and bank1 of code flash memory in dual mode. For details of the startup bank selection, see section 52, Flash Memory.

BLCKSWP[6:0] bit (Block Swap Select)

The BLCKSWP[6:0] bits enable the block swap and select the valid blocks of the code flash memory in linear mode. Figure 6.2 shows the mapping of the flash memory in linear mode. Table 6.2 shows the specification of BLCKSWP bits for each product. Unused bits are reserved and should be set to 1. For details of block swap, see section 52, Flash Memory.

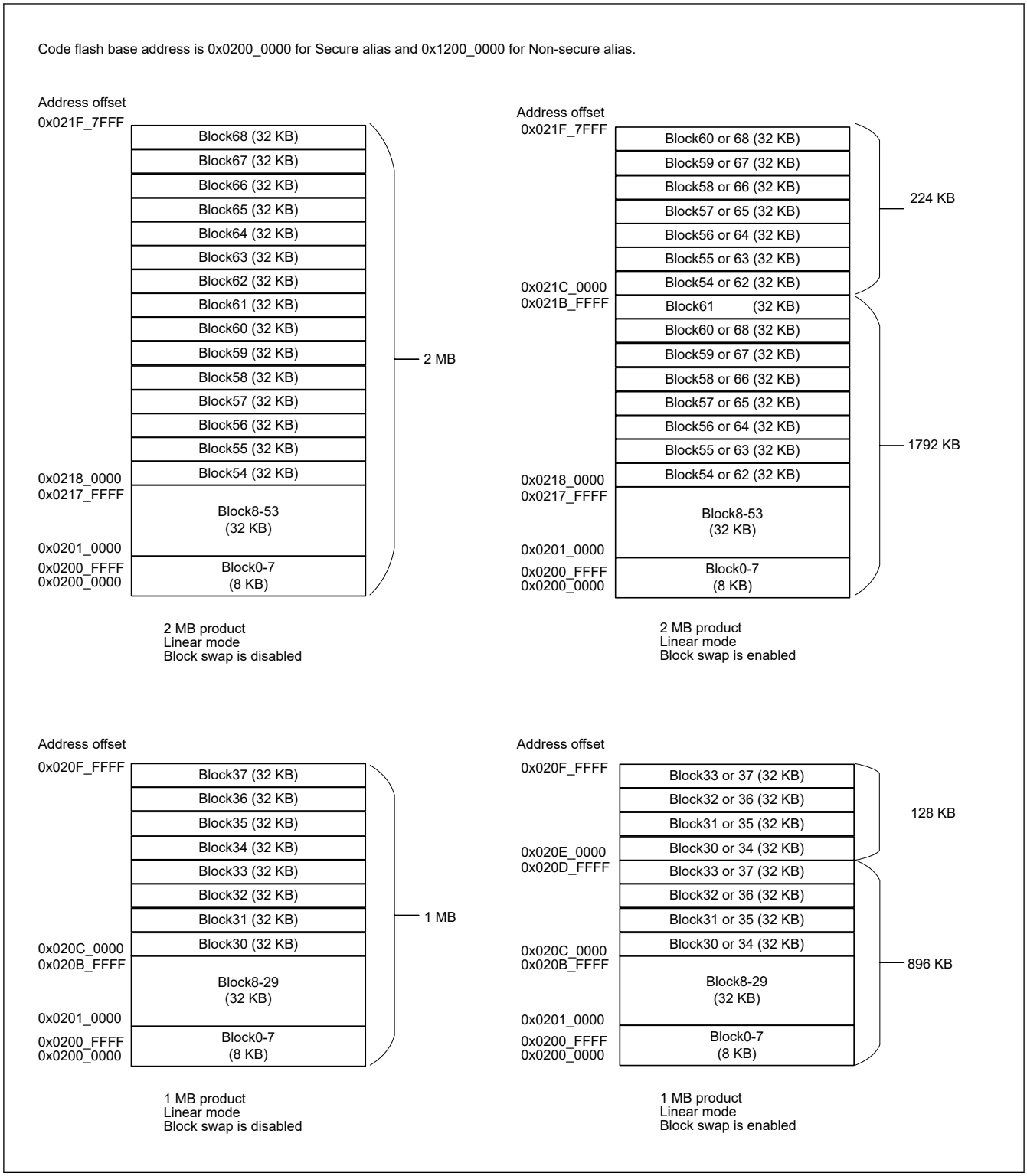


Figure 6.2 Mapping of the flash memory

Table 6.2 Specification of BLCKSWP bits for each product

	BLCKSWP bits	Select 0/1: Start address offset for each Block
2 MB product	BLCKSWP[0]	1 : Block54 start address is 0x0018_0000. Block62 start address is 0x001C_0000. 0 : Block54 start address is 0x001C_0000. Block62 start address is 0x0018_0000.
	BLCKSWP[1]	1 : Block55 start address is 0x0018_8000. Block63 start address is 0x001C_8000. 0 : Block55 start address is 0x001C_8000. Block63 start address is 0x0018_8000.
	BLCKSWP[2]	1 : Block56 start address is 0x0019_0000. Block64 start address is 0x001D_0000. 0 : Block56 start address is 0x001D_0000. Block64 start address is 0x0019_0000.
	BLCKSWP[3]	1 : Block57 start address is 0x0019_8000. Block65 start address is 0x001D_8000. 0 : Block57 start address is 0x001D_8000. Block65 start address is 0x0019_8000.
	BLCKSWP[4]	1 : Block58 start address is 0x001A_0000. Block66 start address is 0x001E_0000. 0 : Block58 start address is 0x001E_0000. Block66 start address is 0x001A_0000.
	BLCKSWP[5]	1 : Block59 start address is 0x001A_8000. Block67 start address is 0x001E_8000. 0 : Block59 start address is 0x001E_8000. Block67 start address is 0x001A_8000.
	BLCKSWP[6]	1 : Block60 start address is 0x001B_0000. Block68 start address is 0x001F_0000. 0 : Block60 start address is 0x001F_0000. Block68 start address is 0x001B_0000.
1 MB product	BLCKSWP[0]	1 : Block30 start address is 0x000C_0000. Block34 start address is 0x000E_0000. 0 : Block30 start address is 0x000E_0000. Block34 start address is 0x000C_0000.
	BLCKSWP[1]	1 : Block31 start address is 0x000C_8000. Block35 start address is 0x000E_8000. 0 : Block31 start address is 0x000E_8000. Block35 start address is 0x000C_8000.
	BLCKSWP[2]	1 : Block32 start address is 0x000D_0000. Block36 start address is 0x000F_0000. 0 : Block32 start address is 0x000F_0000. Block36 start address is 0x000D_0000.
	BLCKSWP[3]	1 : Block33 start address is 0x000D_8000. Block37 start address is 0x000F_8000. 0 : Block33 start address is 0x000F_8000. Block37 start address is 0x000D_8000.

6.2.8 BANKSEL_SEL : Bank Select Register for Security Attribution

Address: BANKSEL_SEL: 0x0300_A290

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	BLCKSWP[6:0]						
Value after reset:	User setting*1															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	BANKSWP[2:0]		
Value after reset:	User setting*1															

Bit	Symbol	Function	R/W
2:0	BANKSWP[2:0]	Security Attributes of Startup Bank Switch 0 0 0: Select BANKSEL_SEC.BANKSWP[2:0] 1 1 1: Select BANKSEL.BANKSWP[2:0] Others: Reserved	R
15:3	—	The program value is read from these bits. *1	R
22:16	BLCKSWP[6:0]	Security Attributes of Block Swap Select BANKSEL_SEL.BLCKSWP[6:0] bits can select Secure or Non-secure access for each bit. The bit positions of BLCKSWP[6:0] in the BANKSEL_SEL register and the bit positions of BLCKSWP[6:0] in the BANKSEL and BANKSEL_SEC registers correspond to indicate the same block. For example, when BANKSEL_SEL.BLCKSWP[0] = 0 and BANKSEL_SEL.BLCKSWP[6] = 1, then BANKSEL_SEC.BLCKSWP[0] and BANKSEL.BLCKSWP[6] are selected.	R
31:23	—	The program value is read from these bits..*1	R

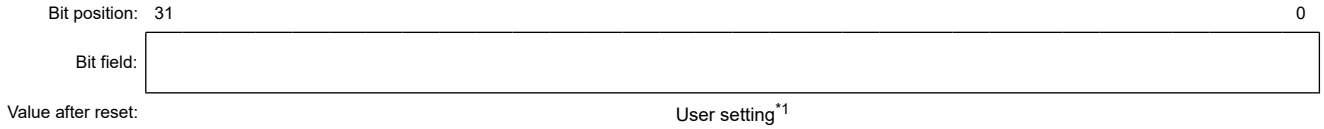
Note: This register can only be programmed by FACI command. When programming, the set value should be 0.
Note 1. The value of BANKSEL_SEL in a blank product is 0x00000000. It is set to the value written by your application.

The BANKSEL_SEL register selects the security attribute. Which security attribute is applied, BANKSEL or BANKSEL_SEC, is determined by the setting value of corresponding bit in the BANKSEL_SEL register. For details, see
section 6.3.3. Security Attribution of Code Flash Option-setting Memory
.

6.2.9 BPS, BPS_SEC, BPS_SEL : Block Protect Setting Register

address:

BPS: 0x1300_A1C0, 0x1300_A1C4, 0x1300_A1C8, 0x1300_A1CC (Non-secure)
BPS_SEC: 0x0300_A240, 0x0300_A244, 0x0300_A248, 0x0300_A24C (Secure)
BPS_SEL: 0x0300_A2C0, 0x0300_A2C4, 0x0300_A2C8, 0x0300_A2CC (Secure)



Note 1. The value of BPS and BPS_SEC in a blank product is 0xFFFFFFFF. The value of BPS_SEL in factory shipment product is 0x00000000. It is set to the value written by your application.

BPS_SEC and BPS_SEL register is for Secure developer, and BPS register is for Non-secure developer.

The BPS and BPS_SEC registers invalidate the programming and erasure to the code flash memory. When the bit of this register is set to 0, the programming and erasure to the corresponding block are invalid. Figure 6.3 shows the code flash block structure of each product. Figure 6.4 shows the relationship between the bit of register and the block number. Unused bits of BPS and BPS_SEC register are reserved and should be programed to 1.

The BPS_SEL register selects the security attribute. Whether the security attribute BPS or BPS_SEC, PBPS or PBPS_SEC is applied is determined by the setting value of the corresponding bit in the BPS_SEL register. For details, see section 6.3.3. Security Attribution of Code Flash Option-setting Memory.

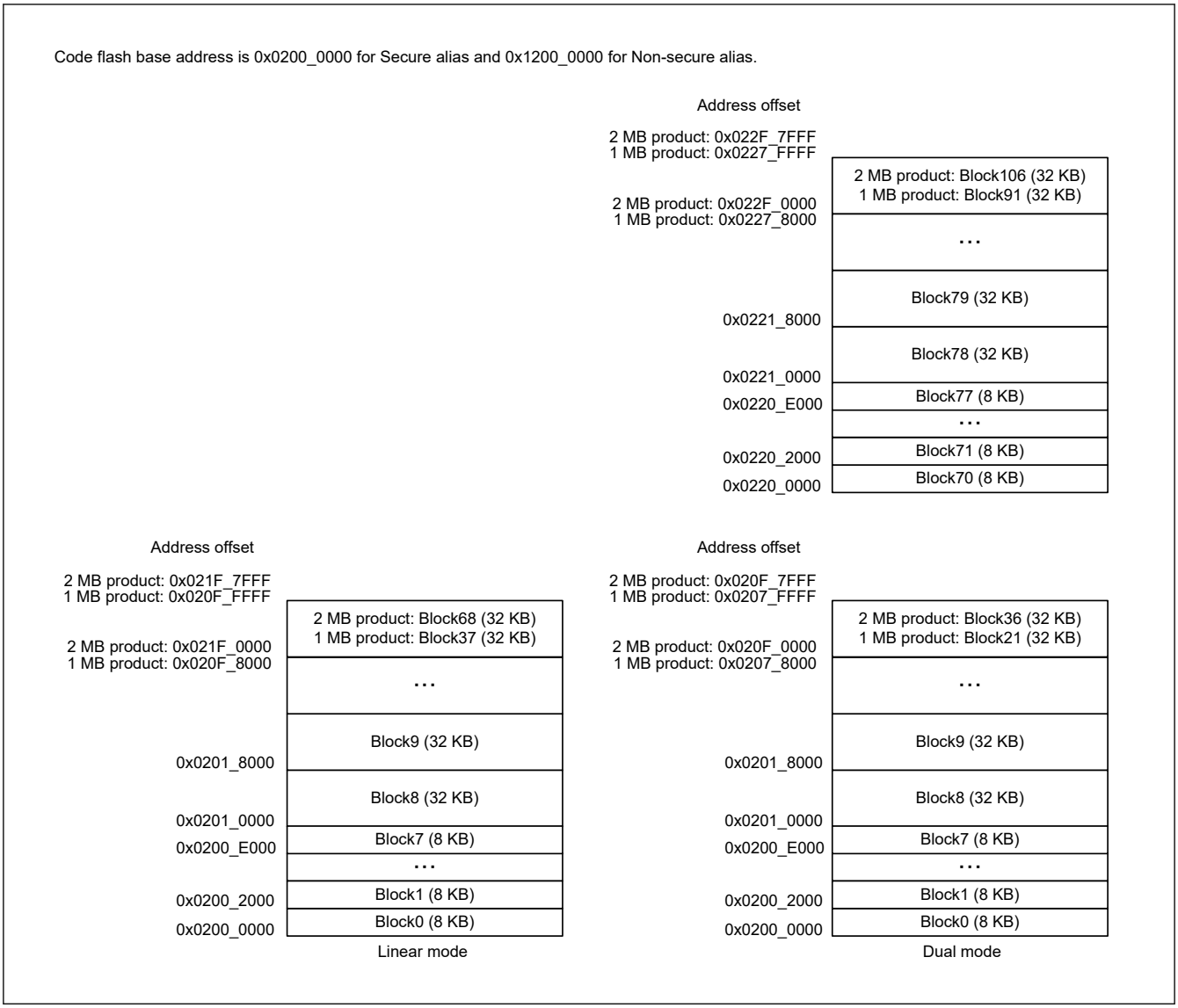


Figure 6.3 Code Flash block structure

Register	Address	+31	+30	+29	+28	+27	+26	+25	+24	+23	+22	+21	+20	+19	+18	+17	+16	+15	+14	+13	+12	+11	+10	+9	+8	+7	+6	+5	+4	+3	+2	+1	+0
BPS_SEL	0x0300_A2CC	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	106	105	104	103	102	101	100	99	98	97	96
	0x0300_A2C8	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
	0x0300_A2C4	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	0x0300_A2C0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	⋮																																
BPS_SEC	0x0300_A24C	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	106	105	104	103	102	101	100	99	98	97	96
	0x0300_A248	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
	0x0300_A244	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	0x0300_A240	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	⋮																																
BPS	0x1300_A1CC	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	106	105	104	103	102	101	100	99	98	97	96
	0x1300_A1C8	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
	0x1300_A1C4	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	0x1300_A1C0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figure 6.4 The relationship between the bit of register and the block number

6.2.10 PBPS, PBPS_SEC : Permanent Block Protect Setting Register

Address: PBPS: 0x1300_A1E0, 0x1300_A1E4, 0x1300_A1E8, 0x1300_A1EC (Non-secure)
PBPS_SEC: 0x0300_A260, 0x0300_A264, 0x0300_A268, 0x0300_A26C (Secure)

Bit position:	31	0
Bit field:		
Value after reset:	User setting*1	

Note 1. The value of PBPS and PBPS_SEC in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

PBPS_SEC register is for secure developer, and PBPS register is for non-secure developer. The applied setting value is determined by the setting value of the corresponding bit in BPS_SEL register. For details, see section 6.3.3. Security Attribution of Code Flash Option-setting Memory. The security attribution register is same BPS_SEL register between the block protection and permanent block protection.

The PBPS and PBPS_SEC registers invalidate writes to bits of BPS and BPS_SEC. The bit of this register can be programmed 0 when corresponding bit of BPS and BPS_SEC is programmed to 0. When the bit of this register is programmed to 0, writing the corresponding bit of BPS and BPS_SEC register is invalid. Once the bit of this register is set to 0, it is impossible to change the bit to 1. Table 6.3 shows the relationship between the bit of applied PBPS and bit of applied BPS.

When the applied setting value of this register selected by BPS_SEL register is 0, corresponding bit of BPS_SEL register cannot be changed.

The relationship between the bit of this register and the block number is same as BPS and BPS_SEC registers (Figure 6.4). Unused bits are reserved and should be set to 1.

Table 6.3 The relationship between the bit of PBPS, PBPS_SEC and bit of BPS, BPS_SEC

The bit of applied PBPS and PBPS_SEC	The bit of applied BPS and BPS_SEC	Content
1	1	Programming and erasure to the corresponding block is valid.
1	0	Programming and erasure to the corresponding block is invalid. This protection can be canceled by FBPROT0 or FBPROT1 registers.
0	1	Can not set this condition
0	0	Programming and erasure to the corresponding block is invalid permanently

Note: When BPS_SEL = 0, BPS and PBPS correspond. When BPS_SEL = 1, BPS_SEC and PBPS_SEC correspond.

6.2.11 FSBLCTRL0 : FSBL Control Register 0

Address: 0x2703_0080

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	User setting*1															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	FSBLCLK[2:0]			FSBLSKIPDS[2:0]			FSBLSKIPSW[2:0]			FSBLEN[2:0]		
Value after reset:	User setting*1															

Bit	Symbol	Function	R/W
2:0	FSBLEN[2:0]	FSBL Enable 0 0 0: Enable FSBL 1 1 1: Disable FSBL Others: Reserved	R

Bit	Symbol	Function	R/W
5:3	FSBLSKIPSW[2:0]	FSBL Skip Enable for Software Reset 0 0 0: Enable skip 1 1 1: Disable skip Others: Reserved	R
8:6	FSBLSKIPDS[2:0]	FSBL Skip Enable for Deep Software Standby Reset 0 0 0: Enable skip 1 1 1: Disable skip Others: Reserved	R
11:9	FSBLCLK[2:0]	Clock Frequency Selection during FSBL Execution 0 0 0: CPUCLK = 120 MHz ICLK = 120 MHz 1 1 1: CPUCLK = 240 MHz ICLK = 240 MHz Others: Reserved	R
31:12	—	The program value is read from these bits.*2	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.
Note 2. This register can only be programmed by FACI command. When programming, the set value should be 1.

6.2.12 FSBLCTRL1 : FSBL Control Register 1

Address: 0x2703_0084

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	User setting*1															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FSBLEXMD[1:0]	
Value after reset:	User setting*1															

Bit	Symbol	Function	R/W
1:0	FSBLEXMD[1:0]	FSBL Execution Mode 0 0: CRC boot and not report measurement 0 1: CRC boot and report measurement 1 0: Secure boot and not report measurement 1 1: Secure boot and report measurement	R
31:2	—	The program value is read from these bits.*2	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.
Note 2. This register can only be programmed by FACI command. When programming, the set value should be 1.

6.2.13 FSBLCTRL2 : FSBL Control Register 2

Address: 0x2703_0088

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	User setting*1															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	PORTGN[4:0]					PORTPN[3:0]			
Value after reset:	User setting*1															

Bit	Symbol	Function	R/W
3:0	PORTPN[3:0]	FSBL Error Notification Port Pin Number*2 0 0 0 0: PORTn00 0 0 0 1: PORTn01 ⋮ 1 1 1 1: PORTn15 (n = 0 to 9, A, B)	R
8:4	PORTGN[4:0]	FSBL Error Notification Port Group Name*2 0 0 0 0 0: PORT0m 0 0 0 0 1: PORT1m ⋮ 0 1 0 1 1: PORTBm Others: Reserved (m = 0 to 15)	R
31:9	—	The program value is read from these bits.*3	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.
Note 2. When non-existent pin is selected, FSBL error notification is not output.
Note 3. This register can only be programmed by FACI command. When programming, the set value should be 1.

FSBLCTRL2 register selects the port to output when an Error occurs during FSBL execution. The PORTGN bits select a group of ports, and the PORTPN bits select the pins of that group. It thereby determines the unique port.

6.2.14 SACC0 : Start Address of Code Certificate 0

Address: 0x2703_008C

Bit position: 31 0

Bit field:

Value after reset: User setting*1

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

6.2.15 SACC1 : Start Address of Code Certificate 1

Address: 0x2703_0090

Bit position: 31 0

Bit field:

Value after reset: User setting*1

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

6.2.16 SAMR : Start Address of Measurement Report

Address: 0x2703_0094

Bit position: 31 0

Bit field:

Value after reset: User setting*1

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

The SAMR register specifies the starting address of measurement report. The measurement report is stored in the SRAM shown in Figure 43.14 in section 43, Security Features .

Specify SRAM0, or SRAM1 or Standby RAM. When SRAM0 is used for measurement report, do not specify the addresses from 0x22007780 to 0x22007BFF. When reading the measurement report in SRAM0, disable the ECC functionality by setting SRAMCR0.ECCMOD[1:0] = b00.

6.2.17 HOEMRTPK :Hash of OEM_ROOT_PK

Base address: FDFS = 0x2703_0000

Offset address: 0x360

Bit position:	31 to 0
Bit field:	
Value after reset:	User setting*1

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application

This register can be programmed only by boot firmware.

This register does not contain the Hash value, but a processed Hash value.

6.2.18 CFGD0LOCK : Configuration Data 0 Lock Bit

Base address: 0x2703_0380

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CD0LK31	CD0LK30	CD0LK29	CD0LK28	CD0LK27	CD0LK26	CD0LK25	CD0LK24	CD0LK23	CD0LK22	CD0LK21	CD0LK20	CD0LK19	CD0LK18	CD0LK17	CD0LK16
Value after reset:	User setting*1															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CD0LK15	CD0LK14	CD0LK13	CD0LK12	CD0LK11	CD0LK10	CD0LK9	CD0LK8	CD0LK7	CD0LK6	CD0LK5	CD0LK4	CD0LK3	CD0LK2	CD0LK1	CD0LK0
Value after reset:	User setting*1															

Base address: 0x2703_0384

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CD0LK63	CD0LK62	CD0LK61	CD0LK60	CD0LK59	CD0LK58	CD0LK57	CD0LK56	CD0LK55	CD0LK54	CD0LK53	CD0LK52	CD0LK51	CD0LK50	CD0LK49	CD0LK48
Value after reset:	User setting*1															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CD0LK47	CD0LK46	CD0LK45	CD0LK44	CD0LK43	CD0LK42	CD0LK41	CD0LK40	CD0LK39	CD0LK38	CD0LK37	CD0LK36	CD0LK35	CD0LK34	CD0LK33	CD0LK32
Value after reset:	User setting*1															

Bit	Symbol	Function	R/W
31:0	CD0LK31 to CD0LK0	Configuration Data 0 Lock Bit 0: LK_CD_A0 protection is enabled 1: LK_CD_A0 protection is disabled	R
31:0	CD0LK63 to CD0LK32	Configuration Data 0 Lock Bit 0: LK_CD_A0 protection is enabled 1: LK_CD_A0 protection is disabled	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application

CD0LKn bits (Configuration Data 0 Lock Bit)

These bits specify write protection for Lockable Configuration Data Area 0 (LK_CD_A0) in configuration area of data flash. CD0_LK(n) protects 4 bytes data in LK_CD_A0.

CD0_LK bits can be set to "0" by the "Configuration set" command and cannot be changed from "0" to "1".

6.2.19 CFGD1LOCK : Configuration Data 1 Lock Bit

Base address: 0x2703_0388

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CD1LK31	CD1LK30	CD1LK29	CD1LK28	CD1LK27	CD1LK26	CD1LK25	CD1LK24	CD1LK23	CD1LK22	CD1LK21	CD1LK20	CD1LK19	CD1LK18	CD1LK17	CD1LK16
Value after reset:	User setting*1															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CD1LK15	CD1LK14	CD1LK13	CD1LK12	CD1LK11	CD1LK10	CD1LK9	CD1LK8	CD1LK7	CD1LK6	CD1LK5	CD1LK4	CD1LK3	CD1LK2	CD1LK1	CD1LK0
Value after reset:	User setting*1															

Base address: 0x2703_038C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CD1LK63	CD1LK62	CD1LK61	CD1LK60	CD1LK59	CD1LK58	CD1LK57	CD1LK56	CD1LK55	CD1LK54	CD1LK53	CD1LK52	CD1LK51	CD1LK50	CD1LK49	CD1LK48
Value after reset:	User setting*1															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CD1LK47	CD1LK46	CD1LK45	CD1LK44	CD1LK43	CD1LK42	CD1LK41	CD1LK40	CD1LK39	CD1LK38	CD1LK37	CD1LK36	CD1LK35	CD1LK34	CD1LK33	CD1LK32
Value after reset:	User setting*1															

Bit	Symbol	Function	R/W
31:0	CD1LK31 to CD1LK0	Configuration Data 1 Lock Bit 0: LK_CD_A1 protection is enabled 1: LK_CD_A1 protection is disabled	R
31:0	CD1LK63 to CD1LK32	Configuration Data 1 Lock Bit 0: LK_CD_A1 protection is enabled 1: LK_CD_A1 protection is disabled	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application

CD1LKn bits (Configuration Data 1 Lock Bit)

These bits specify write protection for Lockable Configuration Data Area 1 (LK_CD_A1) in configuration area of data flash. CD1_LK(n) protects 4 bytes data in LK_CD_A1.

CD1_LK bits can be set to "0" by the "Configuration set" command and cannot be changed from "0" to "1"

6.2.20 CFGD2LOCK : Configuration Data 2 Lock Bit

Base address: 0x2703_0390

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CD2LK15	CD2LK14	CD2LK13	CD2LK12	CD2LK11	CD2LK10	CD2LK9	CD2LK8	CD2LK7	CD2LK6	CD2LK5	CD2LK4	CD2LK3	CD2LK2	CD2LK1	CD2LK0
Value after reset:	User setting*1															

Bit	Symbol	Function	R/W
15:0	CD2LK15 to CD2LK0	Configuration Data 2 Lock Bit 0: LK_CD_A2 protection is enabled 1: LK_CD_A2 protection is disabled	R

Note 1. The value in a blank product is 0xFFFF_FFFF. It is set to the value written by your application

CD2LKn bits (Configuration Data 2 Lock Bit)

These bits specify write protection for Lockable Configuration Data Area 2 (LK_CD_A2) in configuration area of data flash. CD2_LK(n) protects 16 bytes data in LK_CD_A2. CD2_LK bits can be set to "0" by the "Configuration set" command and cannot be changed from "0" to "1".

Settings for Configuration Data Lock Bit (CDn_LK) (n = 0 to 2) are shown in Table 6.4

Table 6.4 The correspondence between the lock bit and lockable are address

Area	Bit	Address
Lockable area 0	CD0_LK0	0x2703_0080 to 0x2703_0083
	CD0_LK1	0x2703_ 0084 to 0x2703_0087
	…	…
	CD0_LKn (n = 0 to 63)	0x2703_0080 + 0x04 × n to 0x2703_0083 + 0x04 × n
	…	…
	CD0_LK63	0x2703_017C to 0x2703_017F
Lockable area 1	CD1_LK0	0x2703_0180 to 0x2703_0183
	CD1_LK1	0x2703_0184 to 0x2703_0187
	…	…
	CD1_LKn (n = 0 to 63)	0x2703_0180 + 0x04 × n to 0x2703_0183 + 0x04 × n
	…	…
	CD1_LK63	0x2703_00x27C to 0x2703_027F
Lockable area 2	CD2_LK0	0x2703_0280 to 0x2703_028F
	CD2_LK1	0x2703_0290 to 0x2703_029F
	…	…
	CD2_LKn (n = 0 to 15)	0x2703_0280 + 0x10 × n to 0x2703_028F + 0x10 × n
	…	…
	CD2_LK15	0x2703_0370 to 0x2703_037F

6.2.21 ARCLS : Anti-Rollback Counter Lock Setting

Base address: 0x2703_03C0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	ARCBL_LK	ARCNS_LK[3:0]				ARCS_LK
Value after reset:	User setting*1															

Bit	Symbol	Function	R/W
0	ARCS_LK	ARC_SEC Lock 0: ARC_SEC protection is enabled 1: ARC_SEC protection is disabled	R/W
4:1	ARCNS_LK[3:0]	ARC_NSEC Lock In case of 64 bits x 4 counter setting x x x 0: ARC_NSEC[63:0] protection is enabled x x 0 x: ARC_NSEC[127:64] protection is enabled x 0 x x: ARC_NSEC[191:128] protection is enabled 0 x x x : ARC_NSEC[255:192] protection is enabled 1 1 1 1: ARC_NSEC[255:0] protection is disabled Others Reserved. In case of 256 bits counter setting 0 0 0 0: ARC_NSEC[255:0] protection is enabled 1 1 1 1: ARC_NSEC[255:0] protection is disabled Others Reserved.	R/W

Bit	Symbol	Function	R/W
5	ARCBL_LK	ARC_OEMBL Lock 0: ARC_OEMBL protection is enabled 1: ARC_OEMBL protection is disabled	R/W
15:6	—	The program value is read from these bits*2	R/W

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application
Note 2. This register can only be programmed by FACI command. When programming, the set value should be 1.

Anti-rollback counter Lock bits setting are located in the data flash configuration area. Anti-rollback counter Lock bits setting can be set to “0” by the "Program" command and cannot be changed from "0" to "1". When the lock bit is 0, the anti-rollback counter corresponding to each bit cannot be updated by the "Increment Counter" command.

6.2.22 ARCCS : Anti-Rollback Counter Configuration Setting for Non-secure Application

Base address: 0x2703_03C2

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CNF_ARCNS [1:0]	
Value after reset:	User setting*1															

Bit	Symbol	Function	R/W
1:0	CNF_ARCNS[1:0]	Configuration setting for ARC_NSEC 0 0: ARC_NSEC configuration is 4 × 64 bits 0 1: ARC_NSEC configuration is 1 × 256 bits Others ARC_NSEC configuration is disable	R/W
15:2	—	The program value is read from these bits*2	R/W

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application
Note 2. This register can only be programmed by FACI command. When programming, the set value should be 1.

ARC_NSEC configuration setting is located in the data flash configuration area. These bits specify two types of ARC_NSEC configuration: 4 × 64 bits or 1 × 256 bits. “Increment Counter” or “Read Counter” command cannot be issued to ARC_NSEC when counter configuration setting is disabled.

6.2.23 ARC_SECn : Anti-Rollback Counter for Secure Application n (n = 0, 1)

Base address: 0x2703_0000

Offset address: 0x850 + 0x004 × n (n = 0, 1)

Bit position:	31	0
Bit field:	ARC_SEC[32 × n + 31 : 32 × n]	
Value after reset:	User setting*1	

Bit	Symbol	Function	R/W
31:0	ARC_SEC[32 × n + 31 : 32 × n]	Anti-Rollback Counter for Secure Application The counter value is obtained by arranging the read values from the upper register (n = 1) to the lower register (n = 0). See section 52.12.5. Anti-Rollback Counter for detail	R/W

Note 1. The value in a blank product is 0x0. It is set to the value written by your application

6.2.24 ARC_NSECn : Anti-Rollback Counter for Non-secure Application n (n = 0 to 7)

Base address: 0x2703_0000

Offset address: 0x858 + 0x004 × n (n = 0 to 7)

Bit position:	31	0
Bit field:	ARC_NSECn[32 × n + 31 : 32 × n]	
Value after reset:	User setting*1	

Bit	Symbol	Function	R/W
31:0	ARC_NSECn[32 × n + 31 : 32 × n]	Anti-Rollback Counter for Non-secure Application The counter value is obtained by arranging the read values from the upper register (n = 7) to the lower register (n = 0). See section 52.12.5. Anti-Rollback Counter for detail	R/W

Note 1. The value in a blank product is 0x0. It is set to the value written by your application

6.2.25 ARC_OEMBLn : Anti-Rollback Counter for OEMBL (n = 0, 1)

Base address: 0x2703_0000

Offset address: 0x878 + 0x004 × n (n = 0, 1)

Bit position:	31	0
Bit field:	ARC_OEMBL[32 × n + 31 : 32 × n]	
Value after reset:	User setting*1	

Bit	Symbol	Function	R/W
31:0	ARC_OEMBL [32*n+31 : 32*n]	Anti-Rollback Counter for OEM_BL Application The counter value is obtained by arranging the read values from the upper register (n = 1) to the lower register (n = 0). See section 52.12.5. Anti-Rollback Counter for detail	R/W

Note 1. The value in a blank product is 0x0. It is set to the value written by your application

6.3 Setting Option-Setting Memory

6.3.1 Allocation of Data in Option-Setting Memory

Programming data is allocated to the addresses in the option-setting memory shown in Figure 6.1. The allocated data is used by FSBL or tools such as a flash programming software or an on-chip debugger.

Note: Programming formats vary depending on the compiler. See the compiler manual for details.

6.3.2 Setting Data for Programming Option-Setting Memory

Allocating data according to the procedure described in section 6.3.1. Allocation of Data in Option-Setting Memory, alone does not actually write the data to the option-setting memory. You must also follow one of the actions described in this section.

(1) Changing the option-setting memory by self-programming

Use the configuration setting command to write data to the option-setting memory in the configuration setting area.

The code flash option-setting memory does not support background operations (BGO). When write the option-setting memory, jump to SRAM after copying writing software to SRAM.

For details of the configuration setting command, see section 52, Flash Memory.

(2) Debugging through an OCD or programming by a flash writer

This procedure depends on the tool in use, see the tool manual for details.

The MCU provides two setting procedures:

- Read the data allocated as described in section 6.3.1. Allocation of Data in Option-Setting Memory, from an object file or Motorola S-format file generated by the compiler, and write the data to the MCU
- Use the GUI interface of the tool to program the same data as allocated in section 6.3.1. Allocation of Data in Option-Setting Memory.

6.3.3 Security Attribution of Code Flash Option-setting Memory

Some functionality has three registers for Non-secure (FUNC NAME), and Secure (FUNC NAME_SEC), and security attribution (FUNC NAME_SEL). Only Secure developer can set the registers for Secure and security attribution. As shown in Figure 6.5, when the bit of security attribution register is set to 0, the corresponding bit of Secure register is applied. When the bit of security attribution register is set to 1, the corresponding bit of Non-secure register is applied.

For example, if the Secure developer wants to configure PVD of OFS1 as Secure, HOCO of OFS1 as Non-secure, the Secure developer needs to set OFS1_SEL as follows.

OFS1_SEL = 0xFFFF_FFF0

By this setting, PVDAS and VDSEL[2:0] values of OFS1_SEC and HOCOFRQ0[2:0] and HOCOEN values of OFS1 are applied to MCU. The reserved bits of the security attribution register (FUNC NAME_SEL) should be set to 0.

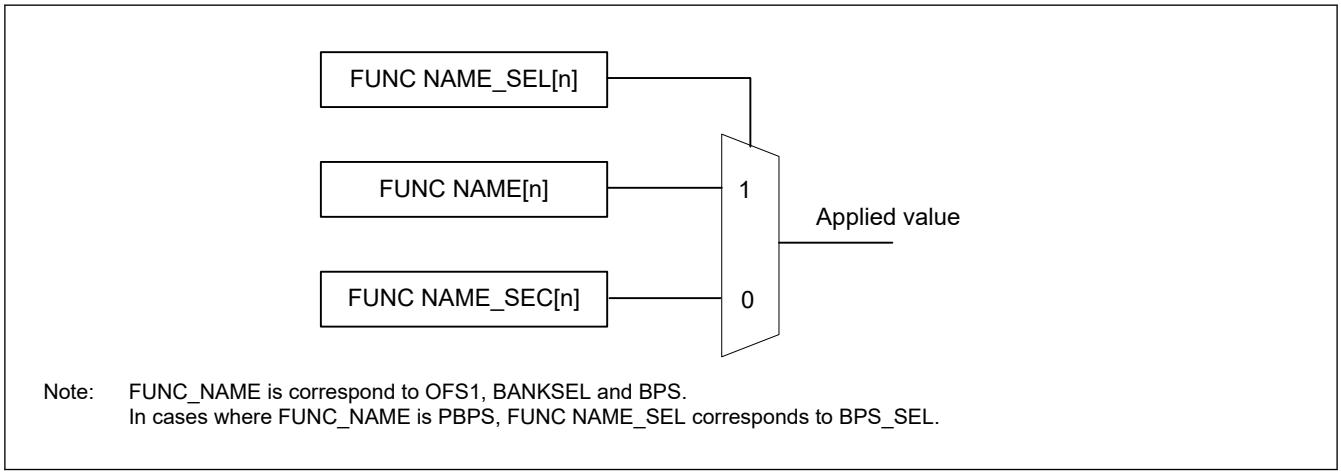


Figure 6.5 Selection of applied value

6.3.4 Timing of the Setting Value

For SAS, BPS, BPS_SEC, PBPS, PBPS_SEC and BPS_SEL registers, the setting value of the related startup area and block protection is applied immediately after programming. For other registers, the setting value is applied after the MCU is reset.

6.4 Usage Notes

6.4.1 Data for Programming Reserved Areas and Reserved Bits in the Option-Setting Memory

- Security attribution region (0x0300 A280 to 0x0300 A2FF)
  When reserved areas and reserved bits in the option-setting memory are within the scope of programming, write 0 to all bits in reserved areas and all reserved bits. Operation is not guaranteed if 1 is written to these bits.
- Other region
  When reserved areas and reserved bits in the option-setting memory are within the scope of programming, write 1 to all bits of reserved areas and all reserved bits. If 0 is written to these bits, normal operation cannot be guaranteed.

6.4.2 FSBL Execution After System Reset

When FSBLCTRL0.FSBLEN bit is set to 000b, FSBL is always executed after system reset is released. However, setting FSBLCTRL0.FSBLSKIPSW or FSBLCTRL0.FSBLSKIPDS bit adds a condition to Skip FSBL. See Table 6.5.

Table 6.5 FSBL execution conditions

FSBLEN	FSBLSKIPSW	FSBLSKIPDS	FSBL execution after system reset
111b	*	*	Not executed
000b	111b	111b	Executed
000b	111b	000b	Deep Software Standby reset : Not executed Other reset than above : Executed
000b	000b	111b	Software reset : Not executed Other reset than above : Executed
000b	000b	000b	Software reset : Not executed Deep Software Standby reset : Not executed Other reset than above : Executed

When using the FSBL skip function, the reset status registers (RSTSR0 and RSTSR1) should be cleared before the system reset is released.

Note: the FSBL will always run if at least one of the reset flags that allows the FSBL to run is set.

6.4.3 FSBL Skip Function

To skip FSBL execution after a software reset release, RSTSR0 and RSTSR1 should be cleared before executing software reset. If any flags other than RSTSR1.SWRF is set, FSBL may be executed.

To skip FSBL execution after a Deep Software Standby Reset release, RSTSR0 and RSTSR1 should be cleared before the transition to Deep Software Standby mode. If any flags other than RSTSR0.DPSRSTF is set, FSBL may be executed.

6.4.4 Notes on FSBL Execution

If FSBL execution is selected, the following conditions are required:

1. OFS1.HOCOEN must be used as oscillation disabled after reset. Therefore, the OFS1.HOCOEN register value must be 1.
2. OFS1_SEL.HOCOEN and OFS1_SEL.HOCOFRQ0 always select OFS1.HOCOEN and OFS1.HOCOFRQ0 respectively. Therefore, use HOCOCR and HOCOCR2, or OFS1.HOCOFREQ0 register to operate, stop and set frequency for HOCO.
3. When SRAM0 is used for measurement report, do not specify the addresses from 0x22007780 to 0x22007BFF. When reading the measurement report in SRAM0, disable the ECC functionality by setting SRAMCR0.ECCMOD[1:0] = b00.
4. MSTPC0, MSTPC1, MSTPC4, MSTPC6, MSTPC7, MSTPC8, MSTPC11, MSTPC12, MSTPC13, MSTPC14, MSTPC16, MSTPC26, MSTPC27, MSTPC31 value after reset is 0. The corresponding module is not stopped. MSTPC10 may be 0 depending on the parts number. If the corresponding module is not in use, change the respective module stop bit to 1 to reduce current consumption.
5. The initial value of CRCDOR register is not 0x00000000. Rewrite the CRCDOR/CRCDOR_HA/CRCDOR_BY register to perform the CRC calculations using a value other than the value after reset.

7. Programmable Voltage Detection (PVD)

7.1 Overview

The Programable Voltage Detection (PVD) module monitors the voltage level input to the VCC pin and the detection level can be selected using a software program. The PVD module consists of several separate voltage level detectors which measure the voltage level input to the VCC pin. PVD registers allow your application to configure detection of VCC change at various voltage thresholds.

Each voltage level detector has a voltage monitor associated with it. Voltage monitor registers are used to configure the PVD to trigger an interrupt, event link output, or reset when the thresholds are crossed.

Table 7.1 lists the PVD specifications. Figure 7.1 shows a block diagram of the voltage monitor 0 reset generation circuit. Figure 7.2 shows a block diagram of the voltage monitor m interrupt and reset circuit (m = 1, 2).

Table 7.1 PVD specifications

Item		Voltage monitor 0	Voltage monitor m (m = 1, 2)
Voltage Monitoring	Monitored pin	VCC	VCC
	Detected Event	Voltage falls past V_{det0}	Voltage rises or falls past V_{detm}
	Detection Voltage	Selectable from among 8 different levels by using OFS1(_SEC).VDSEL [2:0] bits as V_{det0}	Selectable from among 13 different levels by using PVDmCMPCR.PVDLVL[4:0] bits as V_{detm}
	Monitor Flag	None	PVDmSR.MON flag: Monitors whether voltage is higher or lower than V_{detm}
			PVDmSR.DET flag: V_{detm} passage detection
Process on voltage detection	Reset	Voltage monitor 0 reset	Voltage monitor m reset
		Reset when V_{det0} > VCC CPU restart after specified time with VCC > V_{det0}	Reset condition selectable Reset when V_{detm} > VCC or V_{detm} < VCC CPU restart timing selectable: after specified time with VCC > V_{detm} or V_{detm} > VCC
	Interrupt	No interrupt	Voltage monitor m interrupt
			Non-maskable interrupt or maskable interrupt selectable
			Interrupt request issued in response to either or both V_{detm} > VCC and VCC > V_{detm}
Digital filter	Enable/ Disable Switching	Digital filter function not available	Available
	Sampling Time	—	1/n LOCO frequency × 2 (n: 2, 4, 8, 16)
Event linking		None	Available Output of event signals on detection of V_{detm} crossings

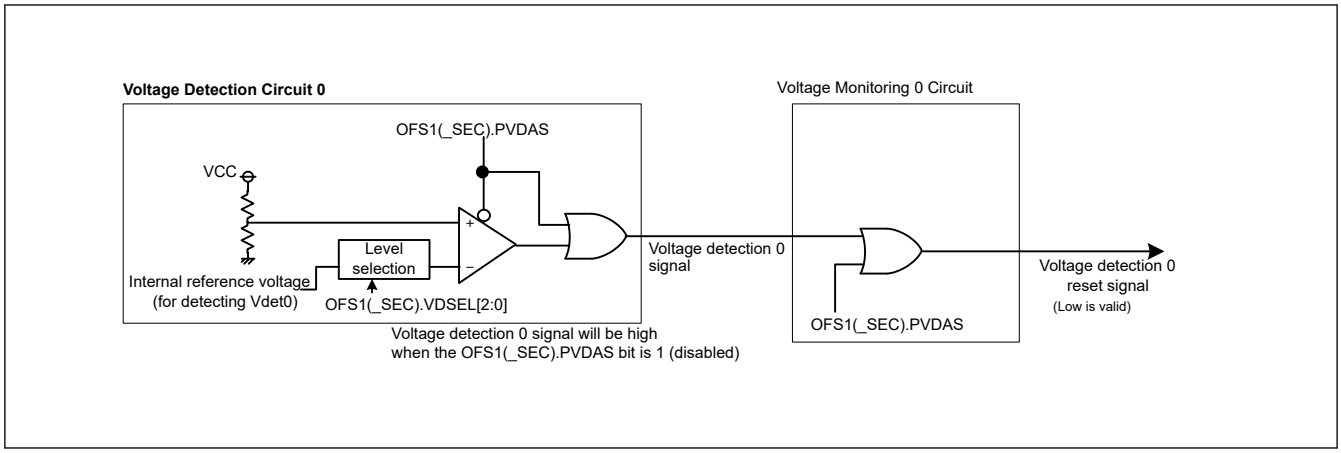


Figure 7.1 Block diagram of voltage monitor 0 reset circuit

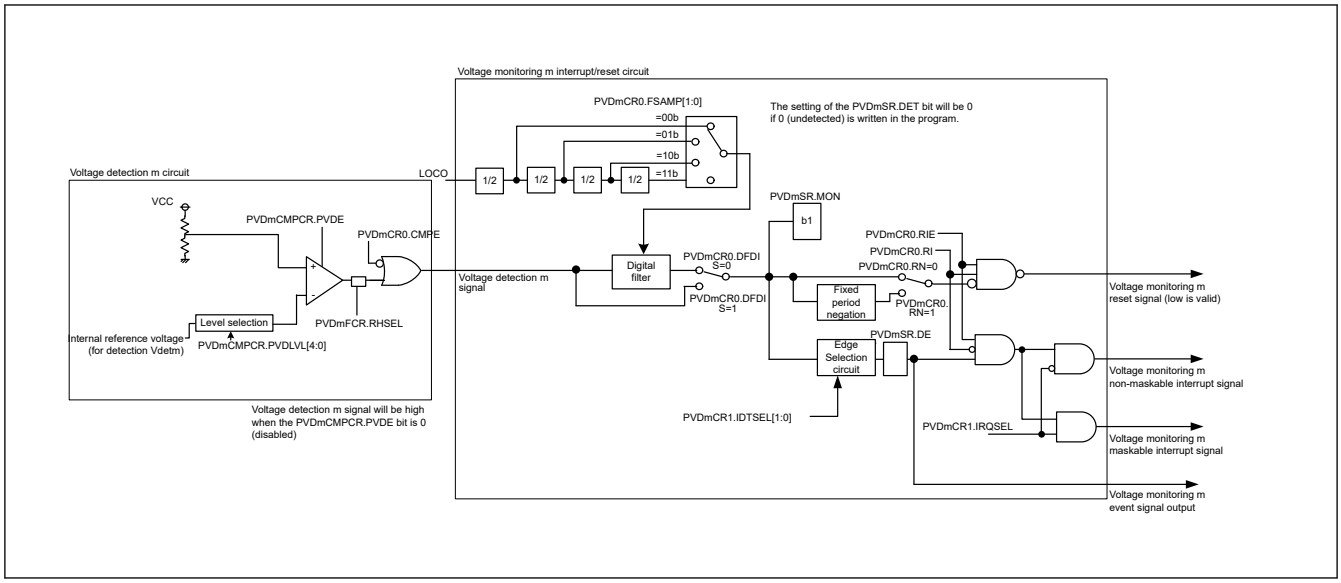


Figure 7.2 Block diagram of voltage monitor m interrupt/reset circuit (m = 1, 2)

7.2 Register Descriptions

7.2.1 PVDSAR : Programable Voltage Detection Security Attribution Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x3CC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	NONSEC1	NONSEC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NONSEC0	Non-secure Attribute bit 0 Target register: registers for PVD1 0: Secure 1: Non-secure	R/W
1	NONSEC1	Non-secure Attribute bit 1 Target register: registers for PVD2 0: Secure 1: Non-secure	R/W
31:2	—	These bits are read as 1. The write value must be 1 when it is possible to write.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: This register is write-protected by PRCR register.

The PVDSAR register controls the secure attribute of PVD registers.

NONSEC0 bit (Non-secure Attribute bit 0)

This bit controls the security attribute of PVD1CMPCR, PVD1CR0, PVD1CR1, PVD1SR and PVD1FCR.

NONSEC1 bit (Non-secure Attribute bit 1)

This bit controls the security attribute of PVD2CMPCR, PVD2CR0, PVD2CR1, PVD2SR and PVD2FCR.

7.2.2 PVDmCMPCR : Voltage Monitor m Comparator Control Register (m = 1, 2)

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA58 + 0x4 × (m - 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PVDE	—	—	PVDLVL[4:0]				
Value after reset:	0	0	0	0	1	1	1	1

Bit	Symbol	Function	R/W
4:0	PVDLVL[4:0]	Detection Voltage m Level Select Settings other than the following are prohibited. These are standard voltage level during the fall of voltage. For details on standard voltage level during the rise of voltage, see section 60, Electrical Characteristics. 0x03: 3.86 V (Vdetm_3) 0x04: 3.14 V (Vdetm_4) 0x05: 3.10 V (Vdetm_5) 0x06: 3.08 V (Vdetm_6) 0x07: 2.85 V (Vdetm_7) 0x08: 2.83 V (Vdetm_8) 0x09: 2.80 V (Vdetm_9) 0x0A: 2.62V (Vdetm_10) 0x0B: 2.33V (Vdetm_11) 0x0C: 1.90V (Vdetm_12) 0x0D: 1.86V (Vdetm_13) 0x0E: 1.74V (Vdetm_14) 0x0F: 1.71V (Vdetm_15) Others: Setting prohibited	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	PVDE	Voltage Detection m Enable 0: Voltage detection m circuit disabled 1: Voltage detection m circuit enabled	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

PVDLVL[4:0] bits (Detection Voltage m Level Select)

The PVDmCMPCR.PVDLVL can be changed only if the PVDmCMPCR.PVDE bits are both 0. All the voltage detection circuits should not be set at the same voltage detection level.

Do not change PVDmCMPCR.PVDLVL and PVDmCMPCR.PVDE at the same time.

The PVDmCMPCR.PVDE can be changed only when the PVDmCR0.CMPE is 0.

PVDE bit (Voltage Detection m Enable)

When using voltage detection m interrupt/reset or the PVDmSR.MON bit, set the PVDmCMPCR.PVDE bit to 1. The voltage detection m circuit starts once td(E-A) passes after the PVDmCMPCR.PVDE bit value is changed from 0 to 1. When using the voltage detection m circuit in Deep Software Standby mode, do not transit to Deep Software Standby mode 3.

7.2.3 PVDmCR0 : Voltage Monitor m Circuit Control Register 0 (m = 1, 2)

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA70 + 0x4 × (m - 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RN	RI	FSAMP[1:0]		—	CMPE	DFDIS	RIE
Value after reset:	1	0	0	0	x	0	1	0

Bit	Symbol	Function	R/W
0	RIE	Voltage Monitor m Interrupt/Reset Enable 0: Disable 1: Enable	R/W
1	DFDIS	Voltage monitor m Digital Filter Disabled Mode Select 0: Enable the digital filter 1: Disable the digital filter	R/W
2	CMPE	Voltage Monitor m Circuit Comparison Result Output Enable 0: Voltage monitor m circuit comparison result output disabled 1: Voltage monitor m circuit comparison result output enabled	R/W
3	—	The read value is undefined. The write value should be 1.	R/W
5:4	FSAMP[1:0]	Sampling Clock Select 0 0: 1/2 LOCO frequency 0 1: 1/4 LOCO frequency 1 0: 1/8 LOCO frequency 1 1: 1/16 LOCO frequency	R/W
6	RI	Voltage Monitor m Circuit Mode Select (1) case of PVDmFCR.RHSEL = 0 0: Voltage monitor m interrupt during V_{detm} passage 1: Voltage monitor m reset enabled when the voltage falls to and below V_{detm} (2) case of PVDmFCR.RHSEL = 1 0: prohibited 1: Voltage monitoring m reset enabled when the voltage rises to and up Vdetm	R/W
7	RN	Voltage Monitor m Reset Negate Select (1) case of PVDmFCR.RHSEL = 0 0: Negation follows a stabilization time (t_{PVDm}) after VCC > V_{detm} is detected 1: Negation follows a stabilization time (t_{PVDm}) after assertion of the PVDm reset. (2) case of PVDmFCR.RHSEL = 1 0: Negation follows a stabilization time (t_{PVDm}) after VCC < Vdetm is detected. 1: Prohibited	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

RIE bit (Voltage Monitor m Interrupt/Reset Enable)

The RIE bit enables or disables voltage monitor m interrupt/reset. Ensure that neither a voltage monitor m interrupt nor a voltage monitor m reset is generated during programming or erasure of the flash memory.

DFDIS bit (Voltage monitor m Digital Filter Disabled Mode Select)

The DFDIS bit disables the digital filter circuit. Set the LOCOCR.LCSTP bit to 0 (the LOCO operates) if the DFDIS bit is 0 (enabling the digital filter circuit).Set the DFDIS bit to 1 (digital filter circuit disabled) when using voltage monitor m circuit in software standby mode or Deep Software Standby mode.

CMPE bit (Voltage Monitor m Circuit Comparison Result Output Enable)

The CMPE bit enables or disables voltage monitor m circuit comparison result output. Set the CMPE bit to 1 after the voltage detection m circuit enables and stabilization time ($t_{d(E-A)}$) elapses. When stopping the voltage detection m circuit, disable the voltage detection m circuit after setting the CMPE bit is 0.

FSAMP[1:0] bits (Sampling Clock Select)

The FSAMP[1:0] bits can be rewritten only when the PVDmCR0.DFDIS bit is 1 (digital filter circuit disabled). The FSAMP[1:0] bits should not be modified when the PVDmCR0.DFDIS bit is 0 (digital filter circuit enabled).

RI bit (Voltage Monitor m Circuit Mode Select)

When the PVDmCR0.RI bit is 1 (voltage monitor m reset selected), a transition to Deep Software Standby mode 2 or 3 cannot be made, instead a transition to Deep Software Standby mode 1 is made. To enter Deep Software Standby mode 2 or 3, set all the PVDmCR0.RI bits to 0 (voltage monitor m interrupt selected).

RN bit (Voltage Monitor m Reset Negate Select)

Case of PVDmFCR.RHSEL = 0

- If the RN bit is set to 1 (negation follows a stabilization time after assertion of the PVDm reset signal), set the LOCOCR.LCSTP bit to 0 (the LOCO operates). Furthermore, if a transition to Software Standby mode or Deep Software Standby mode is to be made, the only possible value for the RN bit is 0 (negation follows a stabilization time after VCC > V_{detm} is detected. Do not set the RN bit to 1 (negation follows a stabilization time after assertion of the PVDm reset signal) when this is the case.

Case of PVDmFCR.RHSEL = 1

- Do not set the RN bit to 1.

.

7.2.4 PVDmCR1 : Voltage Monitor m Circuit Control Register 1 (m = 1, 2)

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0E0 + 0x2 × (m-1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	IRQSEL	IDTSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
1:0	IDTSEL[1:0]	Voltage Monitor m Interrupt Generation Condition Select (1) case of PVDmFCR.RHSEL = 0 0 0: When VCC ≥ V_{detm} (rise) is detected 0 1: When VCC < V_{detm} (fall) is detected 1 0: When fall and rise are detected 1 1: Settings prohibited (2) case of PVDmFCR.RHSEL = 1 This bit is not applicable.	R/W
2	IRQSEL	Voltage Monitor m Interrupt Type Select 0: Non-maskable interrupt 1: Maskable interrupt*1	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC3 bit to 1 (writing enabled) before rewriting this register.
Note 1. When enabling maskable interrupts, do not change the value of the NMIER.PVDmEN bit on the ICU side from the reset state.

7.2.5 PVDmSR : Voltage Monitor m Circuit Status Register (m = 1, 2)

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0E1 +0x2 × (m - 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	MON	DET
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
0	DET	Voltage Monitor m Voltage Change Detection Flag*1 (1) case of PVDmFCR.RHSEL = 0 0: Not detected 1: V_{detm} passage detection (2) case of PVDmFCR.RHSEL = 1 This bit is not applicable.	R/W
1	MON	Voltage Monitor 1 Signal Monitor Flag (1) case of PVDmFCR.RHSEL = 0 0: VCC ≤ V_{detm} 1: VCC > V_{detm} or MON is disabled (2) case of PVDmFCR.RHSEL = 1 This bit is not applicable.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to this bit. After writing 0 to this bit, 2 system clock (ICLK) cycles are required for the bit to be read as 0.

DET flag (Voltage Monitor m Voltage Change Detection Flag)

The DET flag is enabled when the PVDmCMPCR.PVDE bit is 1 (voltage detection m circuit enabled) and the PVDmCR0.CMPE bit is 1 (voltage monitor m circuit comparison result output enabled).

The DET flag should be set to 0 after PVDmCR0.RIE is set to 0 (disabled). PVDmCR0.RIE can be set to 1 (enabled) after a period of two or more cycles of PCLKB has elapsed.

Depending on the number of cycles of PCLKB defined for access to read an I/O register, two or more cycles than PCLKB may have to be secured as waiting time.

MON flag (Voltage Monitor 1 Signal Monitor Flag)

The MON flag is enabled when the PVDmCMPCR.PVDE bit is 1 (voltage detection m circuit enabled) and the PVDmCR0.CMPE bit is 1 (voltage monitor m circuit comparison result output enabled).

7.2.6 PVDmFCR : Voltage Monitor m Function Control Register (m = 1, 2)

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xB20 +0x4 × (m - 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	RHSEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RHSEL	Rise Hysteresis Select 0: Hysteresis level for VCC-fall detection is selected. 1: Hysteresis level for VCC-rise detection is selected.	R/W

Bit	Symbol	Function	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

The PVDmFCR.RHSEL can be changed only when all the PVDmCMPCR.PVDE bits are 0.

RHSEL flag (Rise Hysteresis Select)

When RHSEL=0 for VCC-fall detection, the hysteresis level is set above the voltage detection level set by PVDLVL[4:0].
When RHSEL=1 for VCC-rise detection, the hysteresis level is set under the voltage detection level set by PVDLVL[4:0] .
Refer to the chapter of electrical characteristics about the detail of hysteresis level of PVDm.

When PVDm reset generated by the VCC-fall detection is required, set RHSEL to 0.

When PVDm reset generated by the VCC-rise detection is required, set RHSEL to 1.

RHSEL must not be set to “1” when PVDmCR0.RI = 0.

RHSEL must not be set to “1” when PVDmCR0.RN = 1.

7.3 VCC Input Voltage Monitor

7.3.1 Monitoring V_{det0}

The comparison results from voltage monitor 0 are not available for reading.

7.3.2 Monitoring V_{detm}

After the settings are completed, results of comparison by voltage Monitor m can be monitored by using the PVDmSR.MON flag.

Table 7.2 Procedures to set up monitoring against V_{detm}

Step		Monitoring the comparison results from voltage monitor 1
Setting the voltage detection m circuit	1	Set PVDmCMPCR.PVDE = 0 to disable voltage detection m before writing to the PVDmCMPCR.PVDLVL [4:0] bits.
	2	Select the detection voltage in the PVDmCMPCR.PVDLVL[4:0] bits.
	3	Set PVDmCMPCR.PVDE = 1 to enable the voltage detection m circuit.
	4	Wait for at least $t_{d\ (E\text{-}A)}$.*1
Setting the digital filter*2	5	Select the sampling clock for the digital filter in the PVDmCR0.FSAMP[1:0] bits.
	6	Set PVDmCR0.DFDIS = 0 to enable the digital filter.
	7	Wait for at least 2n + 3 cycles of the LOCO (where n = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n).
Enabling output	8	Set PVDmCR0.CMPE = 1 to enable output of the comparison results from voltage monitor m.

Note 1. Steps 5 to 7 can be performed during the waiting time of step 4.
Note 2. Steps 5 to 7 are not required if the digital filter is not in use.

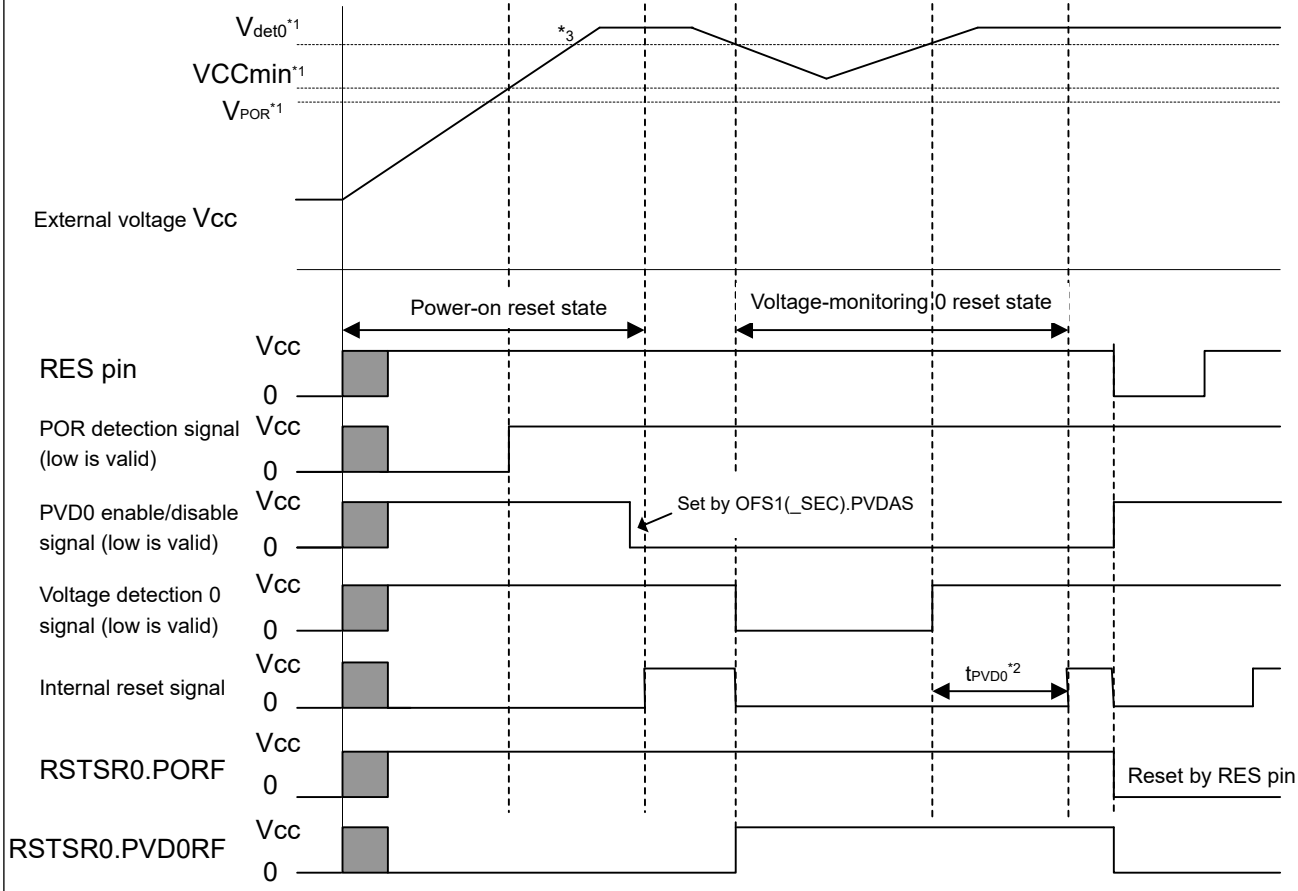
After configuring the PVDm according to the procedure in the table above, if you change the PVDm setting again, first set PVDmCR0.CMPE to 0, and then configure PVDm again according to the procedure in the table above.

When changing the PVDmCMPCR.PVDLVL, comply with the description of the section 7.2.2. PVDmCMPCR : Voltage Monitor m Comparator Control Register (m = 1, 2) .

7.4 Reset from Voltage Monitor 0

When using the reset from voltage monitor 0, clear the OFS1(_SEC).PVDAS bit to 0. (enabling the voltage monitor 0 reset after a reset)

Figure 7.3 shows an example of operations for a voltage monitor 0 reset.



Note: For details of the electrical characteristics, see section 60, Electrical Characteristics.
Note 1. V_{POR} indicates the detection level for a power-on reset and V_{det0} indicates the detection level for a voltage monitor 0 reset and VCC_{min} shows minimum guaranteed voltage of MCU.
Note 2. t_{PVD0}: the period until the voltage monitor 0 reset is released.
Note 3. At the time the power-supply voltage rises, VCC must rise to at least the minimum guaranteed voltage before release from the POR reset state.

Figure 7.3 Example of voltage monitor 0 reset operation

7.5 Interrupt and Reset from Voltage Monitor m (m = 1, 2)

An interrupt or reset can be generated in response to the results of comparison by the voltage detection m circuit

Table 7.3 shows the procedures for setting bits related to the voltage monitor m interrupt and voltage monitor m reset so that voltage monitoring operates. Table 7.4 shows the procedures for setting bits related to the voltage monitor m interrupt and voltage monitor m reset so that voltage monitoring stops. Figure 7.4 shows an example of operations for a voltage monitor m interrupt. For the operation of the voltage monitor m reset, see Figure 5.2 in section 5, Resets.

Furthermore, if you intend to use the voltage Monitor m circuit in software standby or Deep Software Standby mode, make settings for the voltage Monitor m circuit according to the following procedures.

(1) Setting in Software Standby mode

- Disable the digital filter (PVDmCR0.DFDIS = 1).
- Case of PVDmFCR.RHSEL = 0
- After VCC > V_{detm} is detected, negate the voltage monitor m reset signal (PVDmCR0.RN = 0) following a stabilization time.

(2) Settings in Deep Software Standby mode

- Disable the digital filter (PVDmCR0.DFDIS = 1).
- Enable the voltage monitor m interrupt (PVDmCR0.RI = 0). If the voltage monitor m reset is enabled (PVDmCR0.RI = 1), a transition to Deep Software Standby mode 2 or 3 is not be possible, and the transition will be to Deep Software Standby mode 1 instead.
- During Deep Software Standby mode 3, the voltage Monitor m circuit is stopped. If you intend to use the voltage Monitor m circuit in Deep Software Standby mode, do not transit to Deep Software Standby mode 3.

(3) Disabling at on-chip debug mode

- The function of the voltage Monitor m circuit is disabled by setting DBGSTOPCR.DBGSTOP_PVD at on-chip debug mode.

Table 7.3 Procedures for setting bits related to voltage monitor m interrupt and voltage monitor m reset so that voltage monitoring operates (m = 1, 2)

Step		Voltage monitor m interrupt (voltage monitor m ELC event output)	Voltage monitor m reset
Setting up the voltage detection m circuit	1	Set PVDmCMPCR.PVDE = 0 (disable voltage detection m before writing to the PVDmCMPCR.PVDLVL and PVDmFCR.RHSEL register.)	
	2	Select the detection voltage by setting the PVDmCMPCR.PVDLVL bits.	
	3	—	Select the type of the reset condition by setting the PVDmFCR.RHSEL bit.
	4	Set PVDmCMPCR.PVDE = 1 (enabling the voltage detection m circuit).	
	5	Wait for at least $t_{d\ (E\text{-}A)}$. *1	
Setting the digital filter*2	6	Select the sampling clock for the digital filter by the setting PVDmCR0.FSAMP[1:0] bits.	
	7	Set PVDmCR0.DFDIS = 0 (enabling the digital filter).	
	8	Wait for at least 2n + 3 cycles of the LOCO (where n = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n).*4.	
Setting up the voltage monitor m interrupt or reset	9	Set PVDmCR0.RI = 0 to select the voltage monitor m interrupt.	• Set PVDmCR0.RI = 1 (selecting the voltage monitor m reset). • Select the type of the reset negation by setting the PVDmCR0.RN bit.
	10	• Select the timing of interrupt requests by setting the PVDmCR1.IDTSEL[1:0] bits. • Select the type of interrupt by setting the PVDmCR1.IRQSEL bit.	—
Enabling output	11	Set PVDmSR.DET = 0.	
	12	Set PVDmCR0.RIE = 1 (enabling the voltage Monitor m interrupt or reset).*3	
	13	Set PVDmCR0.CMPE = 1 (enabling output of the results of comparison by voltage monitor m).	

Note 1. Steps 6 to 12 can be performed during the wait time in step 4.
Note 2. Steps 6 to 8 are not required if the digital filter is not in use.
Note 3. Step 12 is not required if only the ELC event signal is to be output.
Note 4. Steps 9 to 12 can be performed during the waiting time of step 8.

After configuring the PVDm according to the procedure in the Table 7.3, if you change the PVDm settings again, stop PVDm according to the procedure in Table 7.4, and then configure PVDm again according to the procedure in the Table 7.3.

When changing the PVDmCMPCR.PVDLVL, comply with the description of the section 7.2.2. PVDmCMPCR : Voltage Monitor m Comparator Control Register (m = 1, 2).

Table 7.4 Procedures for setting bits related to voltage monitor m interrupt and voltage monitor m reset so that voltage monitoring stops (m = 1, 2) (1 of 2)

Step		Voltage monitor m interrupt (voltage monitor m ELC event output), voltage monitor m reset
Stopping the enabling output	1	Set PVDmCR0.CMPE = 0 (disabling output of the results of comparison by voltage Monitor m).

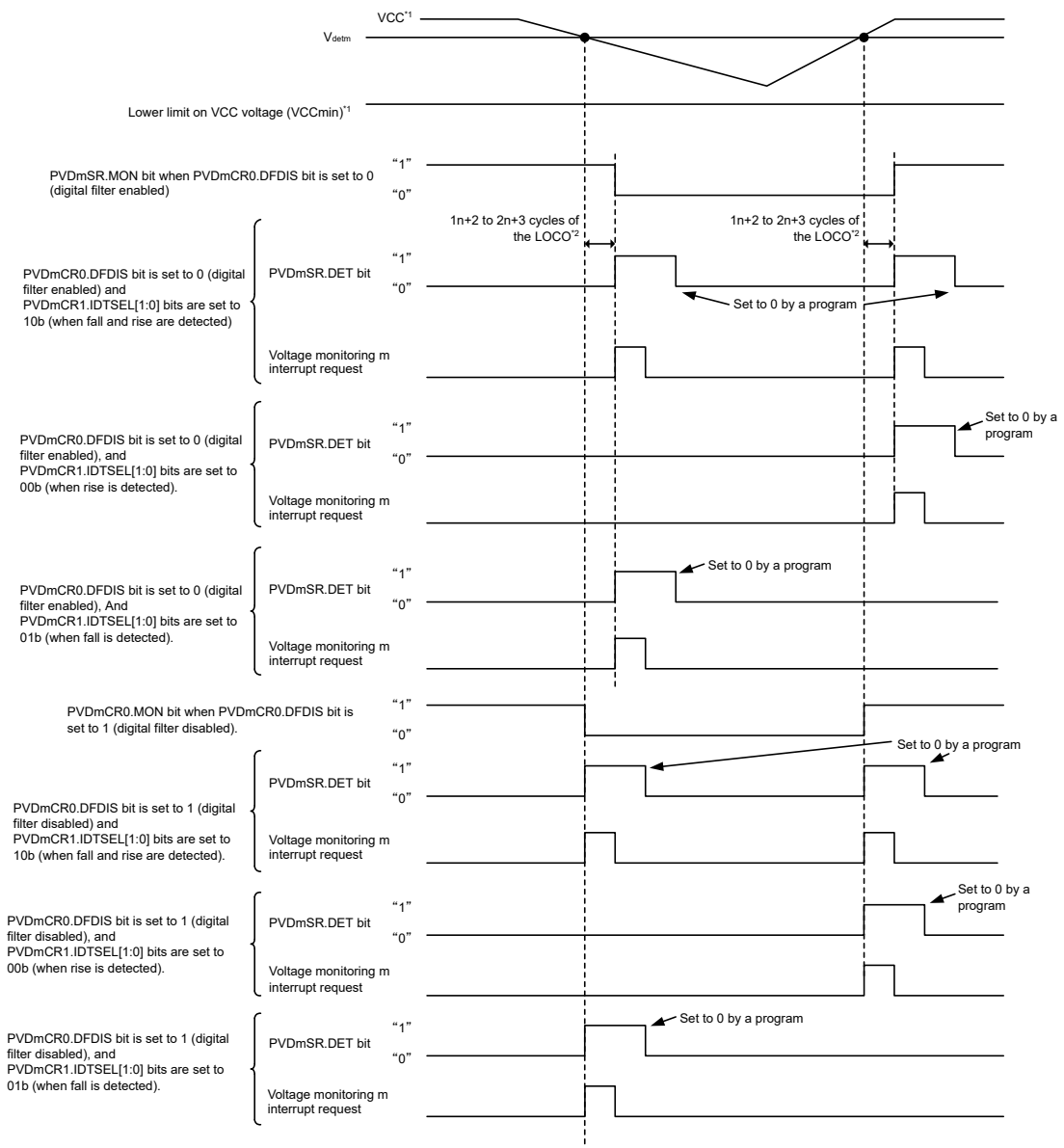
Table 7.4 Procedures for setting bits related to voltage monitor m interrupt and voltage monitor m reset so that voltage monitoring stops (m = 1, 2) (2 of 2)

Step		Voltage monitor m interrupt (voltage monitor m ELC event output), voltage monitor m reset
	2	Wait for at least 2n + 3 cycles of the LOCO (where n = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n.)*1
	3	Set PVDmCR0.RIE = 0 (disabling the voltage monitor m interrupt or reset).*3
Stopping the digital filter	4	Set PVDmCMPCR.DFDIS = 1 (disabling the digital filter). *1, *2
Stopping the voltage detection m circuit	5	Set PVDmCMPCR.PVDE = 0 (disabling the voltage detection m circuit).

Note 1. Steps 2 and 4 are not required if the digital filter is not in use.
Note 2. To disable the digital filter from its enabled state and then re-enable it, disable it and wait for at least two cycles of the LOCO before re-enabling it.
Note 3. Step 3 is not required if only the ELC event signal is to be output.

If the voltage monitor m interrupt or voltage monitor m reset setting is to be made again after it has been used and stopped once, the following steps in the procedures for stopping and making the setting can be omitted according to the condition:

- Setting or stopping the voltage detection m circuit is not required if the setting for the voltage detection m circuit is not to be changed
- Setting or stopping the digital filter is not required if the setting for the digital filter is not to be changed
- Setting the voltage monitor m interrupt or reset is not required if the setting for the voltage monitor m interrupt or voltage monitor m reset is not to be changed.



Note 1. When the voltage monitor 0 reset is not in use, use at VCC ≥ VCCmin.
Note 2. n: Frequency of the sampling clock for the digital filter is the LOCO frequency divided by n.

Figure 7.4 Example of voltage monitor m interrupt operation (m = 1, 2)

7.6 Event Link Controller (ELC) Output

The PVDm (m = 1, 2) can output the event signals to the Event Link Controller (ELC).

(1) V_{detm} passage detection event

The PVDm outputs the event signal when it is detected that the voltage has passed the V_{detm} voltage while both the voltage detection m circuit and the voltage monitor m circuit comparison result output are enabled.

When enabling the event link output function of the PVDm, be sure to specify settings to enable the PVDm before enabling the PVDm event link function of the ELC. To stop the event link output function of the PVDm, be sure to specify settings to stop the PVDm before disabling the PVDm event link function of the ELC.

7.6.1 Interrupt Handling and Event Linking

Each PVD has bits to separately enable or disable the voltage monitor m interrupts. When an interrupt source is generated and the interrupt is enabled by the interrupt enable bit, the interrupt signal is output to the CPU.

On the contrary, as soon as an interrupt source is generated, the event link signal is output as the event signal to the other module through the ELC regardless of the state of the interrupt enable bit.

It is possible to output voltage monitor m interrupts in Software Standby mode, Deep Software Standby mode 1, and Deep Software Standby mode 2. The event signals for the ELC in Software Standby mode, Deep Software Standby mode 1, and Deep Software Standby mode 2 are output as follows:

- When the event V_{detm} is detected in Software Standby mode, no event signals are generated for the ELC because no clock is presented in Software Standby mode. Because V_{detm} passage detection flag is preserved, when the supply of the clock is resumed after restoring from Software Standby mode, the event signals for the ELC is output according to the state of the V_{detm} passage detection flag.
- If event of passing V_{detm} is detected in Deep Software Standby mode, no event signal is generated for the ELC.

8. Clock Generation Circuit

8.1 Overview

The MCU provides a clock generation circuit. Table 8.1 and Table 8.2 list the clock generation circuit specifications. Figure 8.1 show a block diagram, and Table 8.3 lists the I/O pins.

Table 8.1 Clock generation circuit specifications for the clock sources (1 of 2)

Clock source	Description	Specification
Main clock oscillator (MOSC)	Resonator frequency	8 MHz to 48 MHz (For frequencies available in USB boot mode, see the application note of the boot firmware)
	External clock input frequency	Up to 48 MHz
	Connectable resonator or additional circuit: ceramic resonator, crystal Connection pins: EXTAL, XTAL	Available
	Drive capability switching	Available
	Automatic Gain Control Function	Available
	Oscillation stop detection function	Available
Sub-clock oscillator (SOSC)	Resonator frequency	32.768 kHz
	Connectable resonator or additional circuit: crystal resonator Connection pin: XCIN, XCOUT	Available
	External clock input	Available
	Drive capability switching	Available
PLL1 circuit PLL2 circuit	Input clock source	MOSC/HOCO
	Input pulse frequency division ratio	Selectable from 1/2/3/4
	Input clock frequency	8 MHz to 48 MHz
	Input clock frequency (After input frequency division)	6 MHz to 12 MHz
	Frequency multiplication ratio	Selectable from 53 to 180 (after the decimal point : 0/0.33/0.50/0.66)
	VCO frequency	640 MHz to 1440 MHz
	Number of output clocks	Output 3 different clocks
	PLL Output clock P	40 MHz to 480 MHz (output division ratio : 2/4/6/8/16)
	PLL Output clock Q	71 MHz to 480 MHz (output division ratio : 2/3/4/5/6/8/9)
	PLL Output clock R	71 MHz to 480 MHz (output division ratio : 2/3/4/5/6/8/9)
High-speed on-chip oscillator (HOCO)	Oscillation frequency	16/18/20/32/48 MHz
	FLL function	Available
	User trimming	Available
Middle-speed on-chip oscillator (MOCO)	Oscillation frequency	8 MHz
	User trimming	Available
Low-speed on-chip oscillator (LOCO)	Oscillation frequency	32.768 kHz
	User trimming	Available
External clock input for JTAG (TCK)	Input clock frequency	Up to 25 MHz

Table 8.1 Clock generation circuit specifications for the clock sources (2 of 2)

Clock source	Description	Specification
External clock input for SWD (SWCLK)	Input clock frequency	Up to 25 MHz

Table 8.2 Clock generation circuit specifications for the internal clocks (1 of 3)

Item	Clock source	Clock supply	Specification
CPU clock (CPUCLK)	MOSC/SOSC/HOCO/MOCO/ PLL1P	CPU, TCM	Up to 480 MHz Division ratio: 1/2/3/4/6/8/12/16/32/64
System clock (ICLK)	MOSC/SOSC/HOCO/MOCO/ PLL1P	DMAC, DTC, Flash (Reading from code flash), SRAM, Standby SRAM, System BUS, I/O Ports, ICU	Up to 240 MHz Division ratio: 1/2/3/4/6/8/12/16/32/64
Debugger clock (DCLK)	1/2 ICLK	Debug Subsystem	Up to 120 MHz
Peripheral module clock A (PCLKA)	MOSC/SOSC/HOCO/MOCO/ PLL1P	High-speed peripheral bus (ETHERC, EDMAC, USBHS, SCI, I3C, CANFD, CNECC, SPI, CRC, DOC, ADC12, DAC12, RSIP-E51A, GPT, DOTF AES clock, CEU, MIPI, DRW, LCDC)	Up to 120 MHz Division ratio: 1/2/3/4/6/8/12/16/32/64
Peripheral module clock B (PCLKB)	MOSC/SOSC/HOCO/MOCO/ PLL1P	Low-speed peripheral bus (CAC, ELC, POEG, RTC, WDT, IWDT, AGT, ULPT, IIC, USBFS, SSIE, SDHI, TSN, ACMPHS, OSPI, DOTF bus clock)	Up to 60 MHz Division ratio: 1/2/3/4/6/8/12/16/32/64
Peripheral module clock C (PCLKC)	MOSC/SOSC/HOCO/MOCO/ PLL1P	Peripheral module (ADC12 conversion clock)	Up to 60 MHz Division ratio: 1/2/3/4/6/8/12/16/32/64
Peripheral module clock D (PCLKD)	MOSC/SOSC/HOCO/MOCO/ PLL1P	Peripheral module(GPT count clock)	Up to 120 MHz Division ratio: 1/2/3/4/6/8/12/16/32/64
Peripheral module clock E (PCLKE)	MOSC/SOSC/HOCO/MOCO/ PLL1P	Peripheral module (CNECC)	Up to 240 MHz Division ratio: 1/2/3/4/6/8/12/16/32/64
FlashIF clock (FCLK)	MOSC/SOSC/HOCO/MOCO/ PLL1P	FlashIF	4 MHz to 60 MHz (P/E) Up to 60 MHz (read) Division ratio: 1/2/3/4/6/8/12/16/32/64
External bus clock (BCLK)	MOSC/SOSC/HOCO/MOCO/ PLL1P	External bus	Up to 120 MHz Division ratio: 1/2/3/4/6/8/12/16/32/64
EBCLK pin output (EBCLK)	BCLK or 1/2 BCLK	EBCLK pin	Up to 60 MHz Division ratio: 1 or 2
SDCLK pin output (SDCLK)	BCLK	SDCLK pin	Up to 120 MHz
Trace clock (TRCLK)	MOSC/SOSC/HOCO/MOCO/ PLL1P	CPU-OCD	Up to 120 MHz Division ratio: 1/2/3/4/6/8/12/16/32/64/128/256
SCI clock (SCICLK)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL1P/ PLL1Q/PLL1R/ PLL2P/PLL2Q/PLL2R	SCI	Up to 120 MHz Division ratio: 1/2/3/4/5/6/8
SPI clock (SPICLK)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL1P/PLL1Q/PLL1R/ PLL2P/ PLL2Q/PLL2R	SPI	Up to 120 MHz Division ratio: 1/2/3/4/5/6/8
Octal-SPI clock (OCTACLK)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL1P/PLL1Q/PLL1R/ PLL2P/ PLL2Q/PLL2R	Octal-SPI	Up to 200 MHz Division ratio: 1/2/3/4/5/6/8
Octal-SPI divide clock (OCTADIVCLK)	1/2 OCTACLK	Octal-SPI	Up to 100 MHz
CANFD core clock (CANFDCLK)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL1P/PLL1Q/PLL1R/ PLL2P/ PLL2Q/PLL2R	CANFD	Up to 80 MHz Division ratio: 1/2/3/4/5/6/8
LCD clock (LCDCLK)	MOCO/PLL1P/ PLL1Q/ PLL1R/PLL2P/PLL2Q/PLL2R	GLCDC	Up to 240 MHz Division ratio: 1/2/3/4/5/6/8

Table 8.2 Clock generation circuit specifications for the internal clocks (2 of 3)

Item	Clock source	Clock supply	Specification
LCD_CLK pin output (LCD_CLK) and graphic LCD pixel clock (PXCLK)	LCD_EXTCLK, LCDCLK	GLCDC	Up to 54 MHz (parallel RGB) Up to 60 MHz (serial RGB) LCD_CLK division ratios: 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 16, 24, 32 LCD_CLK : PXCLK = 1:1 (parallel RGB) LCD_CLK : PXCLK = 4:1 (serial RGB)
USB clock (USBCLK)	MOSC/HOCO/MOCO/PLL1P/PLL1Q/PLL1R/PLL2P/PLL2Q/PLL2R	USBFS, USBHS	48 MHz Division ratio: 1/2/3/4/5/6/8
USB clock (USB60CLK)	MOSC/HOCO/MOCO/PLL1P/PLL1Q/PLL1R/PLL2P/PLL2Q/PLL2R	USBHS	60 MHz Division ratio: 1/2/3/4/5/6/8
I3C clock (I3CCLK)	MOCO/PLL1P/PLL1Q/PLL1R/PLL2P/PLL2Q/PLL2R	I3C	Up to 200 MHz Division ratio: 1/2/3/4/5/6/8
Clock/buzzer output (CLKOUT)	MOSC/SOSC/HOCO/MOCO/LOCO	CLKOUT pin	Up to 60 MHz Division ratio: 1/2/4/8/16/32/64/128
MOSC clock for USBHS (USBMCLK)	MOSC	USBHS	12/20/24/48 MHz No division
CAN clock (CANMCLK)	MOSC	CANFD	8 MHz to 48 MHz No division
MIPI clock (MIPIMCLK)	MOSC	MIPI	Up to 48 MHz No division
ULPT LOCO clock (ULPTLCLK)	LOCO	ULPT	32.768 kHz No division
ULPT Sub-clock (ULPTSCLK)	SOSC	ULPT	32.768 kHz No division
AGT LOCOclock (AGTLCLK)	LOCO	AGT	32.768 kHz No division
AGT Sub clock (AGTSCLK)	SOSC	AGT	32.768 kHz No division
CAC Main clock (CACMCLK)	MOSC	CAC	Up to 48 MHz No division
CAC Sub clock (CACSCLK)	SOSC	CAC	32.768 kHz No division
CAC HOCO clock (CACHCLK)	HOCO	CAC	16/18/20/32/48 MHz No division
CAC MOCO clock (CACMOCLK)	MOCO	CAC	8 MHz No division
CAC LOCO clock (CACLCLK)	LOCO	CAC	32.768 kHz No division
RTC LOCO clock (RTCLCLK)	LOCO	RTC	32.768 kHz No division
RTC Sub clock (RTCSCLK)	SOCO	RTC	32.768 kHz No division
IWDT clock (IWDTCLK)	1/2 LOCO	IWDT	16.384 kHz No division
SysTick timer clock (SYSTICKCLK)	1/8 MOCO	SysTick timer	1 MHz No division
JTAG clock (JTAGTCK)	TCK	JTAG	Up to 25 MHz No division
Serial wire clock (SWCLK)	TCK	OCD	Up to 25 MHz No division

Table 8.2 Clock generation circuit specifications for the internal clocks (3 of 3)

Item	Clock source	Clock supply	Specification
TCLK pin output (TCLK)	1/2 TRCLK	TCLK pin	Up to 60 MHz

Note: When selecting PLL as the clock source, set the PLL output frequency to 480 MHz or less.
Note: If the PLL reference clock source is HOCO, the PLL multiplication setting must be set within output frequency range of PLL in consideration of HOCO frequency (minimum/maximum).
Note: Restrictions on setting clock frequency: CPUCLK ≥ ICLK, ICLK ≥ PCLKA ≥ PCLKB, ICLK ≥ FCLK, ICLK ≥ BCLK, PCLKD ≥ PCLKA ≥ PCLKB
Restrictions on clock frequency ratio: (N: integer, and up to 64)
CPUCLK:ICLK = N:1, ICLK:FCLK = N:1, ICLK:BCLK = N:1, ICLK:PCLKA = N:1, ICLK:PCLKB = N:1, ICLK:PCLKC = N:1 or 1:N, ICLK:PCLKD = N:1 or 1:N, ICLK:PCLKE = N:1 or 1:N
When one clock selects 3, 6, 12-divisions, the other clocks set by SCKDIVCR and SCKDIVCR2 registers must not select 2, 4, 8, 16, 32, 64- divisions.
If the A/D converter is enabled, the clock frequency ratio is constrained as follows:
PCLKA:PCLKC = 1:1 or 2:1 or 4:1 or 8:1 or 1:2 or 1:4
If the CAN-FD is used, clock frequency ratio is constrained to be PCLKA:PCLKE = 1:2.
Note: Restrictions on the minimum FCLK frequency 4MHz when P/E.
Note: The multiplication of PLL1 and PLL2 should be set to be within the output frequency range of PLL1 and PLL2, taking the frequency of HOCO into consideration when not using the FLL function.
Note: Clocks have a permissible frequency range (See Table 8.2).
Flash memory and SRAM also have a permissible operating frequency range in each wait cycle setting. (See section 50, SRAM, section 52, Flash Memory)
Those clock frequency ranges must be satisfied even if the HOCO has its maximum or minimum frequency when not using FLL function. (See section 60, Electrical Characteristics).
Note: When using ETHERC, the PCLKA frequency is as follows:
12.5 MHz ≤ PCLKA ≤ 120 MHz
Note: PLLCCR.PLSRCSEL bit must be set to 0 when CPUCLK is set to over 360 MHz.

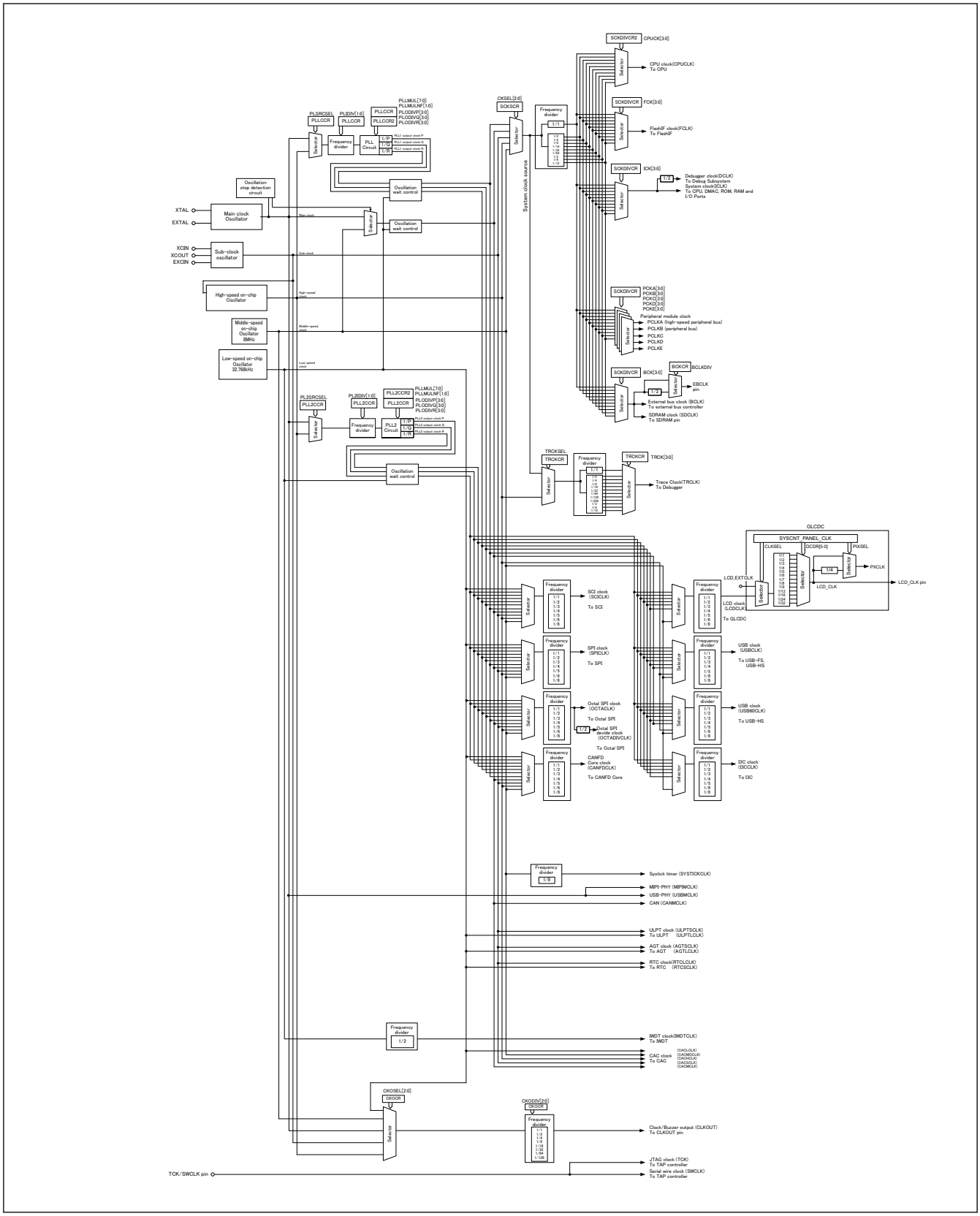


Figure 8.1 Clock generation circuit block diagram

Table 8.3 lists the input/output pins of the clock generation circuit.

Table 8.3 Input/Output pins of clock generation circuit

Pin name	I/O	Description
XTAL	Output	These pins are used to connect a ceramic resonator or crystal resonator. The EXTAL pin can also be used to input an external clock. For details, see section 8.3.2. External Clock Input.
EXTAL	Input	
XCIN	Input	These pins are used to connect a 32.768-kHz crystal resonator
XCOUT	Output	
EXCIN	Input	External sub-clock input
TCK/SWCLK	Input	This pin is used to input the clock for the JTAG/SWD
EBCLK	Output	This pin is used to supply external devices with the external bus clock (EBCLK)
SDCLK	Output	SDRAM clock (SDCLK) supply for external devices
CLKOUT	Output	This pin is used to output the CLKOUT clock
TRCLK	Output	TRACE clock output

8.2 Register Descriptions

8.2.1 CGFSAR : Clock Generation Function Security Attribute Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x3C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	NONSEC26	—	—	—	NONSEC22	NONSEC21	NONSEC20	NONSEC19	NONSEC18	NONSEC17	NONSEC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	NONSEC13	NONSEC12	NONSEC11	—	NONSEC09	NONSEC08	NONSEC07	NONSEC06	NONSEC05	NONSEC04	NONSEC03	NONSEC02	—	NONSEC00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NONSEC00	Non-secure Attribute bit 00 Target register: SCKDIVCR, SCKDIVCR2, SCKSCR Target factor: system clock control 0: Secure 1: Non-secure	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	NONSEC02	Non-secure Attribute bit 02 Target register: HOCOCR, HOCOCR2, FLLCR1, FLLCR2, HOCOUTCR, HOCOSCR Target factor: HOCO 0: Secure 1: Non-secure	R/W
3	NONSEC03	Non-secure Attribute bit 03 Target register: MOCOCR, MOCOUTCR Target factor: MOCO 0: Secure 1: Non-secure	R/W
4	NONSEC04	Non-secure Attribute bit 04 Target register: LOCOCR, LOCOUTCR Target factor: LOCO 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
5	NONSEC05	Non-secure Attribute bit 05 Target register: MOSCCR, MOSCWTCR, MOMCR, MOSCSCR Target factor: MOSC 0: Secure 1: Non-secure	R/W
6	NONSEC06	Non-secure Attribute bit 06 Target register: OSTDCR, OSTDSR Target factor: oscillation stop detection control 0: Secure 1: Non-secure	R/W
7	NONSEC07	Non-secure Attribute bit 07 Target register: SOSCCR, SOMCR Target factor: SOSC 0: Secure 1: Non-secure	R/W
8	NONSEC08	Non-secure Attribute bit 08 Target register: PLLCCR, PLLCCR2, PLLCR Target factor: PLL 0: Secure 1: Non-secure	R/W
9	NONSEC09	Non-secure Attribute bit 09 Target register: PLL2CCR, PLL2CCR2, PLL2CR Target factor: PLL2 0: Secure 1: Non-secure	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	NONSEC11	Non-secure Attribute bit 11 Target register: CKOCR Target factor: CLKOUT control 0: Secure 1: Non-secure	R/W
12	NONSEC12	Non-secure Attribute bit 12 Target register: BCKCR, EBCKOCR Target factor: EBCLK 0: Secure 1: Non-secure	R/W
13	NONSEC13	Non-secure Attribute bit 13 Target register: SDCKOCR Target factor: SDCLK 0: Secure 1: Non-secure	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	NONSEC16	Non-secure Attribute bit 16 Target register: USBCKDIVCR, USBCKCR Target factor: USBCLK 0: Secure 1: Non-secure	R/W
17	NONSEC17	Non-secure Attribute bit 17 Target register: OCTACKDIVCR, OCTACKCR Target factor: OCTACLK 0: Secure 1: Non-secure	R/W
18	NONSEC18	Non-secure Attribute bit 18 Target register: CANFDCKDIVCR, CANFDCKCR Target factor: CANFDCLK 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
19	NONSEC19	Non-secure Attribute bit 19 Target register: USB60CKDIVCR, USB60CKCR Target factor: USB60CLK 0: Secure 1: Non-secure	R/W
20	NONSEC20	Non-secure Attribute bit 20 Target register:I3CCKDIVCR, I3CCKCR Target factor: I3CCLK 0: Secure 1: Non-secure	R/W
21	NONSEC21	Non-secure Attribute bit 21 Target register: SCICKDIVCR, SCICKCR Target factor: SCICLK 0: Secure 1: Non-secure	R/W
22	NONSEC22	Non-secure Attribute bit 22 Target register: SPICKDIVCR, SPICKCR Target factor: SPICLK 0: Secure 1: Non-secure	R/W
25:23	—	These bits are read as 0. The write value should be 0.	R/W
26	NONSEC26	Non-secure Attribute bit 26 Target register: LCDCKDIVCR, LCDCKCR Target factor: LCDCLK 0: Secure 1: Non-secure	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1
Note: Set the PRCR.PRC4 bit to 1 (write enabled) before rewriting this register.

CGFSAR register controls the secure attribute of Clock Generation Function registers.

NONSEC00 bit (Non-secure Attribute bit 00)

This bit controls the security attribute of SCKDIVCR, SCKDIVCR2, SCKSCR.

NONSEC02 bit (Non-secure Attribute bit 02)

This bit controls the security attribute of HOCOCR, HOCOCR2, FLLCR1, FLLCR2, HOCOUTCR, HOCOSCR.

NONSEC03 bit (Non-secure Attribute bit 03)

This bit controls the security attribute of MOCOCR, MOCOUTCR.

NONSEC04 bit (Non-secure Attribute bit 04)

This bit controls the security attribute of LOCOCR, LOCOUTCR.

NONSEC05 bit (Non-secure Attribute bit 05)

This bit controls the security attribute of MOSCCR, MOSCWTCR, MOMCR, MOSCSCR.

NONSEC06 bit (Non-secure Attribute bit 06)

This bit controls the security attribute of OSTDCR, OSTDSR.

NONSEC07 bit (Non-secure Attribute bit 07)

This bit controls the security attribute of SOSCCR, SOMCR.

NONSEC08 bit (Non-secure Attribute bit 08)

This bit controls the security attribute of PLLCCR, PLLCCR2, PLLCR.

NONSEC09 bit (Non-secure Attribute bit 09)

This bit controls the security attribute of PLL2CCR, PLL2CCR2, PLL2CR.

NONSEC11 bit (Non-secure Attribute bit 11)

This bit controls the security attribute of CKOCR.

NONSEC12 bit (Non-secure Attribute bit 12)

This bit controls the security attribute of BCKCR, EBCKOCR.

NONSEC13 bit (Non-secure Attribute bit 11)

This bit controls the security attribute of SDCKOCR.

NONSEC16 bit (Non-secure Attribute bit 16)

This bit controls the security attribute of USBCKDIVCR, USBCKCR.

NONSEC17 bit (Non-secure Attribute bit 17)

This bit controls the security attribute of OCTACKDIVCR, OCTACKCR.

NONSEC18 bit (Non-secure Attribute bit 18)

This bit controls the security attribute of CANFDCKDIVCR, CANFDCKCR.

NONSEC19 bit (Non-secure Attribute bit 19)

This bit controls the security attribute of USB60CKDIVCR, USB60CKCR.

NONSEC20 bit (Non-secure Attribute bit 20)

This bit controls the security attribute of I3CCKDIVCR, I3CCKCR.

NONSEC21 bit (Non-secure Attribute bit 21)

This bit controls the security attribute of SCICKDIVCR, SCICKCR.

NONSEC22 bit (Non-secure Attribute bit 22)

This bit controls the security attribute of SPICKDIVCR, SPICKCR.

NONSEC26 bit (Non-secure Attribute bit 26)

This bit controls the security attribute of LCDCKDIVCR, LCDCKCR.

8.2.2 SCKDIVCR : System Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	FCK[3:0]				ICK[3:0]				PCKE[3:0]				BCK[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PCKA[3:0]				PCKB[3:0]				PCKC[3:0]				PCKD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	PCKD[3:0]*4	Peripheral Module Clock D (PCLKD) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 Others: Setting prohibited.	R/W
7:4	PCKC[3:0]*4	Peripheral Module Clock C (PCLKC) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 Others: Setting prohibited.	R/W
11:8	PCKB[3:0]*3	Peripheral Module Clock B (PCLKB) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 Others: Setting prohibited.	R/W
15:12	PCKA[3:0]*3	Peripheral Module Clock A (PCLKA) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 Others: Setting prohibited.	R/W
19:16	BCK[3:0]*2	External Bus Clock (BCLK) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 Others: Settings prohibited	R/W

Bit	Symbol	Function	R/W
23:20	PCKE[3:0]*4	Peripheral Module Clock E (PCLKE) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 Others: Settings prohibited	R/W
27:24	ICK[3:0]*1*2*3*4*5	System Clock (ICLK) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 Others: Setting prohibited.	R/W
31:28	FCK[3:0]*1	FlashIF Clock (FCLK) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 Others: Setting prohibited.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: When one clock selects 3, 6, 12-divisions, the other clocks set by SCKDIVCR and SCKDIVCR2 registers must not select 2, 4, 8, 16, 32, 64- divisions.
Note 1. The following relation is required between the frequencies of the system clock (ICLK) and the FlashIF clock (FCLK).
ICLK:FCLK=N:1 (N: integer)
Note 2. The following relation is required between the frequencies of the system clock (ICLK) and the external bus clock (BCLK).
ICLK:BCLK=N:1 (N: integer)
Note 3. The following relation is required between the frequencies of the system clock (ICLK) and the peripheral module clocks (PCLKA, PCLKB)
ICLK:PCLKA = N:1, ICLK:PCLKB = N:1 (N: integer)
Note 4. The following relation is required between the frequencies of the system clock (ICLK) and the peripheral module clocks (PCLKC, PCLKD, PCLKE):
ICLK:PCLKC, PCLKD PCLKE = N:1or1:N (N: integer)
Note 5. The following relation is required between the frequencies of the CPU clock (CPUCLK) and the system clock (ICLK):
CPUCLK:ICLK=N:1 (N: integer)

SCKDIVCR selects the frequencies of the system clock (ICLK), peripheral module clock (PCLKA, PCLKB, PCLKC, PCLKD, PCLKE), FlashIF clock (FCLK), and external bus clock (BCLK).

8.2.3 SCKDIVCR2 : System Clock Division Control Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x24

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—				CPUCK[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CPUCK[3:0]*1	CPU Clock (CPUCLK) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 Others: Setting prohibited.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: When one clock selects 3, 6, 12-divisions, the other clocks set by SCKDIVCR and SCKDIVCR2 registers must not select 2, 4, 8, 16, 32, 64- divisions.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. The following relation is required between the frequencies of the CPU clock (CPUCLK) and the system clock (ICLK):
CPUCLK:ICLK = N:1 (N: integer)

SCKDIVCR2 selects the frequencies of the cpu clock (CPUCLK).

8.2.4 SYRACCR : System Register Access Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xCC

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	BUSY
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BUSY	Access Ready monitor 0: Ready to read/write access 1: Writing in progress	R
7:1	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE-5, P-TYPE-5
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

SYRACCR register monitor that the target registers can be accessed.

The target registers are LOCOCR and LOCOUTCR.

BUSY bit (Access Ready monitor)

When SYRACCR.BUSY is 0, you can access the target registers.

Make sure that SYRACCR.BUSY is 0 before accessing the target registers.

When SYRACCR.BUSY is 1, the write access to one of the target registers is in progress. If SYRACCR.BUSY is 1, further writing to one of the target registers is ignored, and the read value from the target registers is not guaranteed.

8.2.5 SCKSCR : System Clock Source Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x026

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CKSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
2:0	CKSEL[2:0]	Clock Source Select 0 0 0: HOCO 0 0 1: MOCO (Value after reset) 0 1 0: Setting prohibited 0 1 1: Main clock oscillator (MOSC) 1 0 0: Sub-clock oscillator (SOSC) 1 0 1: PLL1 output clock P (PLL1P) 1 1 0: Setting prohibited 1 1 1: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: S-TYPE-3, P-TYPE-2

The SCKSCR register selects system clock source.

CKSEL[2:0] bits (Clock Source Select)

The CKSEL[2:0] bits select the source for the following modules:

- CPU clock (CPUCLK)
- System clock (ICLK)
- Peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD and PCLKE)
- FlashIF clock (FCLK)
- external bus clock (BCLK)
- SDRAM clock (SDCLK)

The bits select from one of the following sources:

- Middle-speed on-chip oscillator (MOCO)
- High-speed on-chip oscillator (HOCO)
- Main clock oscillator (MOSC)
- Sub-clock oscillator (SOSC)
- PLL1 output clock P (PLL1P)

The operating state of each clock source is controlled not only by the clock oscillation enable settings but also by the operating modes of the product. Some clock sources might be forcibly stopped depending on the product operating mode being used.

Check the operation state of clock sources in each product operating mode, and do not select the clock source to be stopped in SCKSCR. The clock sources should be switched when there are no occurring internal asynchronous interrupt. For details, see section 10, Low Power Modes.

8.2.6 PLLCCR : PLL Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x028

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PLLMUL[7:0]								PLLMULNF[1:0]		—	PLSR CSEL	—	—	PLIDIV[1:0]	
Value after reset:	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PLIDIV[1:0]*1	PLL1 Input Frequency Division Ratio Select 0 0: 1/1 0 1: 1/2 1 0: 1/3 1 1: 1/4	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	PLSRCSEL	PLL1 Clock Source Select 0: Main clock oscillator*3 1: HOCO*4	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
7:6	PLLMULNF[1:0]*2	PLL1 Frequency Multiplication Fractional Factor Select 0 0: 0.00 (Value after reset) 0 1: 0.33 (1/3) 1 0: 0.66 (2/3) 1 1: 0.50 (1/2)	R/W
15:8	PLLMUL[7:0]*2	PLL1 Frequency Multiplication Factor Select 0x19: × 26 (Value after reset) 0x34: × 53 0x35: × 54 ⋮ 0x58: × 89 0x59: ×90 0x5A: × 91 ⋮ 0xB2: ×179 0xB3: × 180 Others: Setting prohibited.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. PLIDIV[1:0] should be set so that the frequency of PLL1 input signal is within the range of section 8.1. Overview.
Note 2. PLLMUL[7:0] and PLLMULNF[1:0] should be set so that the frequency of PLL1 output signal is within the range of section 8.1. Overview.
Note 3. PLSRCSEL must be set to 0 when CPUCLK is set to over 360MHz.
Note 4. The FLL function must be enabled when using USBCLK.

The PLLCCR register sets the operation of the PLL1 circuit.

Writing to the PLLCCR is prohibited when the PLLCR.PLLSTP bit is 0 (the PLL1 operates).

PLIDIV[1:0] bits (PLL1 Input Frequency Division Ratio Select)

These bits select the frequency division ratio of the PLL1 clock source.

PLSRCSEL bit (PLL1 Clock Source Select)

This bit selects the clock source for the PLL1.

PLLMULNF[1:0] bit (PLL1 Frequency Multiplication Fractional Factor Select)

These bits select the fractional part of the frequency multiplication factor for the PLL1 circuit.

PLLMUL[7:0] bits (PLL1 Frequency Multiplication Factor Select)

These bits select the frequency multiplication factor of the PLL1 circuit.

8.2.7 PLLCCR2 : PLL Clock Control Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x4C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—				PLODIVR[3:0]				PLODIVQ[3:0]				PLODIVP[3:0]			
Value after reset:	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1

Bit	Symbol	Function	R/W
3:0	PLODIVP[3:0]*1	PLL1 Output Frequency Division Ratio Select for output clock P 0001: × 1/2 0011: × 1/4 0101: × 1/6 (Value after reset) 0111: × 1/8 1111: × 1/16 Others: Setting prohibited.	R/W
7:4	PLODIVQ[3:0]*1	PLL1 Output Frequency Division Ratio Select for output clock Q 0001: × 1/2 0010: × 1/3 0011: × 1/4 0100: × 1/5 0101: × 1/6 (Value after reset) 0111: × 1/8 1000: × 1/9 Others: Setting prohibited.	R/W
11:8	PLODIVR[3:0]*1	PLL1 Output Frequency Division Ratio Select for output clock R 0001: × 1/2 0010: × 1/3 0011: × 1/4 0100: × 1/5 0101: × 1/6 (Value after reset) 0111: × 1/8 1000: × 1/9 Others: Setting prohibited.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. It must be set so that the frequency of PLL1 output signal is within the range listed in Table 8.1.

The PLLCCR2 register sets up the operation of the PLL1 circuit. Writing to the PLLCCR2 is prohibited when the PLLCR.PLLSTP bit is 0 (the PLL1 operates).

Even if only one of the PLL1 output clock P, Q, or R is used, PLLCCR2 must be set so that the frequency of PLL1 output signal is within the range listed in Table 8.1.

PLODIVP[3:0] bit (PLL1 Output Frequency Division Ratio Select for output clock P)

These bits select the output frequency division ratio for PLL1 output clock P (PLL1P).

PLODIVQ[3:0] bit (PLL1 Output Frequency Division Ratio Select for output clock Q)

These bits select the output frequency division ratio for PLL1 output clock Q (PLL1Q).

PLODIVR[3:0] bit (PLL1 Output Frequency Division Ratio Select for output clock R)

These bits select the output frequency division ratio for PLL1 output clock R (PLL1R).

8.2.8 PLLCR : PLL Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x02A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	PLLSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	PLLSTP	PLL1 Stop Control 0: PLL1 is operating 1: PLL1 is stopped	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The PLLCR register controls the operation of the PLL1 circuit.

PLLSTP bit (PLL1 Stop Control)

This bit runs or stops the PLL1 circuit.

If the main clock oscillator is to be selected as the clock source for the PLL1 by the PLLCCR.PLSRCSEL bit, the Main Clock Oscillator Wait Control Register (MOSCWTCR) must be set.

After the PLLSTP bit setting is changed to run the PLL1, only use the PLL1 clock after confirming that the OSCSF.PLLSF bit is set to 1. That is, a fixed time for stabilization is required after starting the PLL1 operation. A fixed time is also required for oscillation to stop after stopping the PLL1 operation. Additionally, apply the following limitations when starting and stopping the PLL1 operation by the PLLSTP bit:

- After stopping the PLL1, confirm that the OSCSF.PLLSF bit is 0 before restarting the PLL1.
- Confirm that the PLL1 is operating and that the OSCSF.PLLSF bit is 1 before stopping the PLL1.
- Regardless of whether the PLL1 clock is selected as the system clock source, confirm that the OSCSF.PLLSF is set to 1 before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode after operating the PLL1.
- When transitioning to Software Standby or Deep Software Standby mode after stopping the PLL1, confirm that the OSCSF.PLLSF bit is cleared to 0 before executing a WFI instruction.

Writing 1 to the PLLSTP bit is prohibited when SCKSCR.CKSEL[2:0] = 101 (system clock source = PLL1 output clock P (PLL1P)).

Confirm the following conditions before writing 0 to PLLSTP:

- When PLL1 source clock = MOSC: MOSCCR.MOSTP = 0 (MOSC is enabled)
- When PLL1 source clock = HOCO: HOCOCR.HCSTP = 0 (HOCO is enabled).

8.2.9 PLL2CCR : PLL2 Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x048

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PLL2MUL[7:0]								PLL2MULNF[1:0]		—	PL2SRCSEL	—	—	PL2IDIV[1:0]	
Value after reset:	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PL2IDIV[1:0]*1	PLL2 Input Frequency Division Ratio Select 0 0: 1/1 (Value after reset) 0 1: 1/2 1 0: 1/3 1 1: 1/4	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	PL2SRCSEL	PLL2 Clock Source Select 0: Main clock oscillator 1: HOCO*3	R/W
5	—	These bits are read as 0. The write value should be 0.	R/W
7:6	PLL2MULNF[1:0]*2	PLL2 Frequency Multiplication Fractional Factor Select 00: 0.00 (Value after reset) 01: 0.33 (1/3) 10: 0.66 (2/3) 11: 0.50 (1/2)	R/W
15:8	PLL2MUL[7:0]*2	PLL2 Frequency Multiplication Factor Select 0x19: × 26 (Value after reset) 0x34: × 53 0x35: × 54 ⋮ 0x58: × 89 0x59: × 90 0x5A: × 91 ⋮ 0xD2: × 179 0xD3: × 180 Others: Setting prohibited.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. PL2IDIV[1:0] should be set so that the frequency of PLL2 input signal is within the range of section 8.1. Overview.
Note 2. PLL2MUL[7:0] and PLL2MULNF[1:0] should be set so that the frequency of PLL2 output signal is within the range of section 8.1. Overview.
Note 3. The FLL function must be enabled when using USBCLK.

The PLL2CCR register sets the operation of the PLL2 circuit.

Writing to the PLL2CCR register is prohibited when the PLL2CR.PLL2STP bit is 0 (the PLL2 operates).

PL2IDIV[1:0] bits (PLL2 Input Frequency Division Ratio Select)

These bits select the frequency division ratio of the PLL2 clock source.

PL2SRCSEL bit (PLL2 Clock Source Select)

This bit selects the clock source for the PLL2.

PLL2MULNF[1:0] bit (PLL2 Frequency Multiplication Fractional Factor Select)

These bits select the fractional part of the frequency multiplication factor for the PLL2 circuit.

PLL2MUL[7:0] bits (PLL2 Frequency Multiplication Factor Select)

These bits select the frequency multiplication factor of the PLL2 circuit.

8.2.10 PLL2CCR2 : PLL2 Clock Control Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x4E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—				PL2ODIVR[3:0]				PL2ODIVQ[3:0]				PL2ODIVP[3:0]			
Value after reset:	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1

Bit	Symbol	Function	R/W
3:0	PL2ODIVP[3:0]*1	PLL2 Output Frequency Division Ratio Select for output clock P 0001: × 1/2 0011: × 1/4 0101: × 1/6 (Value after reset) 0111: × 1/8 1111: × 1/16 Others: Setting prohibited.	R/W
7:4	PL2ODIVQ[3:0]*1	PLL2 Output Frequency Division Ratio Select for output clock Q 0001: × 1/2 0010: × 1/3 0011: × 1/4 0100: × 1/5 0101: × 1/6 (Value after reset) 0111: × 1/8 1000: × 1/9 Others: Setting prohibited.	R/W
11:8	PL2ODIVR[3:0]*1	PLL2 Output Frequency Division Ratio Select for output clock R 0001: × 1/2 0010: × 1/3 0011: × 1/4 0100: × 1/5 0101: × 1/6 (Value after reset) 0111: × 1/8 1000: × 1/9 Others: Setting prohibited.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. It must be set so that the frequency of PLL2 output signal is within the range listed in Table 8.1.

The PLL2CCR2 register sets up the operation of the PLL2 circuit. Writing to the PLL2CCR2 is prohibited when the PLL2CR.PLL2STP bit is 0 (the PLL2 operates).

Even if only one of the PLL2 output clock P, Q, or R is used, PLL2CCR2 must be set so that the frequency of PLL2 output signal is within the range listed in Table 8.1.

PL2ODIVP[3:0] bit (PLL2 Output Frequency Division Ratio Select for output clock P)

These bits select the output frequency division ratio for PLL2 output clock P (PLL2P).

PL2ODIVQ[3:0] bit (PLL2 Output Frequency Division Ratio Select for output clock Q)

These bits select the output frequency division ratio for PLL2 output clock Q (PLL2Q).

PL2ODIVR[3:0] bit (PLL2 Output Frequency Division Ratio Select for output clock R)

These bits select the output frequency division ratio for PLL2 output clock R (PLL2R).

8.2.11 PLL2CR : PLL2 Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x04A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	PLL2STP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	PLL2STP	PLL2 Stop Control 0: PLL2 is operating 1: PLL2 is stopped	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The PLL2CR register controls the operation of the PLL2 circuit.

PLL2STP bit (PLL2 Stop Control)

This bit runs or stops the PLL2 circuit.

If the main clock oscillator is to be selected as the clock source for the PLL2 by the PLL2CCR.PL2SRCSEL bit, the Main Clock Oscillator Wait Control Register (MOSCWTCR) must be set.

After the PLL2STP bit setting is changed to run the PLL2, only use the PLL2 clock after confirming that the OSCSF.PLL2SF bit is set to 1. That is, a fixed time for stabilization is required after starting the PLL2 operation. A fixed time is also required for oscillation to stop after stopping the PLL2 operation. Additionally, apply the following limitations when starting and stopping the PLL2 operation by the PLL2STP bit:

- After stopping the PLL2, confirm that the OSCSF.PLL2SF bit is 0 before restarting the PLL2.
- Confirm that the PLL2 is operating and that the OSCSF.PLL2SF bit is 1 before stopping the PLL2.
- Confirm that the OSCSF.PLL2SF bit is set to 1 before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode after operating the PLL2.
- When transitioning to Software Standby or Deep Software Standby mode after stopping the PLL2, confirm that the OSCSF.PLL2SF bit is cleared to 0 before executing a WFI instruction.

Confirm the following conditions before writing 0 to PLL2STP:

- When the PLL2 source clock = MOSC: MOSCCR.MOSTP = 0 (MOSC is enabled)
- When the PLL2 source clock = HOCO: HOCOCR.HCSTP = 0 (HOCO is enabled).

8.2.12 BCKCR : External Bus Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x030

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	BCLKDIV
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BCLKDIV	BCLK Pin Output Select 0: BCLK 1: BCLK/2	R/W

Bit	Symbol	Function	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The BCKCR register controls the external bus clock.

BCLKDIV bit (BCLK Pin Output Select)

This bit selects the clock signal for output from the BCLK pin.

Either the BCLK clock with the frequency selected by the BCK[2:0] bits in SCKDIVCR or the BCLK clock divided by 2 can be selected.

8.2.13 MOSCCR : Main Clock Oscillator Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x032

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MOSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	MOSTP	Main Clock Oscillator Stop 0: Operate the main clock oscillator*1 1: Stop the main clock oscillator	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: S-TYPE-3, P-TYPE-2
Note 1. MOMCR register must be set before setting MOSTP to 0.

The MOSCCR register controls the main clock oscillator.

MOSTP bit (Main Clock Oscillator Stop)

The MOSTP bit starts or stops the main clock oscillator.

When changing the value of the MOSTP bit, execute subsequent instructions only after reading the bit to check that the value is updated.

When using the main clock, the Main Clock Oscillator Mode Oscillation Control Register (MOMCR) and the Main Clock Oscillator Wait Control Register (MOSCWTCR) must be set before setting MOSTP to 0. After setting the MOSTP bit to 0, confirm that the OSCSF.MOSCSF bit is set to 1 before using the main clock oscillator.

A fixed stabilization wait time is required after setting the main clock oscillator to start operation. A fixed wait time is also required for oscillation to stop after stopping the main clock oscillator.

The following restrictions apply when starting and stopping operation:

- After stopping the main clock oscillator, confirm that the OSCSF.MOSCSF bit is 0 before restarting the main clock oscillator
- Confirm that the main clock oscillator operates and that the OSCSF.MOSCSF bit is 1 before stopping the main clock oscillator
- Regardless of whether the main clock oscillator is selected as the system clock source, confirm that the OSCSF.MOSCSF bit is set to 1 before executing a WFI instruction to place the MCU in Software standby after operating the main clock oscillator or Deep Software Standby mode after operating the main clock oscillator.
- When a transition to Software Standby or Deep Software Standby mode is to follow the setting to stop the main clock oscillator, confirm that the OSCSF.MOSCSF bit is set to 0 before executing the WFI instruction.

Writing 1 to MOSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC)
- PLLCCR.PLSRCSEL = 0 (PLL1 source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL1P)
- PLLCCR.PLSRCSEL = 0 (PLL1 source clock = MOSC) and PLLCR.PLLSTP = 0 (PLL1 is operating)
- PLL2CCR.PL2SRCSEL = 0 (PLL2 source clock = MOSC) and PLL2CR.PLL2STP = 0 (PLL2 is operating).

8.2.14 SOSCCR : Sub-Clock Oscillator Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SOSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	SOSTP	Sub-Clock Oscillator Stop 0: Operate the sub-clock oscillator*1 1: Stop the sub-clock oscillator	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. The SOMCR register must be set before setting SOSTP to 0.

The SOSCCR register controls the sub-clock oscillator.

SOSTP bit (Sub-Clock Oscillator Stop)

The SOSTP bit starts or stops the sub-clock oscillator. When changing the value of the SOSTP bit, only execute subsequent instructions after reading the bit to check that the value is updated. Use the SOSTP bit when using the sub-clock oscillator as the source for a peripheral module, for example the RTC. When using the sub-clock oscillator, set the Sub-Clock Oscillator Mode Control Register (SOMCR) before setting SOSTP to 0.

When selecting External clock input by SOMCR.SOSEL, set SOMCR.SOSEL to 1 and wait 50µs, then set SOSTP to 0.

The following restrictions apply when starting and stopping the operation:

- After stopping the sub-clock oscillator, allow a stop interval of at least 5 SOSC clock cycles before restarting it
- After setting the SOSTP bit to 0, use the sub-clock only after the sub-clock oscillation stabilization time ($t_{SUBOSCWT}$) has elapsed.
- Regardless of whether the sub-clock oscillator is selected as the system clock source, confirm that the sub-clock oscillation is stable before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode
- When a transition to Software Standby or Deep Software Standby mode is to follow the setting to stop the sub-clock oscillator, wait for at least 3 SOSC clock cycles before executing the WFI instruction.

Writing 1 to SOSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 100b (system clock source = SOSC).

8.2.15 LOCOCR : Low-Speed On-Chip Oscillator Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x400

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	LCSTP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LCSTP	LOCO Stop 0: Operate the LOCO clock 1: Stop the LOCO clock	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: Make sure that SYRACCR.BUSY is 0 before accessing LOCOCR. When SYRACCR.BUSY is 1, the write access is ignored and the read value is not guaranteed.

The LOCOCR register controls the LOCO clock.

LCSTP bit (LOCO Stop)

The LCSTP bit starts or stops the LOCO clock.

When changing the value of the bit, only execute subsequent instructions after reading the bit and checking that its value has actually been updated.

8.2.16 HOCOCR : High-Speed On-Chip Oscillator Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x036

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	HCSTP
Value after reset:	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Function	R/W
0	HCSTP	HOCO Stop 0: Operate the HOCO clock *2*3 1: Stop the HOCO clock	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. The HCSTP bit value after a reset is 0 when the OFS1(_SEC).HOCOEN bit is 0. It is 1 when the OFS1(_SEC).HOCOEN bit is 1.
Note 2. If you are using the HOCO (HCSTP = 0), set the OFS1(_SEC).HOCOFRQ0[2:0] bit to an optimum value.
Note 3. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] even if OFS1(_SEC).HOCOFRQ0[2:0] is not appropriate value.

The HOCOCR register controls the HOCO clock.

HCSTP bit (HOCO Stop)

The HCSTP bit starts or stops the HOCO clock.

After setting the HCSTP bit to 0 to start the HOCO clock, confirm that the OSCSF.HOCOSF is set to 1 before using the clock. When OFS1(_SEC).HOCOEN is set to 0, confirm that OSCSF.HOCOSF is also set to 1 before using the HOCO

clock. A fixed stabilization wait time is required after setting the HOCO clock to start operation. A fixed wait time is also required after setting the HOCO clock to stop.

The following limitations apply when starting and stopping operation:

- After stopping the HOCO clock, wait more than HOCO stop width time and confirm that the OSCSF.HOCOSF is 0 before restarting the HOCO clock. See section 60, Electrical Characteristics for HOCO stop width time.
- Confirm that the HOCO clock operates and that the OSCSF.HOCOSF is 1 before stopping the HOCO clock.
- Regardless of whether the HOCO clock is selected as the system clock source, confirm that the OSCSF.HOCOSF is set to 1 before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode after setting HOCO operation with the HCSTP bit.
- When a transition to Software Standby or Deep Software Standby mode is to follow the setting of the HOCO clock to stop, confirm that the OSCSF.HOCOSF is set to 0 after setting the HOCO clock and before executing the WFI instruction.

Writing 1 to HCSTP is prohibited under the following conditions:

- SCKSCR.CKSEL[2:0] = 000b (system clock source = HOCO)
- PLLCCR.PLSRCSEL = 1 (PLL1 source clock = HOCO) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL1P)
- PLLCCR.PLSRCSEL = 1 (PLL1 source clock = HOCO) and PLLCR.PLLSTP = 0 (PLL1 is operating)
- PLL2CCR.PL2SRCSEL = 1 (PLL2 source clock = HOCO) and PLL2CR.PLL2STP = 0 (PLL2 is operating).

8.2.17 HOCOCR2 : High-Speed On-Chip Oscillator Control Register2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x037

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	HCFRQ0[2:0]		
Value after reset:	0	0	0	0	0	0/1*1	0/1*1	0/1*1

Bit	Symbol	Function	R/W
2:0	HCFRQ0[2:0]	HOCO Frequency Setting 0 0 0 0: 16 MHz 0 0 1: 18 MHz 0 1 0: 20 MHz 1 0 0: 32 MHz 1 1 1: 48 MHz Others: Setting prohibited.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. Value after reset of the HCFRQ0[2:0] bits depend on OFS1(_SEC).HOCOFRQ0[2:0] bits.

The HOCOCR2 register controls the HOCO clock.

Writing to the HOCOCR2 is prohibited when the HOCOCR.HCSTP bit is 0 (the HOCO operates)

HCFRQ0[2:0] bits (HOCO Frequency Setting 0)

These bits select the frequency of HOCO.

8.2.18 MOCOCR : Middle-Speed On-Chip Oscillator Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x038

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MCSTP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MCSTP	MOCO Stop 0: MOCO clock is operating 1: MOCO clock is stopped	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The MOCOCR register controls the MOCO clock.

MCSTP bit (MOCO Stop)

This bit starts or stops the MOCO clock.

When changing the value of the bit, only execute subsequent instructions after reading the bit and checking that its value has actually been updated.

There are restrictions on MOCO operation conditions, MOCO stop conditions and the setting of the MCSTP bit. For details, see section 8.6. Middle speed On Chip Oscillator (MOCO).

8.2.19 FLLCR1 : FLL Control Register1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x039

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	FLLEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FLLEN	FLL Enable 0: FLL function is disabled 1: FLL function is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: HOCO must be stopped (HOCOCR.HCSTP = 1) before FLLCR1.FLLEN is modified.
Note: SOSC must be operating with stabilization while FLL is enabled (FLLCR1.FLLEN = 1).
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The FLLCR1 register controls the FLL function of the HOCO.

FLLEN bit (FLL Enable)

This bit enables or disables the FLL function of the HOCO.

If FLL is enabled, the frequency accuracy is guaranteed after FLL is stabilized. The FLL stabilization can be checked by the CAC frequency measurement, but it must be executed after HOCO stabilization.

In addition, you must disable FLL by setting the FLLEN bit to 0 before transitioning to Software Standby mode.

Table 8.4 show an example flow of the FLL setting for each case.

Table 8.4 FLL setting flow

Step		Operation
After reset release/ Deep Software Standby cancellation	1	Start (After reset release / Deep Software Standby cancellation)
	2	FLL setting (FLLCR2.FLLCNTL)
	3	Enable FLL (FLLCR1.FLLEN = 1) Note: SOSC must be running with the oscillation stabilization.
	4	Enable HOCO (HOCOCR.HCSTP = 0)
	5	Wait for the FLL stabilization (t_{FLLWT})
	6	Check the HOCO stabilization (OSCSF.HOCOSF = 1)
	7	End (HOCO can be used.)
Software standby transition/ cancellation	1	Start (FLL is being used.)
	2	Stop HOCO (HOCOCR.HCSTP = 1) Note: If HOCO is used as the system clock source or the PLL reference clock, these clock source must be changed to another clock before HOCO is stopped.
	3	Disable FLL (FLLCR1.FLLEN = 0)
	4	WFI instruction
	5	Software standby mode
	6	Software standby cancellation
	7	Enable FLL (FLLCR1.FLLEN = 1)
	8	Enable HOCO (HOCOCR.HCSTP = 0)
	9	Wait for the FLL stabilization (t_{FLLWT})
	10	Check the HOCO stabilization (OSCSF.HOCOSF = 1)
	11	End (HOCO can be used.)

8.2.20 FLLCR2 : FLL Control Register2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x03A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	FLLCNTL[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	FLLCNTL[10:0]	FLL Multiplication Control When OFS1(_SEC).HOCOFRQ0[2:0] is 000b (16 MHz) or 100b (32 MHz), these bits must be set to 0x1E9. When OFS1(_SEC).HOCOFRQ0[2:0] is 001b (18 MHz), these bits must be set to 0x226. When OFS1(_SEC).HOCOFRQ0[2:0] is 010b (20 MHz), these bits must be set to 0x263. When OFS1(_SEC).HOCOFRQ0[2:0] is 111b (48 MHz), these bits must be set to 0x1E9. Settings other than above are prohibited.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

The FLLCR2 register controls the FLL function of the HOCO.

FLLCNTL[10:0] bits (FLL Multiplication Control)

These bits select the multiplication ratio of the FLL reference clock.

These bits must be set before FLL is enabled (FLLCR1.FLLEN = 1).

8.2.21 OSCSF : Oscillation Stabilization Flag Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x03C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	PLL2SF	PLLSF	—	MOSCSF	—	—	HOCOSF
Value after reset:	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Function	R/W
0	HOCOSF	HOCO Clock Oscillation Stabilization Flag 0: The HOCO clock is stopped or is not yet stable 1: The HOCO clock is stable, so is available for use as the system clock source	R
2:1	—	These bits are read as 0.	R
3	MOSCSF	Main Clock Oscillation Stabilization Flag 0: The main clock oscillator is stopped or is not yet stable*2 1: The main clock oscillator is stable, so is available for use as the system clock source	R
4	—	This bit is read as 0.	R
5	PLLSF	PLL1 Clock Oscillation Stabilization Flag 0: The PLL1 clock is stopped or is not yet stable. 1: The PLL1 clock is stable, so is available for use as the system clock source	R
6	PLL2SF	PLL2 Clock Oscillation Stabilization Flag 0: The PLL2 clock is stopped or is not yet stable. 1: The PLL2 clock is stable	R
7	—	This bit is read as 0.	R

Note: S-TYPE-5, P-TYPE-5
Note 1. The value after reset depends on the OFS1(_SEC).HOCOEN setting.
When OFS1(_SEC).HOCOEN = 1 (disable HOCO), the value after reset of HOCOSF is 0.
When OFS1(_SEC).HOCOEN = 0 (enable HOCO), the HOCOSF value is set to 0 immediately after reset is released, and the HOCOSF value is set to 1 after the HOCO oscillation stabilization wait time elapses.
Note 2. This is true when an appropriate value is set in the Wait Control register for the main clock oscillator. If the wait time value is not sufficient, the oscillation stabilization flag is set to 1 and supply of the clock signal to the internal circuits starts before oscillation is stable.

This register is not controlled by CGFSAR register.

The OSCSF register contains flags that indicates the operating status of the oscillation stabilization wait circuit of each oscillator. After oscillation starts, these circuits generate a waiting time until the output clock of each oscillator stabilizes. The status of each oscillation stabilization wait flag set to 1 indicates that the clock supply is stable and can be used in related circuits.

HOCOSF flag (HOCO Clock Oscillation Stabilization Flag)

The HOCOSF flag indicates the operating status of the counter that measures the wait time for the high-speed clock oscillator (HOCO). When OFS1(_SEC).HOCOEN is set to 0, confirm that OSCSF.HOCOSF is set to 1 before using the HOCO clock.

[Setting condition]

- After the HOCO clock stops and the HOCOCR.HCSTP bit is set to 0, HOCOSF is set to 1 after HOCO clock is stable to ±10% or less. If the HOCO clock is used at 1.5%, it must wait for wait time specified in the electrical characteristics. (For details, see section 60, Electrical Characteristics.)
The HOCO clock is supplied when HOCOCR.HCSTP bit is set to 0, regardless of the state of HOCOSF.

[Clearing condition]

- When the HOCO clock is operating and then is deactivated because the HOCOCR.HCSTP bit is set to 1.

Note: Even if OFS1(_SEC).HOCOEN has been set to 0, confirm that the OSCSF.HOCOSF has been set to 1 before starting to use the HOCO clock.

Note: In debugging mode, even if HOCO is oscillating, this flag is masked to 0 for the user program debugging when all of following conditions are met simultaneously:

- The Trace Clock is operating enabled by TRCKCR.TRCKEN bit is 1.
- The Trace Clock source is selected HOCO for oscillation in debug mode by TRCKCR.TRCKSEL bit is 1.
- HOCOCR.HCSTP bit is set to 1.

MOSCSF flag (Main Clock Oscillation Stabilization Flag)

The MOSCSF flag indicates the operating status of the counter that measures the wait time for the main clock oscillator.

[Setting condition]

- When the main clock oscillator is stopped and the MOSCCR.MOSTP bit is set to 0, and then the number of LOCO clock cycles corresponding to the setting of the MOSCWTCR register is counted and supply of the main clock within the MCU is started.

[Clearing condition]

- When the main clock oscillator is operating and then is deactivated because the MOSCCR.MOSTP bit is set to 1.

PLLSF flag (PLL1 Clock Oscillation Stabilization Flag)

The PLLSF flag indicates the oscillation stabilization status for the PLL1.

[Setting condition]

- After the PLL1 stops and the PLLCR.PLLSTP bit is set to 0, supply of the PLL1 clock starts after PLL1 clock is stable. If oscillation by the PLL1 clock source selected in the PLLCCR.PLSRCSEL bit is not stable when the PLLSTP bit is set to 0, waiting for the PLL1 oscillation stabilization after the PLL1 clock source oscillation is stabilized.

[Clearing condition]

- When the PLL1 is operating and then is deactivated because the PLLCR.PLLSTP bit is set to 1.

PLL2SF flag (PLL2 Clock Oscillation Stabilization Flag)

This flag indicates the oscillation stabilization status for the PLL2.

[Setting condition]

- After the PLL2 stops and the PLL2CR.PLL2STP bit is set to 0, supply of the PLL2 clock starts after PLL2 clock is stable.
  If oscillation by the PLL2 clock source selected in the PLLCCR.PLSRCSEL is not stable when the PLL2STP bit is set to 0, waiting for the PLL2 oscillation stabilization after the oscillation of the PLL2 clock source is stabilized.

[Clearing condition]

- When the PLL2 is operating and then is deactivated because the PLL2CR.PLL2STP bit is set to 1.

8.2.22 OSCMONR : Oscillator Monitor Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x43

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	LOCOMON	MOCOMON	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	MOCOMON	MOCO operation monitor 0: MOCO is set to operate. 1: MOCO is set to stop.	R
2	LOCOMON	LOCO operation monitor 0: LOCO is set to operate. 1: LOCO is set to stop.	R
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5, P-TYPE-5

This register is not controlled by CGFSAR register.

MOCOMON bit (MOCO operation monitor)

MOCOMON indicates the operation or stop setting status for MOCO.

[Clearing condition]

- When MOCO operating conditions are met. For details, see section 8.6.1. MOCO Operating conditions.

[Setting condition]

- When MOCO stop conditions are met.For details, see section 8.6.2. MOCO Stop conditions.

LOCOMON bit (LOCO operation monitor)

LOCOMON indicates the operation or stop setting status for LOCO.

[Clearing condition]

- When LOCO operating conditions are met. For details, see section 8.5.1. LOCO Operating conditions.

[Setting condition]

- When LOCO stop conditions are met. For details, see section 8.5.2. LOCO Stop conditions.

8.2.23 OSTDCR : Oscillation Stop Detection Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x040

Bit position:	7	6	5	4	3	2	1	0
Bit field:	OSTDE	—	—	—	—	—	—	OSTDIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OSTDIE	Oscillation Stop Detection Interrupt Enable 0: Disable oscillation stop detection interrupt (do not notify the POEG) 1: Enable oscillation stop detection interrupt (notify the POEG)	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	OSTDE	Oscillation Stop Detection Function Enable 0: Disable oscillation stop detection function 1: Enable oscillation stop detection function	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The OSTDCR register controls the oscillation stop detection function.

OSTDIE bit (Oscillation Stop Detection Interrupt Enable)

The OSTDIE bit enables the oscillation stop detection function interrupt. It also controls whether oscillation stop detection is reported to the POEG.

If the Oscillation Stop Detection flag in the Oscillation Stop Detection Status Register (OSTDSR.OSTDF) requires clearing, set the OSTDIE bit to 0 before clearing OSTDF. Wait for at least 2 PCLKB cycles before setting the OSTDIE bit to 1. By reading the I/O register whose access cycle number is defined by PCLKB, it is possible to secure waiting time of 2 or more cycles of PCLKB.

OSTDE bit (Oscillation Stop Detection Function Enable)

The OSTDE bit enables the oscillation stop detection function.

When the OSTDE bit is 1 (enabled), the MOCO stop bit (MOCOCR.MCSTP) is set to 0 and the MOCO operation starts. The MOCO clock cannot be stopped while the oscillation stop detection function is enabled. Writing 1 to the MOCOCR.MCSTP bit (MOCO stopped) is invalid.

When the Oscillation Stop Detection flag in the Oscillation Stop Detection Status Register (OSTDSR.OSTDF) is 1 (main clock oscillation stop detected), writing 0 to the OSTDE bit is invalid.

The following restrictions apply when using the oscillation stop detection function:

In low-speed mode, selecting division by 1, 2, 4, 8 for ICLK, FCLK, BCLK, PCLKA, PCLKB, PCLKC, PCLKD and PCLKE is prohibited.

8.2.24 OSTDSR : Oscillation Stop Detection Status Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x041

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	OSTDF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OSTDF	Oscillation Stop Detection Flag 0: Main clock oscillation stop not detected 1: Main clock oscillation stop detected	R/W*1
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. This bit can only be set to 0. This bit is cleared to 0 by writing 0 after reading it as 1.

The OSTDSR register indicates the stop detection status of the main clock oscillator.

OSTDF flag (Oscillation Stop Detection Flag)

The OSTDF flag indicates the main clock oscillator status. When this flag is 1, it indicates that the main clock oscillation stop was detected. After this stop is detected, the OSTDF flag is not set to 0 even when the main clock oscillation is restarted. The OSTDF bit is cleared to 0 by writing 0 after reading it as 1.

At least 3 ICLK cycles of wait time are required between writing 0 to OSTDF and reading it as 0. If the OSTDF flag is set to 0 when the main clock oscillation is stopped, the OSTDF flag becomes 0 then returns to 1.

The OSTDF flag cannot be set to 0 under the following conditions:

- SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC)
- PLLCCR.PLSRCSEL = 0 (PLL1 source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (System clock source = PLL1P).

The OSTDF flag must be set to 0 after switching the clock source to sources other than the main clock oscillator and PLL1P.

[Setting condition]

- The main clock oscillator is stopped when OSTDCR.OSTDE = 1 (oscillation stop detection function enabled).

[Clearing condition]

- 1 is read and then 0 is written when the SCKSCR.CKSEL[2:0] bits are neither 011b (system clock source is MOSC) nor, 101b (system clock source is PLL1P) and PLLCCR.PLSRCSEL bit is not 0 (PLL1 source clock is MOSC).

8.2.25 MOSCWTCR : Main Clock Oscillator Wait Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0A2

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	MSTS[3:0]			
Value after reset:	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
3:0	MSTS[3:0]	Main Clock Oscillator Wait Time Setting 0x0: Wait time = 3 cycles (11.4 µs) 0x1: Wait time = 35 cycles (133.5 µs) 0x2: Wait time = 67 cycles (255.6 µs) 0x3: Wait time = 131 cycles (499.7 µs) 0x4: Wait time = 259 cycles (988.0 µs) 0x5: Wait time = 547 cycles (2086.6 µs) 0x6: Wait time = 1059 cycles (4039.8 µs) 0x7: Wait time = 2147 cycles (8190.2 µs) 0x8: Wait time = 4291 cycles (16368.9 µs) 0x9: Wait time = 8163 cycles (31139.4 µs) Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

MSTS[3:0] bits (Main Clock Oscillator Wait Time Setting)

The MSTS[3:0] bits specify the oscillation stabilization wait time for the main clock oscillator.

Set the main clock oscillation stabilization time to a period longer than or equal to the stabilization time recommended by the oscillator manufacturer. When the main clock is input externally, set these bits to 0x0 because the oscillation stabilization time is not required.

The wait time set in these bits is counted using: 1 cycle (µs) = $1/(f_{LOCO}[MHz] \times 8) = 1/(0.032768 \times 8) = 3.81$ (µs) (min.). The LOCO clock automatically oscillates when necessary, regardless of the value of the LOCO.LCSTP bit. After the specified wait time elapses, supply of the main clock starts internally in the MCU, and the OSCSF.MOSCSF flag is set to 1. If the specified wait time is short, supply of the main clock starts before oscillation of the clock becomes stable.

Only rewrite the MOSCWTCR register when the MOSCCR.MOSTP bit is 1 and the OSCSF.MOSCSF flag is 0. Do not rewrite this register under any other conditions.

8.2.26 MOMCR : Main Clock Oscillator Mode Oscillation Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA50

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	MOSEL	—	—	MODRV0[2:0]			—
Value after reset:	0	0	0	1	1	0	1	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
3:1	MODRV0[2:0]	Main Clock Oscillator Drive Capability 0 Switching 0 0 0: 8 MHz 0 1 1: 8 MHz to 24 MHz 1 0 1: 8 MHz to 48 MHz Others: Setting prohibited	R/W
4	—	This bit is read as 1. The write value should be 1.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	MOSEL	Main Clock Oscillator Switching 0: Resonator 1: External clock input	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: The EXTAL/XTAL pins are also used as ports. In the initial state, the pin is set as a port.
Note: The MOSCCR.MOSTP bit must be 1 (MOSC is stopped) before changing this register.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

MODRV0[2:0] bits (Main Clock Oscillator Drive Capability 0 Switching)

The MODRV0[2:0] bits switches the drive capability for the main clock oscillator. The drive capability of Main Clock Oscillator needs to choose suitable drive capability according to the frequency.

MOSEL bit (Main Clock Oscillator Switching)

The MOSEL bit switches the source for the main clock oscillator.

8.2.27 SOMCR : Sub-Clock Oscillator Mode Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	SOSEL	—	—	—	—	SODRV[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SODRV[1:0]	Sub-Clock Oscillator Drive Capability Switching 0 0: Standard (12.5pf) 0 1: Lowpower mode 1 (9pf) 1 0: Lowpower mode 2 (7pf) 1 1: Lowpower mode 3 (4pf)	R/W
5:2	—	These bits are read as 0. The write value should be 0.	R/W
6	SOSEL	Sub-Clock Oscillator Switching 0: Resonator 1: External clock input	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The SOMCR register must be modified when SOSCCR.SOSTP is 1 (SOSC is stopped).

SODRV[1:0] bits (Sub-Clock Oscillator Drive Capability Switching)

The SODRV[1:0] bits switch the drive capability of the sub-clock oscillator.

SOSEL bit (Sub-Clock Oscillator Switching)

The SOSEL bits switchs the source for the Sub-clock oscillator.

For details, see section 8.4. Sub-Clock Oscillator.

If VCC voltage of the MCU is lower than 1.8 V and this bit is set to 1, RTC registers and VBTBKR[n](n = 0 t o127) (VBATT Backup Register) can be accessed 20 µs after the setting of this bit.

8.2.28 CKOCR : Clock Out Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x03E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CKOEN	CKODIV[2:0]			—	CKOSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
2:0	CKOSEL[2:0]	Clock Out Source Select 0 0 0: HOCO 0 0 1: MOCO (Value after reset) 0 1 0: LOCO 0 1 1: MOSC 1 0 0: SOSC Others: Setting prohibited	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	CKODIV[2:0]	Clock Output Frequency Division Ratio 0 0 0: × 1/1 0 0 1: × 1/2 0 1 0: × 1/4 0 1 1: × 1/8 1 0 0: × 1/16 1 0 1: × 1/32 1 1 0: × 1/64 1 1 1: × 1/128	R/W
7	CKOEN	Clock Out Enable 0: Disable clock out 1: Enable clock out	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

CKOSEL[2:0] bits (Clock Out Source Select)

The CKOSEL[2:0] bits select the source of the clock to be output from the CLKOUT pin. When changing the clock source, set the CKOEN bit to 0.

CKODIV[2:0] bits (Clock Output Frequency Division Ratio)

The CKODIV[2:0] bits specify the clock division ratio. Set the CKOEN bit to 0 when changing the division ratio.

CKOEN bit (Clock Out Enable)

The CKOEN bit enables output from the CLKOUT pin.

When this bit is set to 1, the selected clock is output. When this bit is set to 0, low is output. When changing this bit, confirm that the clock out source clock selected in the CKOSEL[2:0] bits is stable. Otherwise, a glitch might be generated in the output.

Clear this bit before entering Software Standby or Deep Software Standby mode if the selecting clock out source clock is stopped in that mode.

8.2.29 EBCKOCR : External Bus Clock Output Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x052

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	EBCKOEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EBCKOEN	EBCLK Pin Output Control 0: EBCLK pin output is disabled (fixed high) 1: EBCLK pin output is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

8.2.30 SDCKOCR : SDRAM Clock Output Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x053

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SDCKOEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SDCKOEN	SDCLK Pin Output Control 0: SDCLK pin output is disabled. (Fixed high) 1: SDCLK pin output is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

SDCKOEN bit (SDCLK Pin Output Control)

This bit selects controls the output of the SDCLK pin.

8.2.31 LOCOUTCR : LOCO User Trimming Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x402

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LOCOUTRM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	LOCOUTRM[7:0]	LOCO User Trimming 0x80: -128 0x81: -127 ⋮ 0xFF: -1 0x00: Center Code 0x01: +1 ⋮ 0x7E: +126 0x7F: +127	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: Make sure that SYRACCR.BUSY is 0 before accessing LOCOUTCR. When SYRACCR.BUSY is 1, the write access is ignored and the read value is not guaranteed.
Note: MCU operation is not guaranteed when LOCOUTCR is set to a value that causes the LOCO frequency to be outside of the specification range.
Note: After changing LOCOUTCR, it takes certain time until the frequency to be stable. The time is same as the oscillation stabilization time in startup.

The LOCOUTCR register is added to the original LOCO trimming data.

Changing LOCOUTCR during RTC operation is prohibited.

8.2.32 MOCOUTCR : MOCO User Trimming Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x061

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MOCOUTRM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	MOCOUTRM[7:0]	MOCO User Trimming 0x80: -128 0x81: -127 ⋮ 0xFF: -1 0x00: Center Code 0x01: +1 ⋮ 0x7E: +126 0x7F: +127	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: MCU operation is not guaranteed when MOCOUTCR is set to a value that causes the MOCO frequency to be outside of the specification range.
Note: After changing MOCOUTCR, it takes certain time until the frequency to be stable. The time is same as the oscillation stabilization time in startup.

The MOCOUTCR register is added to the original MOCO trimming data.

8.2.33 HOCOUTCR : HOCO User Trimming Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x062

Bit position:	7	6	5	4	3	2	1	0
Bit field:	HOCOUTRM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	HOCOUTRM[7:0]	HOCO User Trimming 0x80: -128 0x81: -127 ⋮ 0xFF: -1 0x00: Center Code 0x01: +1 ⋮ 0x7E: +126 0x7F: +127	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: MCU operation is not guaranteed when HOCOUTCR is set to a value that causes the HOCO frequency to be outside the specification range.
Note: After changing HOCOUTCR, it takes certain time until the frequency to be stable. The time is same as the oscillation stabilization time in startup.
Note: These bits must be 0x00 when FLL is enabled (FLLCR1.FLLEN = 1).

The HOCOUTCR register is added to the original HOCO trimming data.

8.2.34 MOSCSCR : Main Clock Oscillator Standby Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x07C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MOSCSOKP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MOSCSOKP	Main Clock Oscillator Standby Oscillation Keep Select 0: Disable 1: Enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The MOSCSCR must be changed while MOSCCR.MOSTP bit is 1 (MOSC is stopped).

If MOSCSOKP is set to 1, see section 8.12.6. Notes on MOSCSCR.MOSCSOKP.

MOSCSOKP bit (Main Clock Oscillator Standby Oscillation Keep Select)

This bit controls that Main Clock Oscillator keeps oscillating in Software Standby mode.

If this bit is disabled and MOSCCR.MOSTP bit is 0 (MOSC is run), the Main Clock Oscillator stops in Software Standby mode.

If this bit is enabled and MOSCCR.MOSTP bit is 0 (MOSC is run), the Main Clock Oscillator will keep oscillating in Software Standby mode.

If MOSCCR.MOSTP bit is 1 (MOSC is stopped), the Main Clock Oscillator stops in Software Standby mode regardless of the setting of this bit.

8.2.35 HOCOSCR : High-Speed On-Chip Oscillator Standby Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x07D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	HOCOSOKP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	HOCOSOKP	This bit is read as 0. The write value should be 0.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

8.2.36 USBCKDIVCR : USB Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x06C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	USBCKDIV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	USBCKDIV[2:0]	USB Clock (USBCLK) Division Select 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/6 1 0 0: 1/8 1 0 1: 1/3 1 1 0: 1/5 Others: Setting prohibited.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The USBCKDIVCR register controls the USB clock.

USBCKDIV[2:0] bits (USB Clock (USBCLK) Division Select)

These bits select the frequency of the USB clock (USBCLK) and must be modified when USBCKCR.USBCKSRDY = 1.

When switching the division ratio setting from n (n ≠ 1), set MSTPCRB.MSTPBi (i = 11, 12) to 1 before setting USBCKCR.USBCKSREQ = 1.

8.2.37 OCTACKDIVCR : Octal-SPI Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x06D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	OCTACKDIV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	OCTACKDIV[2:0]	Octal-SPI Clock (OCTACLK) Division Select 0 0 0: 1/1 (Value after reset) 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/6 1 0 0: 1/8 1 0 1: 1/3 1 1 0: 1/5 Others: Setting prohibited.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

OCTACKDIVCR controls the Octal-SPI clock.

OCTACKDIV[2:0] bits (Octal-SPI Clock (OCTACLK) Division Select)

These bits select the frequency of the Octal-SPI clock (OCTACLK) and must be modified when OCTACKCR.OCTACKSRDY = 1.

When switching the division ratio setting from n (n ≠ 1), set MSTPCRB.MSTPB16 to 1 before setting OCTACKCR.OCTACKSREQ = 1.

8.2.38 CANFDCKDIVCR : CANFD Core Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x06E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CANFDCKDIV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CANFDCKDIV[2:0]	CANFD core clock (CANFDCLK) Division Select 0 0 0: 1/1 (Value after reset) 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/6 1 0 0: 1/8 1 0 1: 1/3 1 1 0: 1/5 Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

CANFDCKDIVCR controls the CANFD core clock.

CANFDCKDIV[2:0] bits (CANFD core clock (CANFDCLK) Division Select)

These bits select the frequency of the CANFD core clock (CANFDCLK) and must be modified when CANFDCKCR.CANFDCKSRDY = 1.

When switching the division ratio setting from n (n ≠ 1), set MSTPCRC.MSTPC27 to 1 before setting CANFDCKCR.CANFDCKSREQ = 1.

8.2.39 USB60CKDIVCR : USB60 Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x06F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	USB60CKDIV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	USB60CKDIV[2:0]	USB clock (USB60CLK) Division Select 0 0 0: 1/1(Value after reset) 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/6 1 0 0: 1/8 1 0 1: 1/3 1 1 0: 1/5 Others: Setting prohibited.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

USB60CKDIVCR controls the USB clock.

USB60CKDIV[2:0] bits (USB clock (USB60CLK) Division Select)

These bits select the frequency of the USB clock USB60CKDIV and must be modified when USB60CKCR.USB60CKSRDY = 1.

When switching the division ratio setting from n (n ≠ 1), set MSTPCRB.MSTPB12 to 1 before setting USB60CKCR.USB60CKSREQ = 1.

8.2.40 I3CCKDIVCR : I3C Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x070

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	I3CCKDIV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	I3CCKDIV[2:0]	I3C clock (I3CCLK) Division Select 0 0 0: 1/1 (Value after reset) 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/6 1 0 0: 1/8 1 0 1: 1/3 1 1 0: 1/5 Others: Setting prohibited.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

I3CCKDIVCR controls the I3C clock.

I3CCKDIV[2:0] bits (I3C clock (I3CCLK) Division Select)

These bits select the frequency of the I3C clock (I3CCLK) and must be modified when I3CCKCR.I3CCKSRDY = 1.

When switching the division ratio setting from n (n ≠ 1), set MSTPCRB.MSTPB4 to 1 before setting I3CCKCR.I3CCKSREQ = 1.

8.2.41 USBCKCR : USB Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x074

Bit position:	7	6	5	4	3	2	1	0
Bit field:	USBCKSRDY	USBCKSREQ	—	—	USBCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	USBCKSEL[3:0]	USB Clock (USBCLK) Source Select 0000: HOCO 0001: MOCO (value after reset) 0011: Main clock oscillator 0101: PLL1P 0110: PLL2P 0111: PLL1Q 1000: PLL1R 1001: PLL2Q 1010: PLL2R Others: Setting prohibited	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	USBCKSREQ	USB Clock (USBCLK) Switching Request 0: No request 1: Request switching	R/W
7	USBCKSRDY	USB Clock (USBCLK) Switching Ready state flag 0: Impossible to switch 1: Possible to switch	R

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The USBCKCR register controls the USB clock.

When switching the clock source, ensure that the clock before the switch and the clock after the switch generate stable output. To change the set value of USBCKDIVCR.USBCKDIV[2:0] and USBCKSEL[3:0], use the following procedure:

1. Write 1 to MSTPCRB.MSTPBi (i = 11, 12) (only when switching the division ratio setting from n (n ≠ 1) to m (m ≠ 1)).

2. Wait for 2 USBCLK (only when switching the division ratio setting from n (n ≠ 1) to m (m ≠ 1)).
3. Write 1 to USBCKSREQ.
4. Poll until USBCKSRDY is read as 1. While USBCKSRDY = 1, no clock is output to USBCLK.
5. Write to USBCKDIVCR.USBCKDIV[2:0] and USBCKSEL[3:0].
6. Write 0 to USBCKSREQ.
7. Poll until USBCKSRDY is read as 0.
8. When USBCKSRDY becomes 0, USBCLK starts to output. Clock switching is complete.

When transitioning to Software Standby or Deep Software Standby mode, do not execute the WFI instruction while performing clock switching. That is, do not execute the WFI instruction when USBCKSREQ = 1 and USBCKSRDY = 0, or when USBCKSREQ = 0 and USBCKSRDY = 1.

USBCKSEL[3:0] bits (USB Clock (USBCLK) Source Select)

These bits select the clock source of the USB clock (USBCLK) and must be modified when USBCKCR.USBCKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

USBCKSREQ bit (USB Clock (USBCLK) Switching Request)

This bit selects the USBCLK switching request.

USBCKSRDY flag (USB Clock (USBCLK) Switching Ready state flag)

This flag indicates the state of switching ready for the USBCLK. When USBCKSRDY = 1, no clock is output to USBCLK.

8.2.42 OCTACKCR : Octal-SPI Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x075

Bit position:	7	6	5	4	3	2	1	0
Bit field:	OCTACKSRDY	OCTACKSREQ	—	—	OCTACKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	OCTACKSEL[3:0]	Octal-SPI Clock (OCTACLK) Source Select 0 0 0 0: HOCO 0 0 0 1: MOCO (value after reset) 0 0 1 0: LOCO 0 0 1 1: Main clock oscillator 0 1 0 0: Sub-clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Others: Setting prohibited	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	OCTACKSREQ	Octal-SPI Clock (OCTACLK) Switching Request 0: No request 1: Request switching	R/W
7	OCTACKSRDY	Octal-SPI Clock (OCTACLK) Switching Ready state flag 0: Impossible to switch 1: Possible to switch	R

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The OCTACKCR register controls the Octal-SPI clock.

When switching the clock source, ensure that the clock before the switch and the clock after the switch generate stable output.

To change the set value of OCTACKDIVCR.OCTACKDIV[2:0] and OCTACKSEL[3:0], use the following procedure:

1. Write 1 to MSTPCRB.MSTPB16 (only when switching the division ratio setting from n (n ≠ 1)).
2. Wait for 2 OCTACLK (only when switching the division ratio setting from n (n ≠ 1)).
3. Write 1 to OCTACKSREQ.
4. Poll until OCTACKSRDY is read as 1. While OCTACKSRDY = 1, no clock is output to OCTACLK.
5. Write to OCTACKDIVCR.OCTACKDIV[2:0] and OCTACKSEL[3:0].
6. Write 0 to OCTACKSREQ.
7. Poll until OCTACKSRDY is read as 0.
8. When OCTACKSRDY becomes 0, OCTACLK starts to output. Clock switching is complete.

When transitioning to Software Standby or Deep Software Standby mode, do not execute the WFI instruction while performing clock switching. That is, do not execute the WFI instruction when OCTACKSREQ = 1 and OCTACKSRDY = 0, or when OCTACKSREQ = 0 and OCTACKSRDY = 1.

OCTACKSEL[3:0] bits (Octal-SPI Clock (OCTACLK) Source Select)

These bits select the clock source of the Octal-SPI clock (OCTACLK) and must be modified when OCTACKCR.OCTACKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

OCTACKSREQ bit (Octal-SPI Clock (OCTACLK) Switching Request)

This bit selects the OCTACLK switching request.

OCTACKSRDY flag (Octal-SPI Clock (OCTACLK) Switching Ready state flag)

This flag indicates the state of switching ready for the OCTACLK. When OCTACKSRDY = 1, no clock is output to OCTACLK.

8.2.43 CANFDCKCR : CANFD Core Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x076

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CANFDCKSRDY	CANFDCKSREQ	—	—	CANFDCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	CANFDCKSEL[3:0]	CANFD Core Clock (CANFDCLK) Source Select 0 0 0 0: HOCO 0 0 0 1: MOCO (value after reset) 0 0 1 0: LOCO 0 0 1 1: Main clock oscillator 0 1 0 0: Sub-clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	CANFDCKSREQ	CANFD Core Clock (CANFDCLK) Switching Request 0: No request 1: Request switching	R/W
7	CANFDCKSRDY	CANFD Core Clock (CANFDCLK) Switching Ready state flag 0: Impossible to switch 1: Possible to switch	R

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The CANFDCKCR register controls the CANFD Core Clock (CANFDCLK).

When switching the clock source, ensure that the clock before the switch and the clock after the switch generate stable output. To change the set value of CANFDCKDIVCR.CANFDCKDIV[2:0] and CANFDCKSEL[3:0], use the following procedure:

1. Write 1 to MSTPCRC.MSTPC27 (only when switching the division ratio setting from n (n ≠ 1)).
2. Wait for 2 CANFDCLK (only when switching the division ratio setting from n (n ≠ 1)).
3. Write 1 to CANFDCKSREQ.
4. Poll until CANFDCKSRDY is read as 1. While CANFDCKSRDY = 1, no clock is output to CANFDCLK.
5. Write to CANFDCKDIVCR.CANFDCKDIV[2:0] and CANFDCKSEL[3:0].
6. Write 0 to CANFDCKSREQ.
7. Poll until CANFDCKSRDY is read as 0.
8. When CANFDCKSRDY becomes 0, CANFDCLK starts to output. Clock switching is complete.

When transitioning to Software Standby or Deep Software Standby mode, do not execute the WFI instruction while performing clock switching. That is, do not execute the WFI instruction when CANFDCKSREQ = 1 and CANFDCKSRDY = 0, or when CANFDCKSREQ = 0 and CANFDCKSRDY = 1.

CANFDCKSEL[3:0] bits (CANFD Core Clock (CANFDCLK) Source Select)

These bits select the clock source of the CANFD Core clock (CANFDCLK) and must be modified when CANFDCKCR.CANFDCKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

CANFDCKSREQ bit (CANFD Core Clock (CANFDCLK) Switching Request)

This bit selects the CANFDCLK switching request.

CANFDCKSRDY flag (CANFD Core Clock (CANFDCLK) Switching Ready state flag)

This flag indicates the state of switching ready for the CANFDCLK. When CANFDCKSRDY = 1, no clock is output to CANFDCLK.

8.2.44 USB60CKCR : USB60 Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x077

Bit position:	7	6	5	4	3	2	1	0
Bit field:	USB60CKSRDY	USB60CKSREQ	—	—	USB60CKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	USB60CKSEL[3:0]	USB60 Clock (USB60CK) Source Select 0 0 0 0: HOCO 0 0 0 1: MOCO (value after reset) 0 0 1 1: Main clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Others: Setting prohibited	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	USB60CKSREQ	USB60 Clock (USB60CK) Switching Request 0: No request 1: Request switching	R/W
7	USB60CKSRDY	USB60 Clock (USB60CK) Switching Ready state flag 0: Impossible to switch 1: Possible to switch	R

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The USB60CKCR register controls the USB60 Clock.

To change the set value of USB60CKDIVCR. USB60CKDIV[2:0] and USB60CKSEL[3:0], follow the procedure below.

When switching the clock source, it is necessary that the clock before switching and the clock after switching are stably output.

Clock selection switching procedure:

1. Write 1 to MSTPCRB.MSTPB12 (only when switching the division ratio setting from n (n ≠ 1)).
2. Wait for 2 USB60CLK (only when switching the division ratio setting from n (n ≠ 1)).
3. Write 1 to USB60CKSREQ.
4. Poll until USB60CKSRDYY is read as 1. While USB60CKSRDY = 1, no clock is output to USB60CLK.
5. Write to USB60CKDIVCR.USB60CKDIV[2:0] and USB60CKSEL[3:0].
6. Write 0 to USB60CKSREQ.
7. Poll until USB60CKSRDY is read as 0.
8. When USB60CKSRDY becomes 0, USB60CLK starts to output. Clock switching is complete.

When transitioning to Software Standby or Deep Software Standby mode, do not execute the WFI instruction while performing clock switching. That is, do not execute the WFI instruction when USB60CKSREQ = 1 and USB60CKSRDY = 0, or when USB60CKSREQ = 0 and USB60CKSRDY = 1.

USB60CKSEL[3:0] bits (USB60 Clock (USB60CK) Source Select)

These bits select the clock source of the USB Clock (USB60CLK) and must be modified when USB60CKCR.USB60CKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

USB60CKSREQ bit (USB60 Clock (USB60CK) Switching Request)

This bit selects the USB60CLK switching request.

USB60CKSRDY flag (USB60 Clock (USB60CK) Switching Ready state flag)

This flag indicates the state of switching ready for the USB60CLK. When USB60CKSRDY = 1, no clock is output to USB60CLK.

8.2.45 I3CCKCR : I3C Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x078

Bit position:	7	6	5	4	3	2	1	0
Bit field:	I3CCKSRDY	I3CCKSREQ	—	—	I3CCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	I3CCKSEL[3:0]	I3C clock (I3CCLK) Source Select 0001: MOCO (value after reset) 0101: PLL1P 0110: PLL2P 0111: PLL1Q 1000: PLL1R 1001: PLL2Q 1010: PLL2R Others: Setting prohibited	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	I3CCKSREQ	I3C clock (I3CCLK) Switching Request 0: No request 1: Request switching	R/W
7	I3CCKSRDY	I3C clock (I3CCLK) Switching Ready state flag 0: Impossible to switch 1: Possible to switch	R

Note: S-TYPE-3, P-TYPE-2

I3CCKCR controls the I3C clock.

To change the set value of I3CCKDIVCR.I3CCKDIV[2:0] and I3CCKSEL[3:0], follow the procedure below.

When switching the clock source, it is necessary that the clock before switching and the clock after switching are stably output.

Clock selection switching procedure:

1. Write 1 to MSTPCRB.MSTPB4 (only when switching the division ratio setting from n (n ≠ 1).
2. Wait for 2 I3CCLK (only when switching the division ratio setting from n (n ≠ 1)).
3. Write 1 to I3CCKSREQ.
4. Polling until I3CCKSRDY is read as 1. While I3CCKSRDY = 1, no clock is output to I3CCLK.
5. Write the setting value to I3CCKDIVCR.I3CCKDIV[2:0] and I3CCKSEL[3:0].
6. Write 0 to I3CCKSREQ.
7. Polling until I3CCKSRDY is read as 0.
8. When I3CCKSRDY becomes 0, I3CCLK starts outputting. Clock switching complete.

When a transition to Software Standby or Deep Software Standby, do not execute the WFI instruction while performing clock selection switching. In other words, do not execute the WFI instruction with I3CCKSREQ = 1 and I3CCKSRDY = 0, or I3CCKSREQ = 0 and I3CCKSRDY = 1.

I3CCKSEL[3:0] bits (I3C clock (I3CCLK) Source Select)

These bits select the clock source of the I3C clock (I3CCLK) and must be modified when I3CCKCR.I3CCKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

I3CCKSREQ bit (I3C clock (I3CCLK) Switching Request)

This bit selects the I3CCLK switching request.

I3CCKSRDY bit (I3C clock (I3CCLK) Switching Ready state flag)

This flag indicates the state of switching ready for the I3CCLK.

While I3CCKSRDY = 1, no clock is output to I3CCLK.

8.2.46 SCICKDIVCR : SCI Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x054

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SCICKDIV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SCICKDIV[2:0]	SCI clock (SCICLK) Division Select 0 0 0: 1/1 (value after reset) 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/6 1 0 0: 1/8 1 0 1: 1/3 1 1 0: 1/5 Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

SCICKDIVCR controls the SCI clock.

SCICKDIV[2:0] bits (SCI clock (SCICLK) Division Select)

These bits select the frequency of the SCI clock (SCICLK) and must be modified when SCICKCR.CKSRDY = 1.

When switching the division ratio setting from n (n ≠ 1), set MSTPCRB.MSTPBi (i = 22, 27, 28, 29, 30, 31) to 1 before setting SCICKCR.SCICKSREQ = 1.

8.2.47 SPICKDIVCR : SPI Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x056

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SPICKDIV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SPICKDIV[2:0]	SPI clock (SPICLK) Division Select 0 0 0: 1/1 (Value after reset) 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/6 1 0 0: 1/8 1 0 1: 1/3 1 1 0: 1/5 Others: Setting prohibited.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

SPICKDIVCR controls the SPI clock.

SPICKDIV[2:0] bits (SPI clock (SPICLK) Division Select)

These bits select the frequency of the SPI clock (SPICLK) and must be modified when SPICKCR.CKSRDY = 1.

When switching the division ratio setting from n (n ≠ 1), set MSTPCRB.MSTPBi (i = 18, 19) = 1 before setting SPICKCR.SPICKSREQ = 1.

8.2.48 LCDCKDIVCR : LCD Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x05E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	LCDCKDIV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	LCDCKDIV[2:0]	LCD clock (LCDCLK) Division Select 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/6 1 0 0: 1/8 1 0 1: 1/3 1 1 0: 1/5 Others: Setting prohibited.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

LCDCKDIVCR controls the LCD clock.

LCDCKDIV[2:0] bits (LCD clock (LCDCLK) Division Select)

These bits select the frequency of the LCD clock (LCDCLK) and must be modified when LCDCKCR.CKSRDY = 1.

When switching the division ratio setting from n (n ≠ 1), set MSTPCRC.MSTPC4 = 1 before setting LCDCKCR.LCDCKSREQ = 1.

8.2.49 SCICKCR : SCI Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x055

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SCICK SRDY	SCICK SREQ	—	—	SCICKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	SCICKSEL[3:0]0	SCI clock (SCICLK) Source Select 0000: HOCO 0001: MOCO (value after reset) 0010: LOCO 0011: Main clock oscillator 0100: Sub-clock oscillator 0101: PLL1P 0110: PLL2P 0111: PLL1Q 1000: PLL1R 1001: PLL2Q 1010: PLL2R Others: Setting prohibited	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	SCICKSREQ	SCI clock (SCICLK) Switching Request 0: No request 1: Request switching	R/W
7	SCICKSRDY	SCI clock (SCICLK) Switching Ready state flag 0: Impossible to switch 1: Possible to switch	R

Note: S-TYPE-3, P-TYPE-2

SCICKCR controls the SCI clock.

To change the set value of SCICKDIVCR.SCICKDIV[2:0] and SCICKSEL[3:0], follow the procedure below.

When switching the clock source, it is necessary that the clock before switching and the clock after switching are stably output.

Clock selection switching procedure:

1. Write 1 to MSTPCRB.MSTPBi (i = 22, 27, 28, 29, 30, 31) (only when switching the division ratio setting from n (n ≠ 1)).
2. Wait for 2 SCICLK (only when switching the division ratio setting from n (n ≠ 1)).
3. Write 1 to SCICKSREQ.
4. Polling until SCICKSRDY is read as 1. While SCICKSRDY = 1, no clock is output to SCICLK.
5. Write the setting value to SCICKDIVCR.SCICKDIV[2:0] and SCICKSEL[3:0].
6. Write 0 to SCICKSREQ.
7. Polling until SCICKSRDY is read as 0.
8. When SCICKSRDY becomes 0, SCICLK starts outputting. Clock switching complete.

When a transition to Software Standby or Deep Software Standby, do not execute the WFI instruction while performing clock selection switching. In other words, do not execute the WFI instruction with SCICKSREQ = 1 and SCICKSRDY = 0, or SCICKSREQ = 0 and SCICKSRDY = 1.

SCICKSEL[3:0] bits (SCI clock (SCICLK) Source Select)

These bits select the clock source of the SCI clock (SCICLK) and must be modified when SCICKCR.SCICKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

SCICKSREQ bit (SCI clock (SCICLK) Switching Request)

This bit selects the SCICLK switching request.

SCICKSRDY bit (SCI clock (SCICLK) Switching Ready state flag)

This flag indicates the state of switching ready for the SCICLK.

While SCICKSRDY = 1, no clock is output to SCICLK.

8.2.50 SPICKCR : SPI Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x057

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SPICKSRDY	SPICKSREQ	—	—	SPICKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	SPICKSEL[3:0]	SPI clock (SPICLK) Source Select 0000: HOCO 0001: MOCO (value after reset) 0010: LOCO 0011: Main clock oscillator 0100: Sub-clock oscillator 0101: PLL1P 0110: PLL2P 0111: PLL1Q 1000: PLL1R 1001: PLL2Q 1010: PLL2R Others: Setting prohibited	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	SPICKSREQ	SPI clock (SPICLK) Switching Request 0: No request 1: Request switching	R/W
7	SPICKSRDY	SPI clock (SPICLK) Switching Ready state flag 0: Impossible to switch 1: Possible to switch	R

Note: S-TYPE-3, P-TYPE-2

SPICKCR controls the SPI clock.

To change the set value of SPICKDIVCR.SPICKDIV[2:0] and SPICKSEL[3:0], follow the procedure below.

When switching the clock source, it is necessary that the clock before switching and the clock after switching are stably output.

Clock selection switching procedure:

1. Write 1 to MSTPCRB.MSTPBi (i = 18, 19) (only when switching the division ratio setting from n (n ≠ 1)).
2. Wait for 2 SPICLK (only when switching the division ratio setting from n (n ≠ 1)).
3. Write 1 to SPICKSREQ.
4. Polling until SPICKSRDY is read as 1. While SPICKSRDY = 1, no clock is output to SPICLK.
5. Write the setting value to SPICKDIVCR.SPICKDIV[2:0] and SPICKSEL[3:0].
6. Write 0 to SPICKSREQ.
7. Polling until SPICKSRDY is read as 0.
8. When SPICKSRDY becomes 0, SPICLK starts outputting. Clock switching complete.

When a transition to Software Standby or Deep Software Standby, do not execute the WFI instruction while performing clock selection switching. In other words, do not execute the WFI instruction with SPICKSREQ = 1 and SPICKSRDY = 0, or SPICKSREQ = 0 and SPICKSRDY = 1.

SPICKSEL[3:0] bits (SPI clock (SPICLK) Source Select)

These bits select the clock source of the SPI clock (SPICLK) and must be modified when SPICKCR.SPICKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

SPICKSREQ bit (SPI clock (SPICLK) Switching Request)

This bit selects the SPICLK switching request.

SPICKSRDY bit (SPI clock (SPICLK) Switching Ready state flag)

This flag indicates the state of switching ready for the SPICLK.

While SPICKSRDY = 1, no clock is output to SPICLK.

8.2.51 LCDCKCR : LCD Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x05F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LCDCKSRDY	LCDCKSREQ	—	—	LCDCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	LCDCKSEL[3:0]	LCD clock (LCDCLK) Source Select 0001: MOCO (Value after reset) 0101: PLL1P 0110: PLL2P 0111: PLL1Q 1000: PLL1R 1001: PLL2Q 1010: PLL2R Others: Setting prohibited.	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	LCDCKSREQ	LCD clock (LCDCLK) Switching Request 0: No request 1: Request switching	R/W
7	LCDCKSRDY	LCD clock (LCDCLK) Switching Ready state flag 0: Impossible to Switch 1: Possible to Switch	R

Note: S-TYPE-3, P-TYPE-2

LCDCKCR controls the LCD clock.

To change the set value of LCDCKDIVCR.LCDCKDIV[2:0] and LCDCKSEL[3:0], follow the procedure below.

When switching the clock source, it is necessary that the clock before switching and the clock after switching are stably output.

Clock selection switching procedure:

1. Write 1 to MSTPCRC.MSTPC4 (only when switching the division ratio setting from n (n ≠ 1)).
2. Wait for 2 LCDCLK (only when switching the division ratio setting from n (n ≠ 1)).
3. Write 1 to LCDCKSREQ.
4. Polling until LCDCKSRDY is read as 1. While LCDCKSRDY = 1, no clock is output to LCDCLK.
5. Write the setting value to LCDCKDIVCR.LCDCKDIV[2:0] and LCDCKSEL[3:0].
6. Write 0 to LCDCKSREQ.
7. Polling until LCDCKSRDY is read as 0.
8. When LCDCKSRDY becomes 0, LCDCLK starts outputting. Clock switching complete.

When a transition to Software Standby or Deep Software Standby, do not execute the WFI instruction while performing clock selection switching. In other words, do not execute the WFI instruction with LCDCKSREQ = 1 and LCDCKSRDY = 0, or LCDCKSREQ = 0 and LCDCKSRDY = 1.

LCDCKSEL[3:0] bits (LCD clock (LCDCLK) Source Select)

These bits select the clock source of the LCD clock (LCDCLK) and must be modified when LCDCKCR.LCDCKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

LCDCKSREQ bit (LCD clock (LCDCLK) Switching Request)

This bit selects the LCDCLK switching request.

LCDCKSRDY bit (LCD clock (LCDCLK) Switching Ready state flag)

This flag indicates the state of switching ready for the LCDCLK.

While LCDCKSRDY = “1”, no clock is output to LCDCLK.

8.2.52 TRCKCR : Trace Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x03F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TRCKEN	—	—	TRCKSEL	TRCK[3:0]			
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
3:0	TRCK[3:0]	Trace Clock operating frequency select 0000: 1/1 0001: 1/2 0010: 1/4 (Value after reset) 0011: 1/8 0100: 1/16 0101: 1/32 0110: 1/64 0111: 1/128 1000: 1/256 1001: 1/3 1010: 1/6 1011: 1/12 Others: Setting prohibited	R/W
4	TRCKSEL	Trace Clock source select 0: System clock source (Value after reset) 1: HOCO (oscillation in debug mode)	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	TRCKEN	Trace Clock operating Enable 0: Stop 1: Operation enable	R/W

Note: S-TYPE5, P-TYPE-2

Trace Clock Control Register controls switching the trace clock.

This register is not controlled by CGFSAR register.

When changing the frequency of the Trace Clock, the Trace Clock must be stopped (TRCKCR.TRCKEN = 0).

TRCK[3:0] bit (Trace Clock operating frequency select)

These bits select the frequency of the Trace Clock.

This bit must change when TRCKCR.TRCKEN = 0.

TRCKSEL bit (Trace Clock source select)

This bit selects the clock source of the Trace Clock. According to this bit, the Trace Clock is as follows:

[The Case of this bit is set 0]

- The system clock source is selected as the clock source for the trace clock.
  The system clock source is selected by SCKSCR.CKSEL[2:0] bits.
- The clock supply to the trace clock is selected of MOCO divided by 1 when in Software Standby mode or Deep Software Standby mode.

[The Case of this bit is set 1]

- The HOCO is selected as the clock source for the trace clock
- If TRCKSEL is set to 1, HOCO is operated no depending on the state of HOCOCR.HCSTP. However, OSCSF does not confirm that the swing is stable, so wait for the user to wait for the stable wait time
- HOCO continues to oscillate and supply the Trace Clock even if in Software Standby mode or Deep Software Standby mode.

This bit must change when TRCKCR.TRCKEN = 0.

TRCKEN bit (Trace Clock operating Enable)

This bit enables or disables the clock supply of the Trace Clock.

TRCK[3:0] bits and TRCKSEL bit must be set before changing this bit from 0 (stop) to 1 (enable).

When changing this bit from 1 (enable) to 0 (stop), Trace function must be disable.

The trace clock is output with this bit set to 1 and under the following conditions.
CDBGPWRUPREQ = 1 & AL ≠ AL0 | DBGAUTH0.DBGEN0 = 1 & OFS1(_SEC).SWDBG = 0.

8.3 Main Clock Oscillator

To supply the clock signal to the main clock oscillator, use one of the following ways:

- Connect an oscillator
- Connect the input of an external clock signal.

8.3.1 Connecting a Crystal Resonator

Figure 8.2 shows an example of connecting a crystal resonator. A damping resistor (Rd) can be added, if required.

Because the resistor values vary according to the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If the manufacturer recommends using an external feedback resistor (Rf), insert an Rf between EXTAL and XTAL by following the instructions.

When connecting a resonator to supply the clock, the frequency of the resonator must be in the frequency range of the resonator for the main clock oscillator as described in Table 8.1.

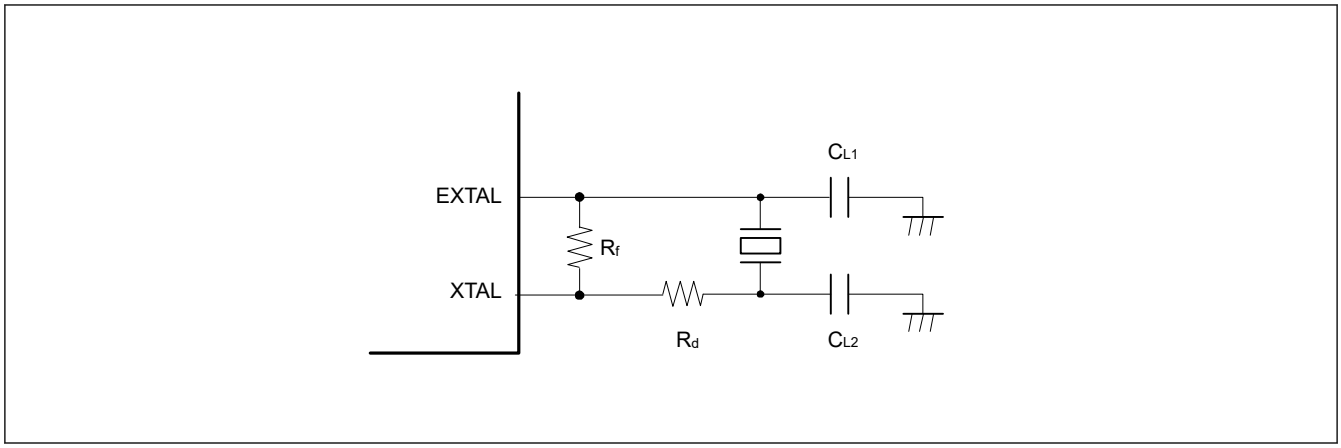


Figure 8.2 Example of crystal resonator connection

Figure 8.3 shows an equivalent circuit of the crystal resonator.

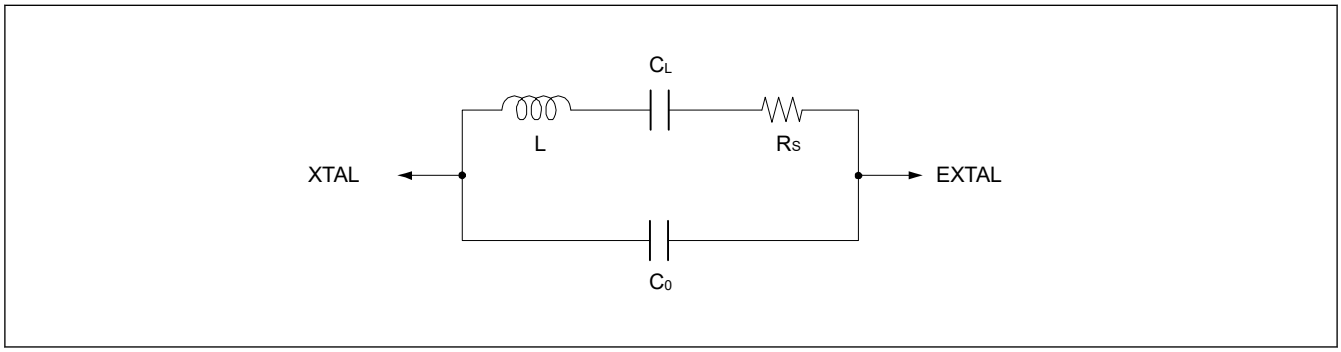


Figure 8.3 Equivalent circuit of the crystal resonator

8.3.2 External Clock Input

Figure 8.4 shows an example of connecting an external clock input. To operate the oscillator with an external clock signal, set the MOMCR.MOSEL bit to 1. The XTAL pin becomes high impedance.

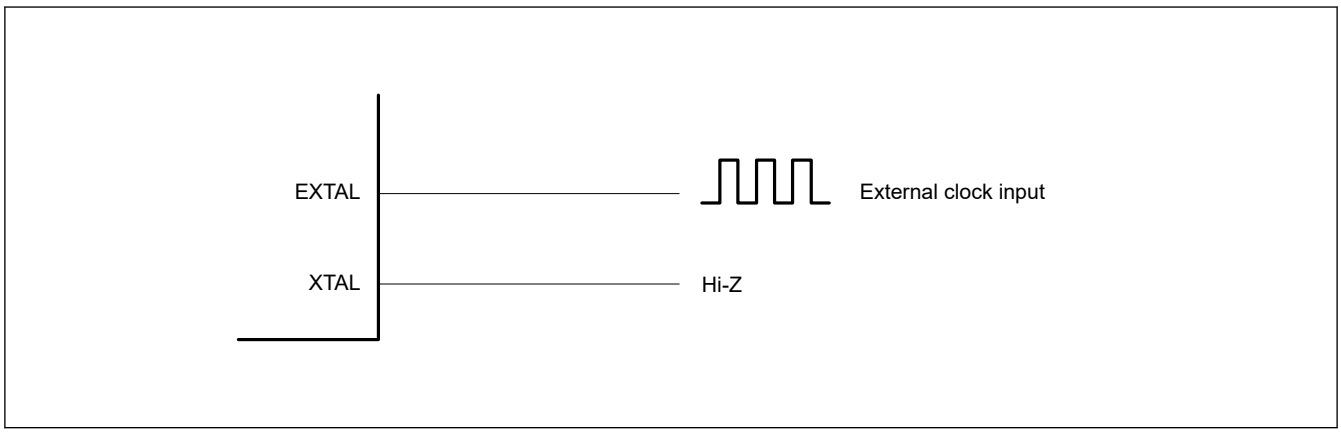


Figure 8.4 Equivalent circuit for external clock

8.3.3 Notes on External Clock Input

- The frequency of the external clock input can only be changed when the main clock oscillator is stopped. Do not change the frequency of the external clock input when the setting of the Main Clock Oscillator Stop bit (MOSCCR.MOSTP) is 0.
- To reduce current consumption during software standby mode, it is recommended to stop the external clock and keep it pull-up or pull-down.

However, if an external clock is used as the system clock source, it must be supplied while transitioning to and returning from software standby mode.
To do this safely, it is recommended to change the system clock source to another clock source before transitioning to software standby mode, or continue to use the external clock despite current consumption increase.

8.3.4 Note on Oscillation Keep in Software Standby mode

By using the oscillation keep function of the Main Clock Oscillator in Software Standby mode, the Main Clock Oscillator stabilization wait time after canceling Software standby mode can be shortened.

If MCU enters Software Standby mode under the MOSCSCR.MOSCSOKP bit is set to enabled (MOSCSCR.MOSCSOKP bit is 1) and under the Main Clock Oscillator is operating, the Main Clock Oscillator will keep oscillating in Software Standby mode. When canceling Software Standby mode in this state, the Main clock can be used without the oscillator stabilization wait time.

If using this function, confirm that the setting value is written to the register before entering the Software standby mode.

However, even if this function is enabled, the clock supply to the system and peripheral modules during Software Standby mode is stopped.

8.4 Sub-Clock Oscillator

To supply the clock signal to the sub-clock oscillator, use one of the following ways:

- Connect an oscillator
- Connect the input of an external clock signal.

8.4.1 Connecting a 32.768-kHz Crystal Resonator

To supply a clock to the sub-clock oscillator, connect a 32.768-kHz crystal resonator as shown in Figure 8.5. A damping resistor (Rd) can be added, if necessary. Because the resistor values vary according to the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If the resonator manufacturer recommends the use of an external feedback resistor (Rf), insert an Rf between XCIN and XCOUT by following the instructions. When connecting a resonator to supply the clock, the frequency of the resonator must be in the frequency range of the resonator for the sub-clock oscillator as described in Table 8.1.

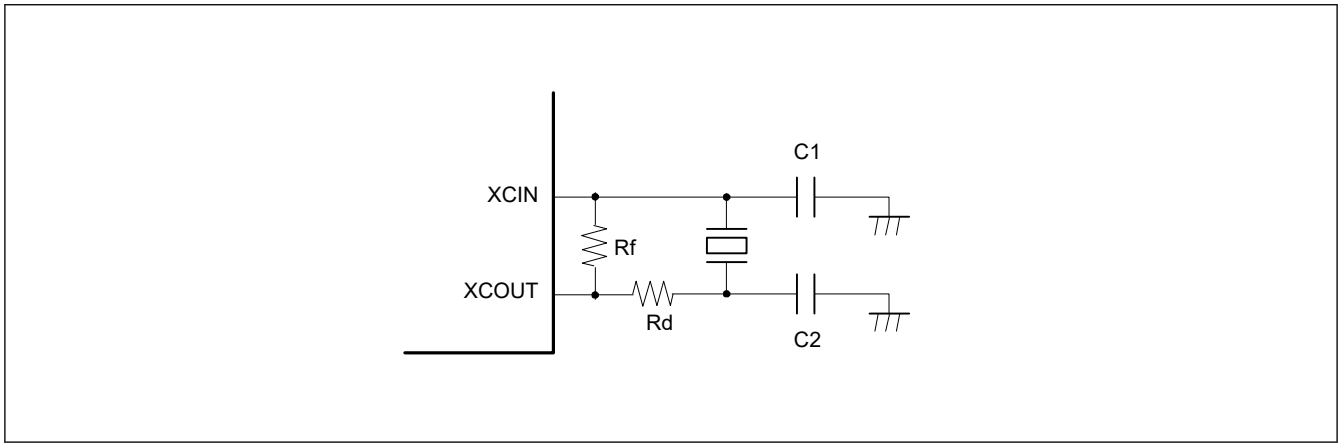


Figure 8.5 Connection example of 32.768-kHz crystal resonator

Figure 8.6 shows an equivalent circuit for the 32.768-kHz crystal resonator.

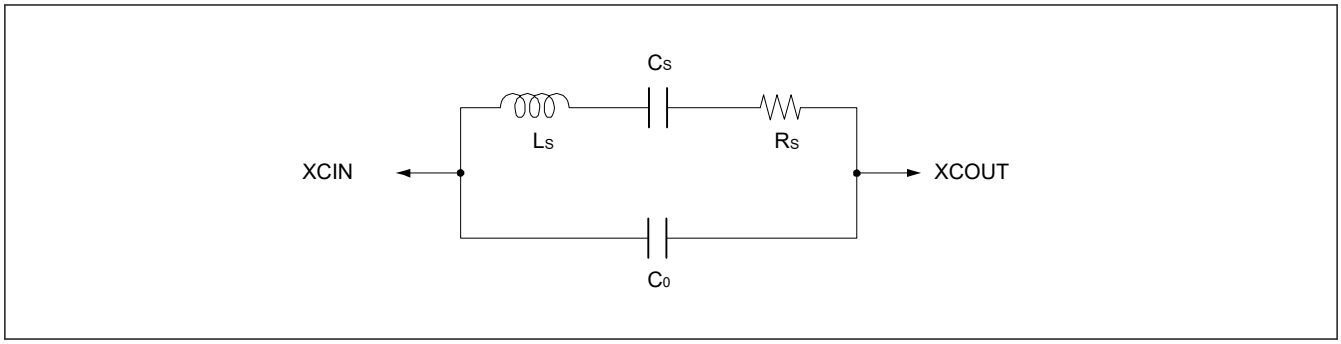


Figure 8.6 Equivalent circuit for the 32.768-kHz crystal resonator

8.4.2 Pin Handling When the Sub-Clock Oscillator Is Not Used

When the sub-clock oscillator is not in use, connect the XCIN pin to VSS through a resistor (to pull VSS down) and leave the XCOUT pin open as shown in Figure 8.7. In addition, if an oscillator is not connected, set the Sub-Clock Oscillator Stop bit (SOSCCR.SOSTP) to 1 to stop the oscillator.

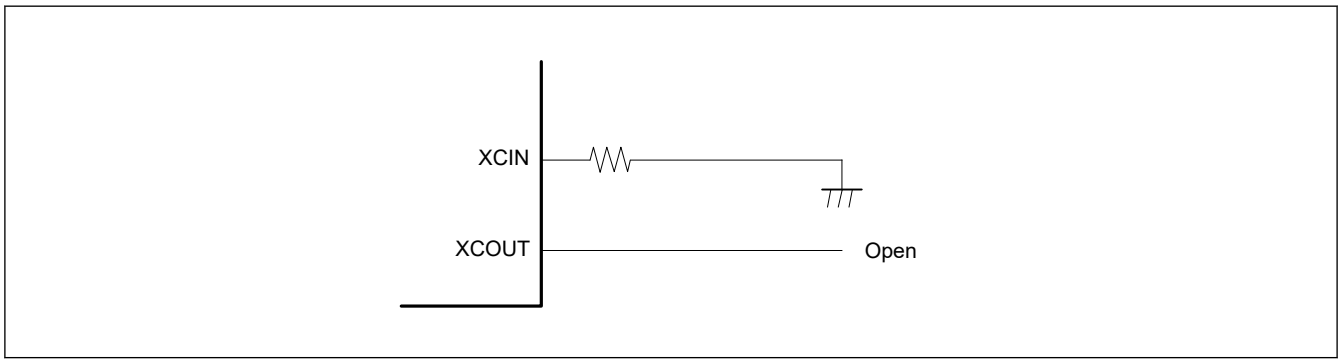


Figure 8.7 Pin handling when the sub-clock oscillator is not used

8.4.3 External Clock Input

In the external clock input mode, clock is input from EXCIN pin. In this case, set the XCIN/XCOUT pin as shown in as Figure 8.7.

8.5 Low speed On Chip Oscillator (LOCO)

Low speed On Chip Oscillator (LOCO) is the internal oscillator that supplies a low-speed clock.

8.5.1 LOCO Operating conditions

There are restriction on LOCO operation setting. For detail, see section 8.5.3. Restrictions on LOCO operation and stopping

LOCO oscillates when any of the following conditions is met.*1

[LOCO operating conditions]

- When LOCO is set to operate in the LOCOCR register (LOCOCR.LCSTP bit is 0).
- When IWDT is set to the auto start mode. (OFS0.IWDTSTRT bit is 0)*2
- When IWDT is set to the register start mode (OFS0.IWDTSTRT bit is 1) and counting is started.*3
- \When waiting for the oscillation stabilization of the Main Clock Oscillator (while MOCOCR.MCSTP bit is 0 and OSCSF.MOSCSF bit is 0).*4

Note 1. LOCO stops at Deep Software Standby mode 2 or Deep Software Standby mode 3.
Note 2. When IWDT is set to the auto start mode, regardless of the setting of IWDTSTPCTL bit in OFS0 register, LOCO continues to oscillate.
Note 3. When IWDT is set to the register start mode, regardless of the setting of SLCSTP bit in IWDTCSTPR register, LOCO continues to oscillate once IWDT starts counting.

Note 4. Even if the Main Clock Oscillator is set to the external clock input mode, LOCO operates because the oscillation stabilization wait circuit operates when the Main Clock Oscillator starts operating.

8.5.2 LOCO Stop conditions

There are restriction on LOCO stop setting. For detail, see section 8.5.3. Restrictions on LOCO operation and stopping.

There are two ways to top LOCO: stop by transition to Low Power mode or stop by LOCOCR register.

(A) LOCO stop conditions related to Low Power mode

LOCO stops when transitioning to the following Low Power mode

- Deep Software Standby mode 2
- Deep Software Standby mode 3

(B) LOCO stop conditions related to LOCOCR register

When LOCO is stopped using the LOCOCR register, even if the LCSTP bit set to stop (LOCOCR.LCSTP bit is 1), LOCO may not be stopped depending on the operating status of IWDT and oscillator.

LOCO can be stopped when all of the following conditions are met.

- LOCO is set to stop using the LOCOCR register (LOCOCR.LCSTP bit is 1)
- IWDT is not operating. (Include the case of stopping in CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode, and Deep Software Standby mode 1 by setting the OFS0.IWDTSTPCTL bit or the IWDTCSTPR.SLCSTP bit.).
- Main Clock Oscillator is stopped or Main Clock oscillation is stable. (MOCOCR.MCSTP bit is 1 or OSCSF.MOSCSF bit is 1)

8.5.3 Restrictions on LOCO operation and stopping

After setting the LCSTP bit in LOCOCR to 0 to start the LOCO clock, only use the clock after the LOCO clock oscillation stabilization wait time (tLOCOWT) elapses. A fixed stabilization wait is required after setting the LOCO clock to start operation. A fixed wait for the oscillation to stop is also required.

The following restrictions apply when starting and stopping operation:

- After stopping the LOCO clock, read OSCMONR.LOCOMON, make sure LOCO is stopped, and then set the operation.
- Confirm that LOCO oscillation is stable before stopping the LOCO clock
- Confirm that LOCO oscillation is stable before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode.
- When a transition to Software Standby mode or Deep Software Standby mode after setting LOCO to stop, read OSCMONR.LOCOMON, make sure LOCO is stopped, and executing the WFI instruction.

The LOCO clock is used to measure the oscillation stabilization wait time of the main clock oscillator. While the main clock oscillator is waiting for oscillation stabilization, the LOCO clock oscillates regardless of the LCSTP bit setting. At this time, it is necessary to prevent conflicts between the LOCO stop (or operation) by the LCSTP bit in LOCOCR and the LOCO clock oscillation for the Main clock Oscillator stabilization wait time (or LOCO clock stop after Main clock oscillation stabilization). Therefore, do not change the LOCOCR.LCSTP bit during the period from the 3 LOCO clock cycle before setting the start of main clock oscillator operation until the main clock oscillator stabilization can be confirmed with the oscillation stabilization flag in the OSCSF register.

When using IWDT in the register start mode (OFS0.IWDTSTRT bit is 1), set the LOCO to run (LOCOCR.LCSTP = 0) before IWDT count operation, and use IWDT after LOCO oscillation has stabilized. When IWDT is used in register start mode, do not change the LOCOCR.LCSTP bit during the period from the 3 LOCO clock cycles before IWDT count operation starts until the start of IWDT count operation can be confirmed.

When using the Digital filter or the Reset negate select for Voltage monitor m (m = 1 to 2) in Programmable Voltage Detection (PVDm), set the LOCO to run (LOCOCR.LCSTP = 0) before enabling PVDm and use PVDm after LOCO oscillation has stabilized.

8.6 Middle speed On Chip Oscillator (MOCO)

Middle speed On Chip Oscillator (MOCO) is the internal oscillator that supplies a middle-speed clock.

8.6.1 MOCO Operating conditions

There are restriction on MOCO operation setting. For detail, see section 8.6.3. Restrictions on MOCO operation and stopping.

MOCO oscillates when any of the following conditions is met.

[MOCO operating conditions]

- When MOCO is set to operate in the LOCOCR register (MOCOCR.MCSTP = 0)
- When waiting for the HOCO oscillation stabilization (while OFS1(_SEC).HOCOEN bit is 0 or HOCOCR.HCSTP bit is 0, and OSCSF.HOCOSF bit is 0).
- When waiting for the PLL1 oscillation stabilization (while PLLCR.PLLSTP bit is 0, and OSCSF.PLLSF bit is 0).
- When waiting for the PLL2 oscillation stabilization (while PLL2CR.PLL2STP bit is 0, and OSCSF.PLL2SF bit is 0).

Note: MOCO stops at Software Standby mode or Deep Software Standby mode.

8.6.2 MOCO Stop conditions

There are restriction on MOCO stop setting. For detail, see section 8.6.3. Restrictions on MOCO operation and stopping.

There are two ways to top MOCO: stop by transition to Low Power mode or stop by MOCOCR register.

(A) MOCO stop conditions related to Low Power mode

MOCO stops when transitioning to the following Low Power mode.

- Software Standby mode
- Deep Software Standby mode

Note: When the On-Chip Debugger function is enabled, MOCO does not stop when transitioning to each standby mode.

(B) MOCO stop conditions related to MOCOCR register

When MOCO is stopped using the MOCOCR register, even if the MCSTP bit set to stop (MOCOCR.MCSTP bit is 1), MOCO may not be stopped depending on the operating status of the system.

MOCO can be stopped when all of the following conditions are met.

[MOCO stop conditions]

- MOCO is set to stop using the MOCOCR register (MOCOCR.MCSTP bit is 1)
- When HOCO is stopped or HOCO oscillation is stable. (OFS1(_SEC).HOCOEN bit is 1 and HOCOCR.HCSTP bit is 1, or OSCSF.HOCOSF bit is 1)
- PLL1 circuit is stopped or PLL1 oscillation is stable. (PLLCR.PLSTP bit is 1 or OSCSF.PLLSF bit is 1)
- PLL2 circuit is stopped or PLL2 oscillation is stable. (PLL2CR.PL2STP bit is 1 or OSCSF.PLL2SF bit is 1)

8.6.3 Restrictions on MOCO operation and stopping

After setting the MCSTP bit to 0 to start the MOCO clock, only use the clock after the MOCO clock oscillation stabilization time (tMOCOWT) elapses. A fixed stabilization wait is required after setting the MOCO clock to start operation. A fixed wait for oscillation to stop is also required.

The following restrictions apply when starting and stopping operation.

- After stopping the MOCO clock, read OSCMONR.MOCOMON, make sure MOCO is stopped, and then set the operation.
- Confirm that MOCO oscillation is stable before stopping the MOCO clock.
- Regardless of whether the MOCO clock is selected as the system clock source, confirm that MOCO oscillation is stable before executing a WFI instruction to place the MCU in Software Standby or CPU Deep Software Standby mode. If MOCO is stopped, operate the MOCO.

- Regardless of whether the MOCO clock is selected as the system clock source, confirm that MOCO oscillation is stable before executing the wait instruction to place the CPU in the CPU Deep Sleep mode. If MOCO is stopped, operate the MOCO.
- Confirm that MOCO oscillation is stable before executing a Power Control for each power domain by Software. If MOCO is stopped, operate the MOCO.
- Confirm that MOCO oscillation is stable before programming or erasing Flash Memory. If MOCO is stopped, operate the MOCO.

Writing 1 to MCSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 001b (system clock source = MOCO).
- Oscillation stop detection is enabled (OSTDCR.OSTDE bit is 1).
- While waiting for the HOCO oscillation stabilization (OFS1(_SEC).HOCOEN bit is 0 or HOCOCR.HCSTP bit is 0, and OSCSF.HOCOSF bit is 0)
- While waiting for the PLL1 oscillation stabilization (PLLCR.PLLSTP bit is 0, and OSCSF.PLLSF bit is 0)
- While waiting for the PLL2 oscillation stabilization (PLL2CR.PLL2STP bit is 0, and OSCSF.PLL2SF bit is 0)
- While executing a Power Control for each power domain by software (During the Power Gating Control)
- While measuring the Operation Stability wait time for Flash Memory
- While Flash Memory programming or erasure

The MOCO clock is used to measure the oscillation stabilization wait time of the HOCO/PLL1/PLL2 clock. While waiting for the HOCO/PLL1/PLL2 clock to stabilize, the MOCO clock oscillates regardless of the MOCOCR.MCSTP bit setting. At this time, it is necessary to prevent conflicts between the MOCO stop (or operation) by the MCSTP bit in MOCOCR and the MOCO clock oscillation for HOCO/PLL1/PLL2 clock oscillation stabilization wait time (or MOCO clock stop after HOCO/PLL1/PLL2 clock oscillation stabilization). Therefore, do not change the MOCOCR.MCSTP bit during the period from the 3 MOCO clock cycles before the start of HOCO/PLL1/PLL2 clock operation setting until the HOCO/PLL1/PLL2 clock oscillation stability can be confirmed by the HOCOSF/PLLSF/PLL2SF flag in the OSCSF register.

8.7 High speed On Chip Oscillator (HOCO)

High-speed On Chip Oscillator (HOCO) is the internal oscillator that supplies a high-speed clock.

8.7.1 Frequency Locked Loop Function

HOCO supports the Frequency Locked Loop (FLL) function that corrects the frequency accuracy of the HOCO clock using the Sub-Clock Oscillator. When the FLL function is enabled, the frequency accuracy of the HOCO clock can be improved compared to when the FLL function is disabled.

When using the FLL function, the oscillation of the Sub-Clock Oscillator must be stable. In addition, the sub-clock oscillator must meet the frequency accuracy specified in the electrical characteristics. (For details, section 60, Electrical Characteristicssee section Electrical Characteristics)

When enabling the FLL function and using HOCO, enable the FLL function in advance and enable HOCO (HOCOCR.HCTSP=0) after SOSC has stabilized.

Set FLLCR2.FLLCNTL[10:0] as follows for each HOCO frequency setting. And the HOCO clock reaches the desired frequency after tFLLWT after enabling HOCO.

Table 8.5 FLLCNTL setting for each HOCO frequency setting

FLLCR2.FLLCNTL[10:0]	OFS1(_SEC).HOCOFRQ0[2:0]	Target Frequency (FHOCO)
0x1E9	000b	16 MHz
0x226	001b	18 MHz
0x263	010b	20 MHz
0x1E9	100b	32 MHz
0x1E9	111b	48 MHz

8.7.2 Note on Oscillation Keep in Software Standby mode

By using the oscillation keep function of the HOCO in Software Standby mode, the HOCO oscillation stabilization wait time after canceling Software standby mode can be shortened.

To keep the HOCO oscillation in Software Standby mode, make the following settings:

- Enable the HOCO Standby Oscillation Keep function in the HOCOSCR register. (HOCOSCR.HOCOSOKP bit is set to 1)
- The setting of power control for HOCO in HOCOLDOCR.

If MCU enters Software Standby mode under the above settings and under the HOCO is operating, the HOCO will keep oscillating in Software Standby mode. When canceling Software Standby mode in this state, the HOCO clock can be used without the oscillator stabilization wait time.

If using this function, confirm that the setting value is written to the register before entering the Software standby mode.

However, even if this function is enabled, the clock supply to the system and peripheral modules during Software Standby mode is stopped.

8.7.3 Note on using as TRACE clock source

If HOCO is selected as Trace clock source with TRCKCRSEL bit on TRCKCR register, and then Trace clock function is enable (TRCKCR.TRCKEN=1), HOCO oscillates regardless of the value of HCSTP bit on HOCOCR register. In this situation, HOCO will continue to oscillate even if MCU is in Software Standby mode, Deep Software Standby mode, or Reset (except POR and PVD0) to keep supplying the clock to the TRACE module.

If HOCO is used as Trace clock source, Trace clock should be enabled with TRCKEN bit on TRCKCR register while HOCO is stopped (OFS1(_SEC).HOCOEN=1 and HOCOCR.HCSTP=1) or HOCO is oscillating and stable (OSCSF.HOCOSF=1).

8.8 PLL Circuit

The PLL circuit has a function to multiply the frequency from the oscillator.

Figure 8.8 shows the block diagram of PLL1 circuit, and Figure 8.9 shows the PLL2 circuit. PLL1 and PLL2 have the same circuit configuration. The PLL1 circuit is explained the following as an example.

The input clock source selection circuit selects the clock source input to the PLL.

The input clock divider circuit divides the input clock and generates the PLL reference clock.

The frequency control circuit controls the VCO clock to be generated from the PLL reference clock according to the frequency multiplier setting. The frequency multiplication ratio is determined by the combination of the PLLMUL bits and PLLMULNF bits. The PLLMUL bit determines the integer part of the multiplication ratio, and the PLLMULNF bit determines the fractional part of the multiplication ratio. For example, if the PLLMUL bit setting is 26 and the PLLMULNF bit setting is 0.33, the multiplication ratio is 26.33.

The PLL circuit has the 3 output clocks. The output clock divider circuit divides the VCO clock to generate each PLL clock output. The division ratio for generating each output clock can be set independently with the PLODIVP, PLODIVQ, and PLODIVR bits.

The PLL Input clock frequency, the PLL Reference clock frequency, the VCO clock frequency, and each Output clock frequency must be within the specified in section 60, Electrical Characteristics.

The PLL frequency setting can only be changed the PLL circuit is stopped. Do not change the PLL frequency setting when PLL stop control bit (PLLSTP) is 0.

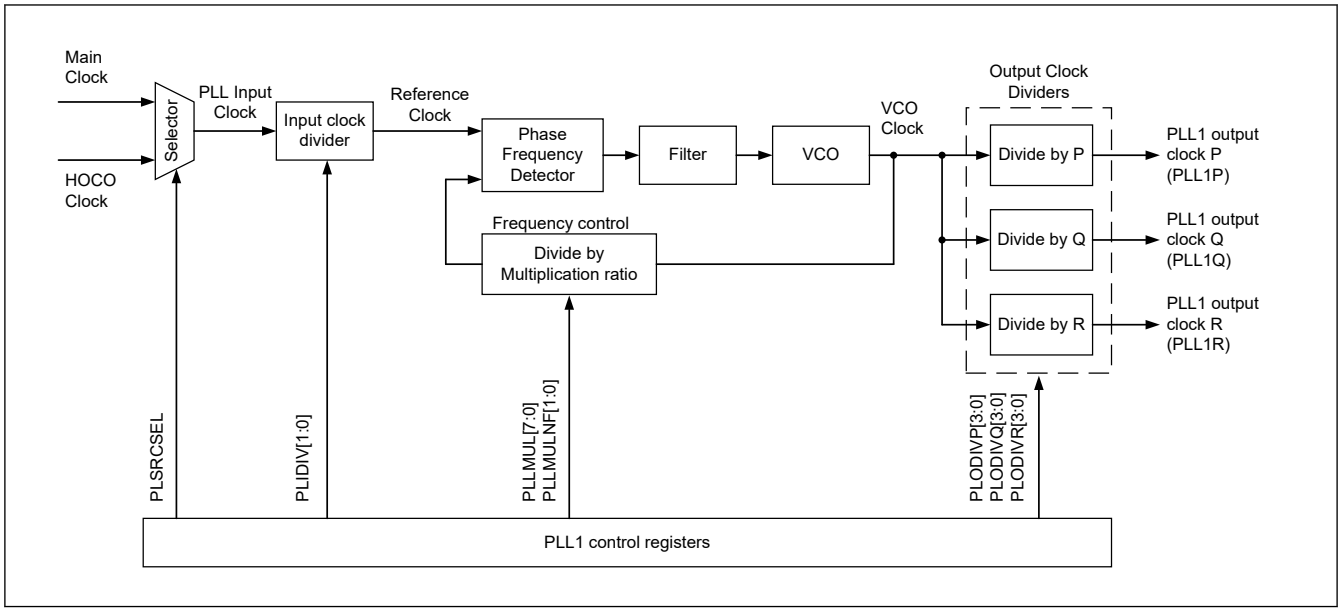


Figure 8.8 PLL1 block diagram

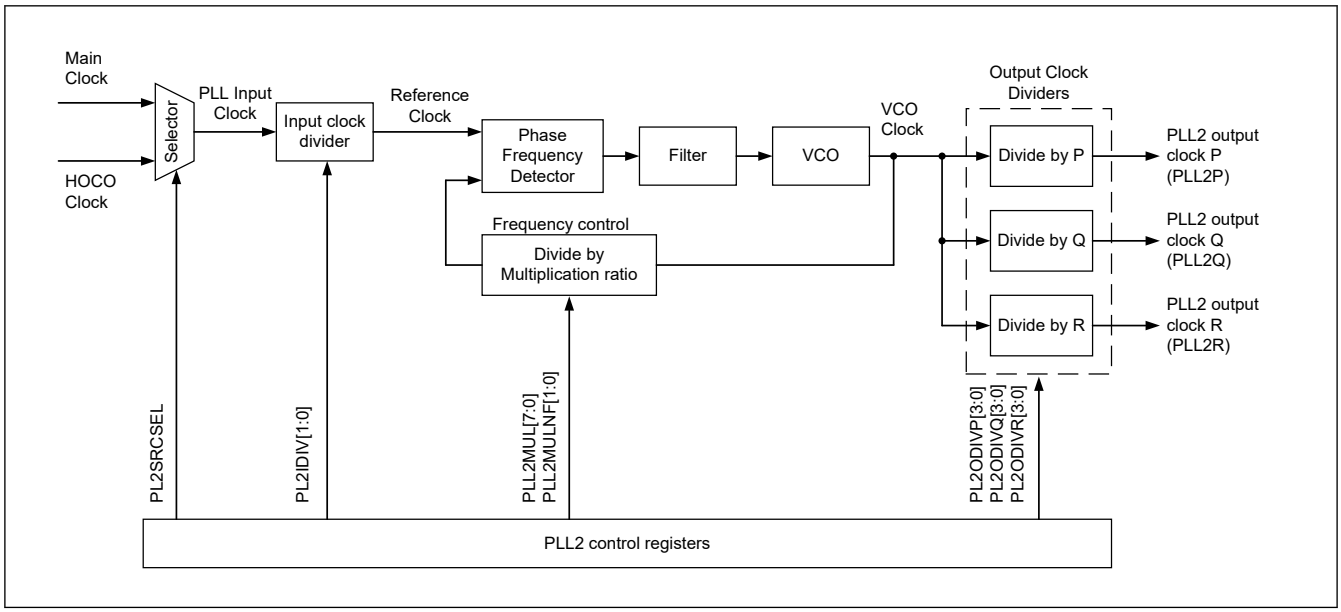


Figure 8.9 PLL2 block diagram

8.9 Oscillation Stop Detection Function

8.9.1 Oscillation Stop Detection and Operation after Detection

The oscillation stop detection function detects the main clock oscillator stop. When oscillation stop is detected, the system clock source switches as follows:

- If an oscillation stop is detected with SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC), the system clock source switches to the MOCO clock.
- If an oscillation stop is detected with PLLCCR.PLSRCSEL = 0 (PLL1 source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL1P), PLL1 clock remains the system clock source. The frequency becomes free-running, and the setting in the SCKSCR.CKSEL[2:0] bits does not change.

An oscillation stop detection interrupt request can be generated when an oscillation stop is detected. In addition, the General PWM Timer (GPT) output can be forced to stop state on detection.

The main clock oscillation stop is detected when the input clock remains at 0 or 1 for a certain period, for example, when a malfunction occurs in the main clock oscillator. See section 60, Electrical Characteristics.

Switching between the main clock oscillator and the MOCO clock or between the PLL1 clock and PLL1 free-running clock is controlled by the Oscillation Stop Detection Flag (OSTDSR.OSTDF).

OSTDF controls the switched clock as follows:

- When SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC):
  - When OSTDF changes from 0 to 1, the clock source switches to the MOCO clock.
  - When OSTDF changes from 1 to 0, the clock source switches back to MOSC.
- When PLLCCR.PLSRCSEL = 0 (PLL1 source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (System clock source = PLL1P):
  - When OSTDF changes 0 to 1, the clock source switches to the PLL1 free-running oscillation clock.
  - When OSTDF changes 1 to 0, the clock source switches back to PLL1.

To switch the clock source to the main clock or PLL1 clock again after the oscillation stop detection, set the CKSEL[2:0] bits to a clock source other than the main clock or PLL1 clock and clear the OSTDF flag to 0. Also, check that the OSTDF flag is not 1, then set the CKSEL[2:0] bits to the main clock or PLL1 clock after the specified oscillation stabilization time elapses.

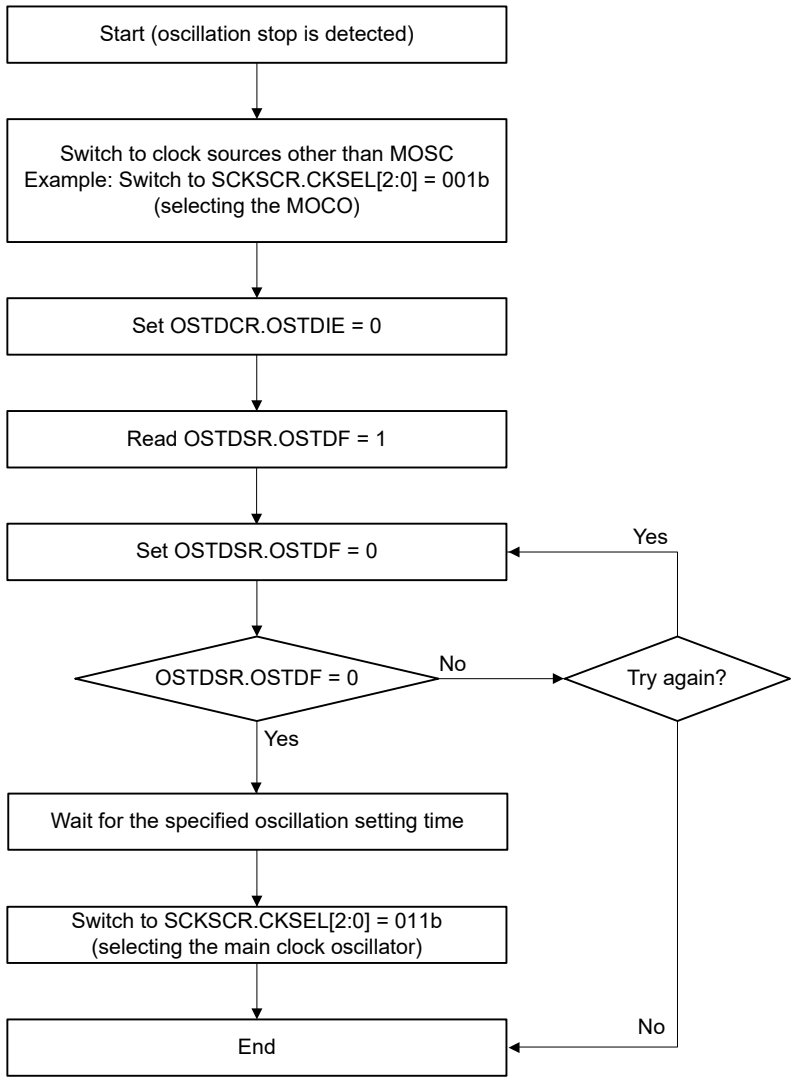
After a reset release, the main clock oscillator is stopped and the oscillation stop detection function is disabled. To enable the oscillation stop detection function, activate the main clock oscillator and write 1 to the oscillation stop detection function enable bit (OSTDCR.OSTDE) after a specified oscillation stabilization time elapses.

The oscillation stop detection function detects when the main clock is stopped by an external cause.

The oscillation stop detection function switches the following clocks to the MOCO clock (when the system clock source is MOSC) or the PLL1 free-running clock (when the system clock source is PLL1P) :

- All clocks that can be selected as the MOSC clock or PLL1 except CLKOUT, USBMCLK, MIPIMCLK and CACMCLK.
- The system clock (ICLK) frequency during the MOCO (when the system clock source is MOSC) or PLL1 free-running (when the system clock source is PLL1P) operation is specified by the MOCO oscillation frequency and the division ratio set by the system clock select bits (SCKDIVCR.ICK[3:0])

Example of returning when CKSEL[2:0] = 011b (selecting the main clock oscillator) after an oscillation stop is detected



Note: On return from the oscillation-stopped state, the factor responsible for stopping the main clock oscillation circuit must be removed from the system to allow oscillation to resume.

Figure 8.10 Flow of recovery on detection of oscillator stop

8.9.2 Oscillation Stop Detection Interrupts

An oscillation stop detection interrupt (MOSC_STOP) is generated when the Oscillation Stop Detection Flag (OSTDSR.OSTDF) is 1 and the Oscillation Stop Detection Interrupt Enable bit in the Oscillation Stop Detection Control Register (OSTDCR.OSTDIE) is 1 (enabled). The Port Output Enable for GPT (POEG) is notified of the main clock oscillator stop. On receiving the notification, the POEG sets the Oscillation Stop Detection Flag in the POEG Group n Setting Register (POEGGn.OSTPF) to 1 (n = A, B, C, D).

After the oscillation stop is detected, wait at least 10 PCLKB clock cycles before writing to the POEGGn.OSTPF flag. When the OSTDSR.OSTDF flag requires clearing, do so after clearing the Oscillation Stop Detection Interrupt Enable bit in the Oscillation Stop Detection Control Register (OSTDCR.OSTDIE). Wait at least 2 PCLKB clock cycles before setting the OSTDCR.OSTDIE bit to 1 again. A longer PCLKB wait time might be required, depending on the number of cycles required to read a given I/O register.

The oscillation stop detection interrupt is a non-maskable interrupt. Because non-maskable interrupts are disabled in the initial state after a reset release, enable non-maskable interrupts through software before using oscillation stop detection interrupts. For details, see section 13, Interrupt Controller Unit (ICU).

8.10 Internal Clock

Clock sources for the internal clock signals include:

- Main clock oscillator
- Sub-clock oscillator
- HOCO clock
- MOCO clock
- LOCO clock
- PLL1 clock (PLL1P, PLL1Q and PLL1R)
- PLL2 clock (PLL2P, PLL2Q and PLL2R)
- External clock input for JTAG
- External clock input for SWD

The following internal clocks are produced from these sources.

- Operating clock of the CPU : CPU clock (CPUCLK)
- Operating clock of the DMAC, DTC, Flash, SRAM, System Bus, I/O Port and ICU: System clock (ICLK)
- Operating clock of the Debug Subsystem : Debug clock (DCLK)
- Operating clocks of peripheral modules: Peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD, and PCLKE)
- Operating clock of the FlashIF: FlashIF clock (FCLK)
- Clock for the external bus controller and external pin output: External bus clock (BCLK, EBCLK)
- Clock for the external bus controller and external pin output for the SDRAM : SDRAM clock (SDCLK)
- Clock for the trace function and external pin output : Trace clock (TRCLK)
- Operating clock for the SCI : SCI clock (SCICLK)
- Operating clock for the SPI : SPI clock (SPICLK)
- Operating clock for the Octal SPI : Octal-SPI clock (OCTACLK, OCTADIVCLK)
- Operating clock for the CANFD Core : CANFD Core clock (CANFDCLK)
- Operating clock for the USBFS and USBHS : USB clock (USBCLK)
- Operating clock for the USBHS : USB clock (USB60CLK)
- Operating clock for the LCD : LCD clock (LCDCLK)
- Operating clock for the I3C : I3C clock (I3CCLK)
- Clock for external pin output : Clock/Buzzer output (CLKOUT)
- Operating clock for the USBHS : MOSC clock for USBHS (USBMCLK)
- Operating clock for the MIPI : MIPI clock (MIPIMCLK)
- Operating clock for the CAN : CAN clock (CANMCLK)
- Operating clocks for the ULPT : ULPT LOCO clock (ULPTLCLK) and ULPT sub-clock (ULPTSCLK)
- Operating clock for the AGT : AGT LOCO clock (AGTLCLK) and AGT sub-clock (AGTSCLK)
- Operating clocks for the CAC : CAC clock (CACCLK)
- Operating clocks for the RTC : RTC LOCO clock (RTCLCLK) and RTC Sub-clock (RTCSCLK)
- Operating clock for the IWDT : IWDT clock (IWDTCLK)
- Operating clock for the SysTick timer : SysTick timer clock (SYSTICKCLK)
- Operating clock for the JTAG : JTAG clock (JTAGTCK)

For details on the registers used to set the frequencies of the internal clocks, see section 8.10.2. System Clock (ICLK) to section 8.10.27. JTAG Clock (JTAGTCK)

If the value of any of these bits is changed, subsequent operation is at the frequency determined by the new value.

8.10.1 CPU Clock (CPUCLK)

The CPU clock (CPUCLK) is the operating clock for the CPU. Specify the frequency in the following bits:

- CPUCK[3:0] bits in SCKDIVCR2
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0] bits in PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

8.10.2 System Clock (ICLK)

The system clock (ICLK) is the operating clock of the DMAC, DTC, Flash, SRAM, System Bus, I/O Port, and ICU. Specify the frequency in the following bits:

- ICK[3:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0] bits in PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

When the ICLK clock source is switched, the duration of the ICLK clock cycle becomes longer during the clock source transition period. See Figure 8.11 and Figure 8.12.

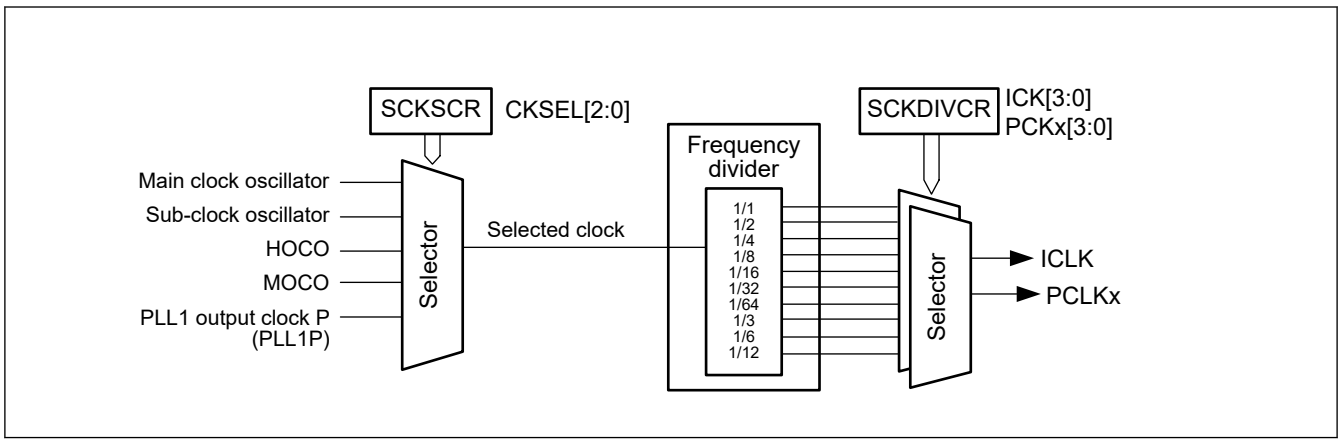


Figure 8.11 Block diagram of clock source selector

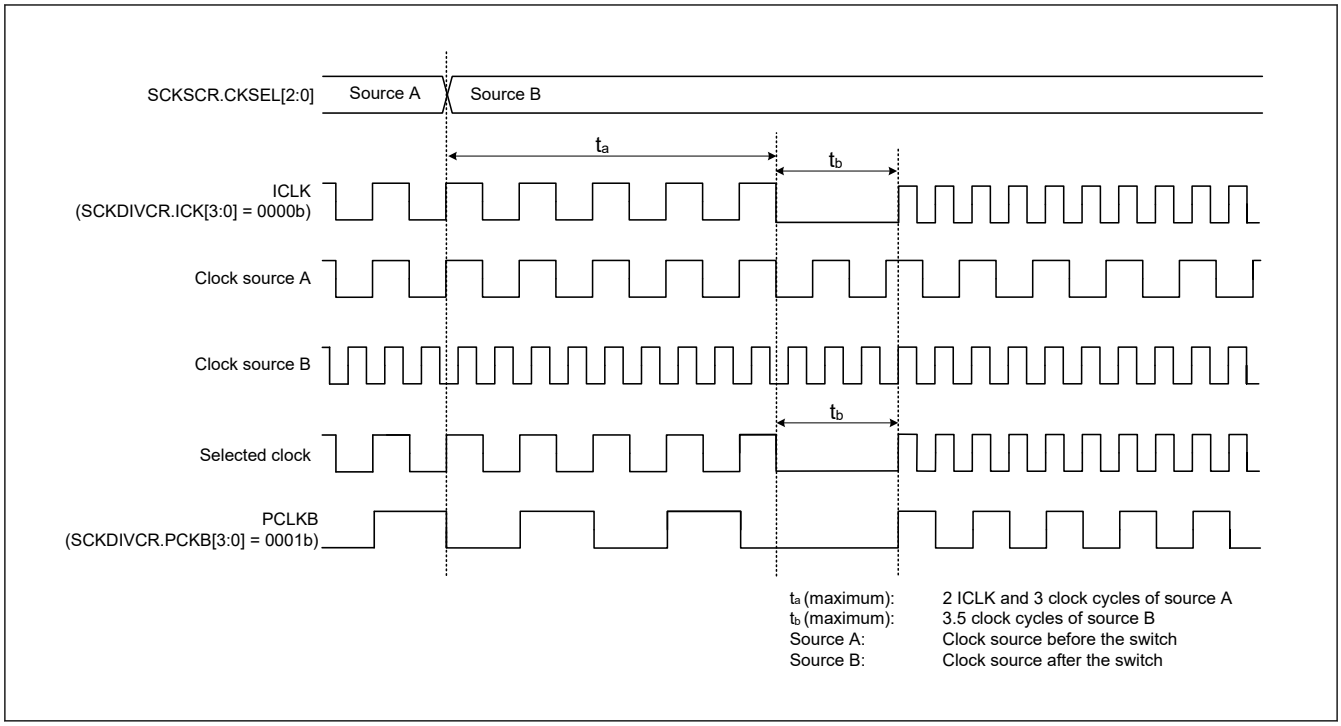


Figure 8.12 Timing of clock source switching

8.10.3 Debugger Clock (DCLK)

The Debugger clock (DCLK) is the operating clock for the Debug Subsystem. The frequency is ICLK divided by 2.

8.10.4 Peripheral Module Clock (PCLKA, PCLKB, PCLKC, PCLKD, PCLKE)

The peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD and PCLKE) are the operating clocks for the peripheral modules.

The frequency of the given clock is specified in the following bits:

- PCKA[3:0], PCKB[3:0], PCKC[3:0], PCKD[3:0] and PCKE[3:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0] bits in PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1.

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

When the clock source of the peripheral module clock is switched, the duration of the peripheral module clock cycle becomes longer during the clock source transition period. See Figure 8.11 and Figure 8.12.

8.10.5 FlashIF Clock (FCLK)

The flash interface clock (FCLK) is the operating clock for the flash memory interface. In addition to reading from the data flash, FCLK is used for the programming and erasure of the code flash and data flash.

The FCLK frequency is specified in the following bits:

- FCK[3:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0] bits in PLLCCR

- PLODIVP[3:0] bits in PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1.

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

8.10.6 External Bus Clock (BCLK, EBCLK)

The external bus clock (BCLK) is an operating clock for the external bus controller. It is also output externally from the EBCLK pin for the external connection bus.

BCLK can be output from the EBCLK pin by setting the EBCKOCR.EBCKOEN bit to 1 and setting the PmnPFS.PSEL[4:0] to 01011b. Make sure that modification of the PmnPFS.PSEL[4:0] to 01011b must always be performed while the EBCKOCR.EBCKOEN bit is 0.

When the BCKCR.BCLKDIV bit is set to 1, the BCLK clock divided by 2 is output from the EBCLK pin.

Specify the frequency in the following bits:

- BCK[3:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0] bits in PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1.

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

A frequency higher than the system clock (ICLK) should not be set for the BCLK.

8.10.7 SDRAM Clock (SDCLK)

The SDRAM clock (SDCLK) is an operating clock for the external bus controller. It is output externally from the SDCLK pin for the SDRAM that is connected to the external bus. To output SDCLK on the SDCLK pin, set the SDCKOCR.SDCKOEN bit to 1 and set the PmnPFS.PSEL[4:0] bits to 01011b(enabling SDCLK output). Only change the value in the PmnPFS.PSEL[4:0] bits when the SDCKOCR.SDCKOEN bit is 0. Specify the frequency in the following bits:

- BCK[3:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0] bits in PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1.

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

Do not set SDCLK to a frequency higher than that of the system clock (ICLK).

8.10.8 Trace Clock (TRCLK)

The trace clock (TRCLK) is the CPU tracing clock for the On-Chip Debugger function.

The TRCLK frequency is specified by the TRCK[3:0] bits and TRCKSEL bit in TRCKCR (The Trace Clock frequency division ratio and the Trace Clock Source select). When changing the frequency of the Trace Clock, the Trace Clock must be stopped (TRCKCR.TRCKEN = 0).

8.10.9 SCI Clock (SCICLK)

The SCI clock (SCICLK) is the operating clock for the SCI module.

Specify the frequency in the following bits:

- SCICKDIV[2:0] bits in SCICKDIVCR
- SCICKSEL[3:0] bits in SCICKCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[7:0], PLL2MULNF[1:0], PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1.

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

8.10.10 SPI Clock (SPICLK)

The SPI clock (SPICLK) is the operating clock for the SPI module.

Specify the frequency in the following bits:

- SPICKDIV[2:0] bits in SPICKDIVCR
- SPICKSEL[3:0] bits in SPICKCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[7:0], PLL2MULNF[1:0], PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1.

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

8.10.11 Octal-SPI clock (OCTACLK, OCTADIVCLK)

The Octal-SPI clock (OCTACLK) is the operating clock for the Octal-SPI module.

Specify the frequency in the following bits:

- OCTACKDIV[2:0] bits in OCTACKDIVCR
- OCTACKSEL[3:0] bits in OCTACKCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[7:0], PLL2MULNF[1:0], PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1.

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

OCTADIVCLK always outputs 2-division of OCTACLK.

8.10.12 CANFD Core clock (CANFDCLK)

The CANFD Core clock (CANFDCLK) is the operating clock for the CANFD module.

Specify the frequency in the following bits:

- CANFDCKDIV[2:0] bits in CANFDCKDIVCR
- CANFDCKSEL[3:0] bits in CANFDCKCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[7:0], PLL2MULNF[1:0], PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1.

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

8.10.13 USB Clock (USBCLK)

The USB clock (USBCLK) is the operating clock for the USBFS and USBHS module.

A 48-MHz clock must be supplied when using the USBFS module or when using the USBHS module in CL-Only mode. USBCLK does not need to be supplied when not using USBHS module in CL-Only mode.

The USBCLK frequency is specified in the following bits:

- USBCKDIV[2:0] bits in USBCKDIVCR
- USBCKSEL[3:0] bits in USBCKCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[7:0], PLL2MULNF[1:0], PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1.

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

8.10.14 USB Clock (USB60CLK)

The USB clock (USB60CLK) is the operating clock for the USBHS module. A 60-MHz clock must be supplied when using the USBHS module in CL-Only mode. USB60CLK does not need to be supplied when not using USBHS in CL-Only mode.

Specify the frequency in the following bits:

- USB60CKDIV[2:0] bits in USB60CKDIVCR
- USB60CKSEL[3:0] bits in USB60CKCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[7:0], PLL2MULNF[1:0], PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1.

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

8.10.15 MOSC clock for USBHS (USBMCLK)

The MOSC clock for USBHS (USBMCLK) is the operating clock for the USBHS when not using the module in CL-Only mode. USBMCLK does not need to be supplied when using USBHS in CL-Only mode.

The USBMCLK frequency is 12 MHz or 20 MHz or 24 MHz or 48 MHz supplied from the main clock oscillator.

8.10.16 LCD Clock (LCDCLK)

The LCD clock (LCDCLK) is the operating clock for the LCD module.

Specify the frequency in the following bits:

- LCDCKDIV[2:0] bits in LCDCLDIVCR
- LCDCKSEL[3:0] bits in LCDCKCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[7:0], PLL2MULNF[1:0], PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1.

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

8.10.17 I3C Clock (I3CCLK)

The I3C clock (I3CCLK) is the operating clock for the I3C module.

Specify the frequency in the following bits:

- I3CCKDIV[2:0] bits in I3CCKDIVCR
- I3CCKSEL[3:0] bits in I3CCKCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[7:0], PLL2MULNF[1:0], PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1.

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

8.10.18 CAN Clock (CANMCLK)

The CAN clock (CANMCLK) is the operating clock for the CAN module. CANMCLK is generated by the main clock oscillator.

8.10.19 MIPI Clock (MIPIMCLK)

The MIPI clock (MIPIMCLK) is the operating clock for the MIPI module. MIPIMCLK is generated by the main clock oscillator.

8.10.20 ULPT Clock (ULPTLCLK, ULPTSCLK)

The ULPT clocks (ULPTLCLK and ULPTSCLK) are the operating clock for the ULPT module. ULPTLCLK is generated by the LOCO clock and ULPTSCLK is generated by the Sub-clock oscillator.

8.10.21 CAC Clock (CACCLK)

The CAC clock (CACCLK) is the operating clock for the CAC. CACCLK includes the following clocks:

- CAC Main clock (CACMCLK) is generated by the Main clock oscillator
- CAC Sub oscillator clock (CACSCLK) is generated by the Sub-clock oscillator
- CAC HOCO clock (CACHCLK) is generated by the HOCO clock
- CAC MOCO clock (CACMOCLK) is generated by the MOCO clock
- CAC LOCO clock (CACLCLK) is generated by the LOCO clock

8.10.22 RTC Clock (RTCSCLK, RTCLCLK)

The RTC clock (RTCSCLK, RTCLCLK) is the operating clock for the RTC.

RTCSCLK is generated by the sub-clock oscillator, and RTCLCLK is generated by the LOCO clock.

8.10.23 IWDT Clock (IWDTCLK)

The IWDT clock (IWDTCLK) is the operating clock for the IWDT.

IWDTCLK is internally generated by the LOCO clock which is always divided by 2.

8.10.24 AGT Clock (AGTSCLK, AGTLCLK)

The AGT clocks (AGTSCLK and AGTLCLK) are the operating clocks for the AGT. AGTSCLK is generated by the sub-clock oscillator, and AGTLCLK is generated by the LOCO clock.

8.10.25 SysTick Timer Clock (SYSTICKCLK)

The SysTick timer clock (SYSTICKCLK) is the operating clock for the SysTick timer. SYSTICKCLK is generated by the MOCO clock which is always divided by 8.

8.10.26 External Pin Output Clock (CLKOUT)

The CLKOUT is output externally from the CLKOUT pin for the clock or buzzer output. The CLKOUT is output to the CLKOUT pin when the CKOCR.CKOEN bit is set to 1. Only change the value in the CKODIV[2:0] bits or CKOSEL[2:0] bits in CKOCR when the CKOCR.CKOEN bit is 0.

The CLKOUT clock frequency is specified in the following bits:

- CKODIV[2:0] bits or CKOSEL[2:0] bits in CKOCR
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

8.10.27 JTAG Clock (JTAGTCK)

The JTAG clock (JTAGTCK) is the clock for the JTAG.

JTAGTCK is generated by the JTAG external clock (TCK).

8.11 Clock Setting

8.11.1 System Clock Setting

8.11.1.1 Initial System Clock Setting

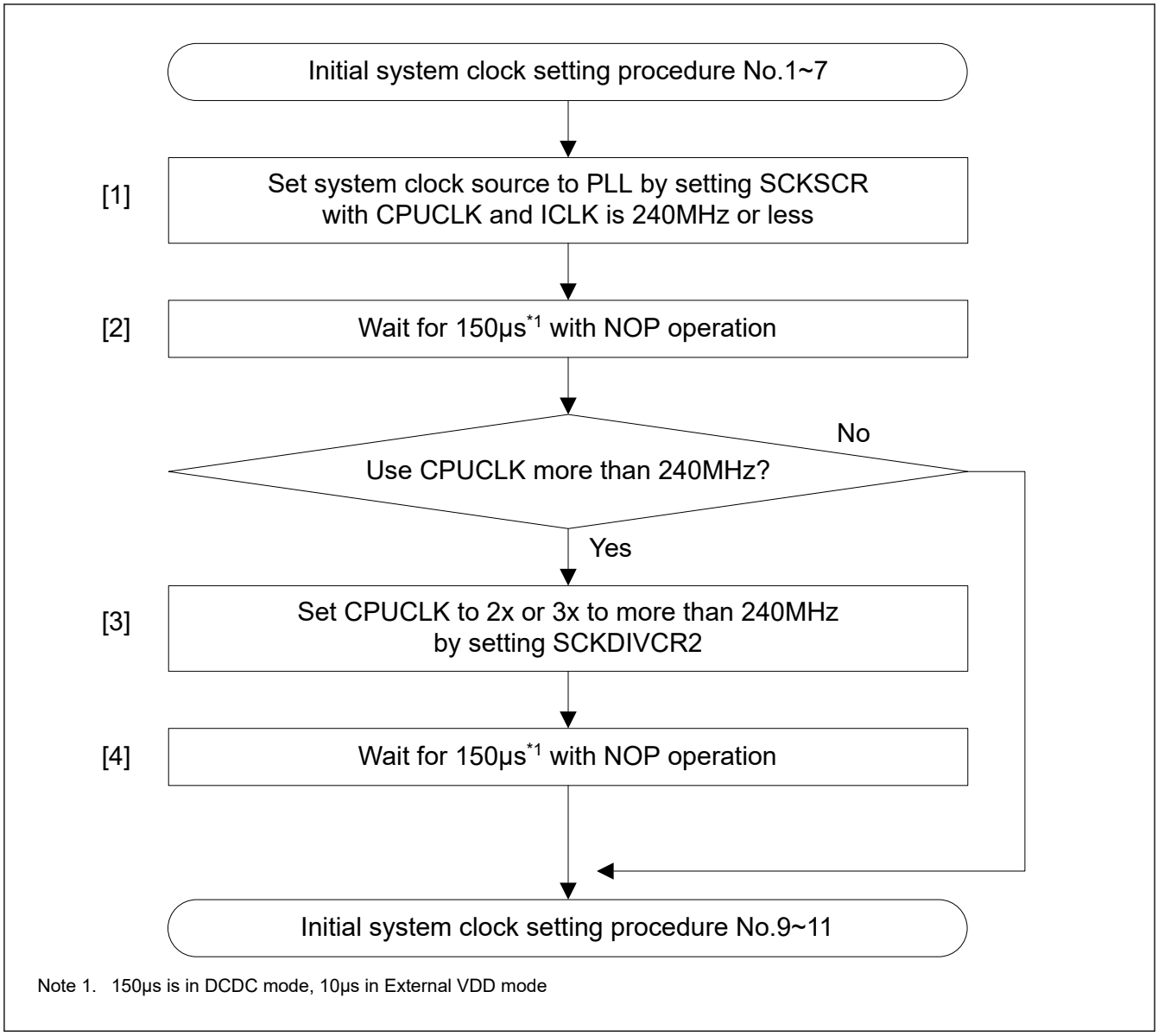
An example of the initial system clock setting procedure is shown in Table 8.6.

Table 8.6 Example of the initial system clock setting procedure after reset release / after Deep Software Standby cancellation

No.	Step	Description
1	Start	After reset release / Deep software standby cancellation (MOCO is selected as system clock source)
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 *1*2bit in PRCR register
3	Change Operating Power Control Mode to High-Speed mode*1	Change to High-Speed mode with OPCCR register. *1 For details, see section 10, Low Power Modes.
4	Set a clock source to oscillate	Set a clock source that switches to the system clock source to oscillate.
5	Set Flash Wait Cycle	Set an appropriate access wait count for the flash memory according to the frequency at which it operates after the system clock (ICLK) is set. For details, see section 52, Flash Memory.
6	Set internal clock division ratio	Set the internal clock division ratio with SCKDIVCR register.
7	Set External Bus clock output / SDRAM clock output*3	Set the EBCLK pin output setting with BCKCR. Set the SDCLK pin output setting with SDCKOCR.
8	Switch the System Clock source	Switch the system clock source with SCKSCR register. Before switching the system clock source, confirm that the clock source oscillation is stable.
9	Change Operating Power Control Mode	Change the Operating Power Control Mode with OPCCR register. *2
10	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 *1*2bit in PRCR register
11	End	System clock setting is completed.

Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.
Note 2. Setting is not necessary if the operating power control mode is not changed.
Note 3. Setting is not necessary if the External bus or SDRAM are not used.

When the PLL1 is selected as the system clock source, additional procedure is required for procedure No.8. This additional procedure is shown in Figure 8.13. It is recommended to use software to measure the wait time. Be sure to consider the worst-case conditions to ensure that the required wait time elapses. If an interrupt is unavoidably generated during the wait time, retry the measurement after return from the interrupt.



Note 1. 150µs is in DCDC mode, 10µs in External VDD mode

Figure 8.13 Additional flow of initial system clock setting when PLL is used for the system clock source. (assuming all module stop bis are initial value)

8.11.1.2 System Clock Setting for Faster Frequency

An example of setting procedure when changing the current system clock frequency to a faster frequency is shown in Table 8.7

Table 8.7 Example of Setting Procedure When changing the current system clock frequency to a Faster frequency (1 of 2)

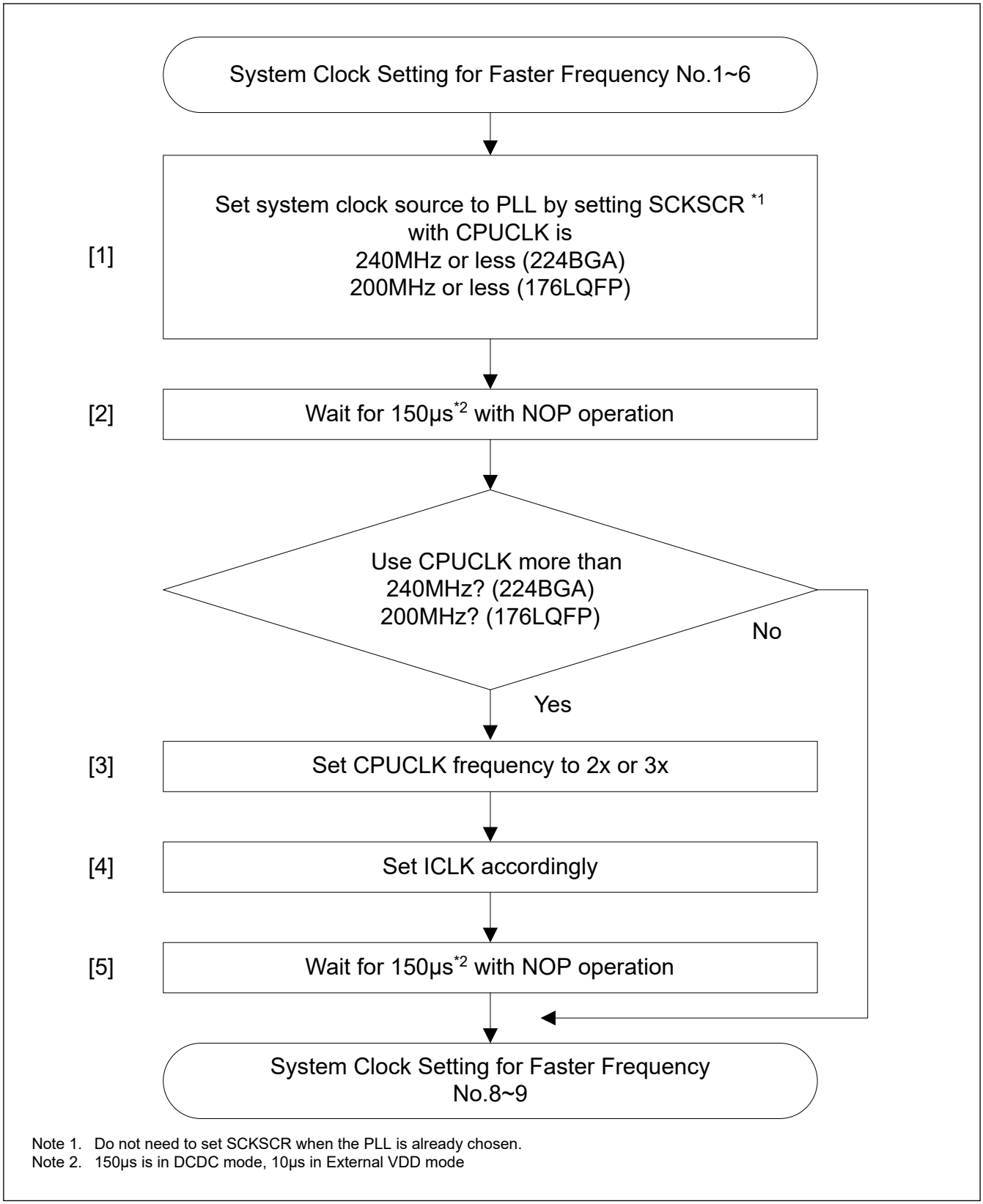
No.	Step	Description
1	Start	
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1[*1*2] bit in PRCR register
3	Change Operating Power Control Mode to High-Speed mode[*1]	Change to High-Speed mode with OPCCR register.[*1] For details, see section 10, Low Power Modes.
4	Set a clock source to oscillate[*3]	Set a clock source that switches to the system clock source to oscillate.[*4]
5	Set Flash Wait Cycle	Set an appropriate access wait count for the flash memory according to the frequency at which it operates after the system clock (ICLK) is set. For details, see section 52, Flash Memory.

Table 8.7 Example of Setting Procedure When changing the current system clock frequency to a Faster frequency (2 of 2)

No.	Step	Description
6	Set internal clock division ratio	Set the internal clock division ratio with SCKDIVCR register.
7	Switch the System Clock source*3	Switch the system clock source with SCKSCR register. Before switching the system clock source, confirm that the clock source oscillation is stable.
8	Change Operating Power Control Mode	Change the Operating Power Control Mode with OPCCR register. *2
10	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1*1*2 bit in PRCR register
11	End	System clock is changed to a Faster frequency.

Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.
Note 2. Setting is not necessary if the operating power control mode is not changed.
Note 3. Setting is not necessary if the system clock source is not changed.
Note 4. To change the frequency of the PLL selected as the system clock source, it is necessary to change the system clock source to another clock source, stop the PLL, and then change the PLL setting.

When the PLL1 is selected as the system clock source, additional procedure is required for procedure No.7. This additional procedure is shown in Figure 8.14. It is recommended to use software to measure the wait time. Be sure to consider the worst-case conditions to ensure that the required wait time elapses. If an interrupt is unavoidably generated during the wait time, retry the measurement after return from the interrupt



Note 1. Do not need to set SCKSCR when the PLL is already chosen.
Note 2. 150µs is in DCDC mode, 10µs in External VDD mode

Figure 8.14 Additional flow of system clock setting when PLL is used for the system clock source and CPUCLK to be faster frequency

8.11.1.3 System Clock Setting for Slower frequency

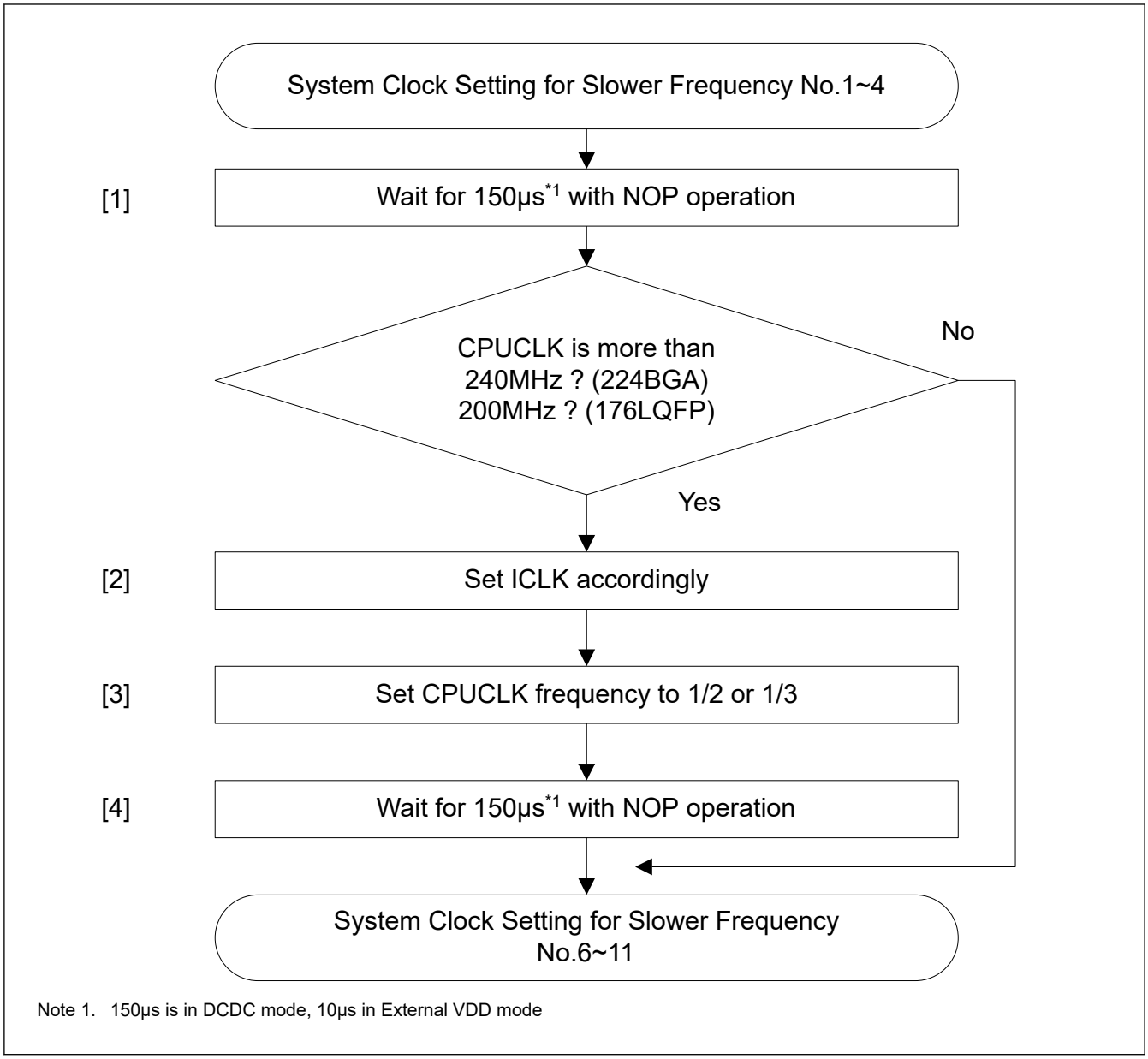
An example of setting procedure when changing the current system clock frequency to a slower frequency is shown in Table 8.8.

Table 8.8 Example of Setting Procedure When changing the current system clock frequency to a Slower frequency

No.	Step	Description
1	Start	
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1[*1*2] bit in PRCR register
3	Change Operating Power Control Mode to High-Speed mode[*1]	Change to High-Speed mode with OPCCR register.[*1] For details, see section 10, Low Power Modes.
4	Set a clock source to oscillate[*3]	Set a clock source that switches to the system clock source to oscillate.[*4]
5	Set internal clock division ratio	Make the following settings so that the frequency does not exceed the electrical characteristics range when the system clock source is switched in the next step. ● Set the internal clock division ratio larger than the current division ratio. ● Set the internal clock division ratio so that the frequency after switching the system clock source is slower than the current frequency.[*3]
6	Switch the System Clock source[*3]	Switch the system clock source with SCKSCR register. Before switching the system clock source, confirm that the clock source oscillation is stable.
7	Re-set internal clock division ratio	Re-set the internal clock division ratio with SCKDIVCR register according to the frequency after switching the system clock source.
8	Set Flash Wait Cycle	Set an appropriate access wait count for the flash memory according to the frequency at which it operates after the system clock (ICLK) is set. For details, see section 52, Flash Memory.
9	Change Operating Power Control Mode	Change the Operating Power Control Mode with OPCCR register. [*2]
10	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 [*1*2] bit in PRCR register
11	End	System clock is changed to a Slower frequency.

Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.
Note 2. Setting is not necessary if the operating power control mode is not changed.
Note 3. Setting is not necessary if the system clock source is not changed.
Note 4. To change the frequency of the PLL selected as the system clock source, it is necessary to change the system clock source to another clock source, stop the PLL, and then change the PLL setting.

When the PLL1 is selected as the system clock source, additional procedure is required for procedure No.5. This additional procedure is shown in Figure 8.15. It is recommended to use software to measure the wait time. Be sure to consider the worst-case conditions to ensure that the required wait time elapses. If an interrupt is unavoidably generated during the wait time, retry the measurement after return from the interrupt.



Note 1. 150µs is in DCDC mode, 10µs in External VDD mode

Figure 8.15 Additional flow of system clock setting when PLL is used for the system clock source and CPUCLK to be slower frequency

8.11.2 Peripheral module-dedicated Clock Setting

This section describes the setting procedure of the operating clock for the peripheral module. The target clocks are as follows : SCICLK, SPICLK, OCTACLK, CANFDCLK, USBCLK, USB60CLK, LCDCLK and I3CCLK.

8.11.2.1 Initial Peripheral module-dedicated Clock Setting

An example of the initial clock setting procedure for peripheral module is shown in Table 8.9.

Table 8.9 Example of the setting procedure of the operating clock for peripheral module after reset release / after Deep Software Standby cancellation (1 of 2)

No.	Step	Description
1	Start	After reset release / Deep software standby cancellation (MOCO is selected as system clock source)
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 *1*2bit in PRCR register.

Table 8.9 Example of the setting procedure of the operating clock for peripheral module after reset release / after Deep Software Standby cancellation (2 of 2)

No.	Step	Description
3	Change Operating Power Control Mode to High-Speed mode*1	Change to High-Speed mode with OPCCR register. *1 For details, see section 10, Low Power Modes.
4	Set CKSREQ and wait for setting CKSRDY	Write 1 to CKSREQ bit in each CKCR register. Polling until CKSRDY bit in each CKCR register is read as 1. While CKSRDY is 1, the clock supplying stops.
5	Set a clock source to oscillate	Set a clock source that switches to the peripheral module-dedicated clock source to oscillate.
6	Set the clock division ratio and switch the clock source	Write the setting value to CKDIV[2:0] in each CKDIVCR register. Write the setting value to CKSEL[3:0] in each CKCR register.
7	Clear CKSREQ and wait for clearing CKSRDY	Write 0 to CKSREQ bit in each CKCR register. Polling until CKSRDY bit in each CKCR register is read as 0. After CKSRDY is 0, the clock supplying starts.
8	Cancel the Module Stop Control	Cancel the Module Stop Control in MSTPCRn register (n = A to E) and supply clock to peripheral module.
9	Change Operating Power Control Mode	Change the Operating Power Control Mode with OPCCR register.*2
10	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register
11	End	Clock setting is completed. And then, set the peripheral modules.

Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.
Note 2. Setting is not necessary if the operating power control mode is not changed.

8.11.2.2 Peripheral module-dedicated Clock source changing

Table 8.10 shows an example of the setting change procedure when switching the clock source of the operating clock for peripheral module in use.

Table 8.10 Example of the setting change procedure when switching the clock source of the operating clock for peripheral module in use

No.	Step	Description
1	Start	The operating clock for the peripheral module is being used.
2	Stop the peripheral module	Stop the operation of the peripheral module that the operating clock is to be changed.
3	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1*1*2bit in PRCR register
4	Change Operating Power Control Mode to High-Speed mode *1	Change to High-Speed mode with OPCCR register.*1 For details, see section 10, Low Power Modes.
5	Set CKSREQ and wait for setting CKSRDY	Write 1 to CKSREQ bit in each CKCR register. Polling until CKSRDY bit in each CKCR register is read as 1. While CKSRDY is 1, the clock supplying stops.
6	Set a clock source to oscillate	Set a clock source that switches to the peripheral module-dedicated clock source to oscillate.
7	Set the clock division ratio and switch the clock source	Write the setting value to CKDIV[2:0] in each CKDIVCR register. Write the setting value to CKSEL[3:0] in each CKCR register.
8	Clear CKSREQ and wait for clearing CKSRDY	Write 0 to CKSREQ bit in each CKCR register. Polling until CKSRDY bit in each CKCR register is read as 0. After CKSRDY is 0, the clock supplying starts.
9	Change Operating Power Control Mode	Change the Operating Power Control Mode with OPCCR register. *2
10	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register
11	Restart the peripheral module	Change the peripheral module settings according to the operating clock after the settings are changed. After that, restart the operation of the peripheral module.
12	End	Clock setting is completed.

Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.
Note 2. Setting is not necessary if the operating power control mode is not changed.

8.11.2.3 Peripheral module-dedicated Clock division ratio changing

Table 8.11 shows an example of the setting change procedure when switching the division ratio of the operating clock for peripheral module in use.

Table 8.11 Example of the setting change procedure when switching the division ratio of the operating clock for peripheral module in use (without the clock source change)

No.	Step	Description
1	Start	The operating clock for the peripheral module is being used.
2	Stop the peripheral module	Stop the operation of the peripheral module that the operating clock is to be changed.
3	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 *1*2 bit in PRCR register
4	Change Operating Power Control Mode to High-Speed mode *1	Change to High-Speed mode with OPCCR register. *1 For details, see section 10, Low Power Modes.
5	Set the clock division ratio	Write the setting value to CKDIV[2:0] in each CKDIVCR register.
6	Change Operating Power Control Mode	Change the Operating Power Control Mode with OPCCR register.*2
7	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register
8	Restart the peripheral module	Change the peripheral module settings according to the operating clock after the settings are changed. After that, restart the operation of the peripheral module.
9	End	Clock setting is completed.

Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.
Note 2. Setting is not necessary if the operating power control mode is not changed.

8.11.3 Main Clock Oscillator Setting

Table 8.12 shows an example of the initial setting procedure for the Main Clock Oscillator.

Table 8.12 Example of the initial setting procedure for Main Clock Oscillator after reset release / after Deep Software Standby cancellation

No.	Step	Description
1	Start	The Main Clock Oscillator is stopped after reset release / Deep software standby cancellation
2	Cancel Register Write Protection	Set 1 to PRC0 bit in PRCR register
3	Set the mode of Main Clock Oscillator	Set the following Main Clock Oscillator mode with MOMCR register ● Resonator or Clock input ● Auto Gain Control function ● Drive Capability
4	Set the oscillation stabilization wait time	Set the Main Clock Oscillator Wait Time in MOSCWTCR register.
5	Set the oscillation keep in Software Standby mode	Set MOSCSCR register if Main Clock Oscillator keeps oscillation in Software Standby mode.
6	Set the Main Clock Oscillator to operate	Set the Main Clock Oscillator to start oscillating with the MOSCCR register.
7	Wait for Main Clock oscillation to stabilize	Polling until MOSCSF bit in OSCSF register is read as 1 (Oscillation is stable)
8	Apply Register Write Protection	Set 0 to PRC0 bit in PRCR register
9	End	The Main Clock Oscillator setting is completed. The Main Clock is available.

8.11.4 Sub-Clock Oscillator Setting

Table 8.13 shows an example of the initial setting procedure for the Sub-Clock Oscillator.

Table 8.13 Example of the initial setting procedure for Sub-Clock Oscillator after VBAT Power On Reset release

No.	Step	Description
1	Start	The Sub-Clock Oscillator is stopped after VBAT Power On Reset release.
2	Cancel Register Write Protection	Set 1 to PRC0 bit in PRCR register
3	Setting the mode of Sub-Clock Oscillator	Set the following Sub-Clock Oscillator mode with SOMCR register ● Resonator or Clock input ● Drive Capability
4	Set the Sub-Clock Oscillator to operate	Set the Sub-Clock Oscillator to start oscillating with the SOSCCR register.
5	Wait for Sub-Clock oscillation to stabilize	Wait for sub-clock oscillation stabilization time (tSUBOSCWT). For the sub-clock oscillation stabilization time, see section 60, Electrical Characteristics.
6	Apply Register Write Protection	Set 0 to PRC0 bit in PRCR register
7	End	The Sub-Clock Oscillator setting is completed. The Sub-Clock is available.

8.11.5 High-speed On Chip Oscillator Setting

8.11.5.1 HOCO Initial Clock Setting

Table 8.14 shows an example of the initial setting procedure for the HOCO.

Table 8.14 Example of the HOCO initial setting procedure after reset release / after Deep Software Standby cancellation (OFS1(_SEC).HOCOEN = 1, without FLL)

No.	Step	Description
1	Start	The HOCO is stopped after reset release / Deep software standby cancellation when OFS1(_SEC).HOCOEN is 1.
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 bit in PRCR register
3	Check the HOCO power supply*1	Check the following bit in the HOCOLDOCR register ● LDOSTP bit is 0 (LDO is enabled)
4	Set the HOCO frequency	Set the HOCO frequency with HOCOCR2 register
5	Set the oscillation keep in Software Standby mode	If HOCO keeps oscillation in Software Standby mode, set the following: ● HOCOSOKP bit in HOCOSCR register ● SKEEP bit in HOCOLDOCR register
6	Set HOCO to operate	Set HOCO to start oscillating with the HOCOCR register.
7	Wait for HOCO clock oscillation to stabilize	Polling until HOCOSF bit in OSCSF register is read as 1 (Oscillation is stable)
8	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register
9	End	HOCO clock setting is completed.HOCO Clock is available.

Note 1. When HOCOLDOCR.LDOSTP bit is 1, do not operate HOCO. If HOCOLDOCR.LDOSTP is switched from 1 to 0, waiting for the HOCO power supply to stabilize is required. See section 60, Electrical Characteristics on waiting time for the HOCO power supply to stabilize.

8.11.5.2 HOCO Setting with FLL function

Table 8.15shows an example of the initial setting procedure for HOCO with FLL function.

Table 8.16 shows the FLL function setting flow for the Software Standby mode transition and cancellation.

Table 8.15 Example of the HOCO setting procedure with FLL function after reset release / after Deep Software Standby cancellation (OFS1(_SEC).HOCOEN = 1) (1 of 2)

No.	Step	Description
1	Start	The HOCO is stopped after reset release / Deep software standby cancellation when OFS1(_SEC).HOCOEN is 1.
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 bit in PRCR register

Table 8.15 Example of the HOCO setting procedure with FLL function after reset release / after Deep Software Standby cancellation (OFS1(_SEC).HOCOEN = 1) (2 of 2)

No.	Step	Description
3	Check the HOCO power supply*1	Check the following bit in the HOCOLDOCR register. ● LDOSTP bit is 0 (LDO is enabled)
4	Set the HOCO frequency	Set the HOCO frequency with HOCOCR2 register
5	Set the FLL function to enable*2	Set the FLL Multiplication Control with FLLCR2 register. Set the FLL function to enable with FLLCR1 register.
6	Set HOCO to operate	Set HOCO to start oscillating with the HOCOCR register.
7	Wait for HOCO clock oscillation to stabilize	Polling until HOCOSF bit in OSCSF register is read as 1 (Oscillation is stable)
8	Wait for FLL stabilization	Wait for FLL stabilization wait time (t_{FLLWT}), or wait until the HOCO clock is measured to confirm that the frequency accuracy is stable.
9	Check the HOCO stabilization	Check that HOCOSF bit in OSCSF register is read as 1
10	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register
11	End	FLL setting is completed. HOCO Clock is available.

Note 1. When HOCOLDOCR.LDOSTP bit is 1, do not operate HOCO.
If HOCOLDOCR.LDOSTP is switched from 1 to 0, waiting for the HOCO power supply to stabilize is required. See section 60, Electrical Characteristics on waiting time for the HOCO power supply to stabilize.
Note 2. Sub-Clock Oscillator must be oscillating with the stabilization.

Table 8.16 FLL setting flow for Software Standby mode transition and cancellation

No.	Step	Description
1	Start	The HOCO is oscillating with FLL is enabled
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 bit in PRCR register
3	Stop HOCO*1	Set HOCO to stop with HOCOCR register.
4	Disable FLL	Set the FLL function to disable with FLLCR1 register.
5	WFI instruction*2	Transition to Software Standby mode with WFI instruction
6	Software Standby mode*2	MCU is in Software Standby mode
7	Cancellation Software Standby mode*2	Software Standby mode is cancelled
8	Set the FLL function to enable*3	Set the FLL Multiplication Control with FLLCR2 register. Set the FLL function to enable with FLLCR1 register.
9	Set HOCO to operate	Set HOCO to start oscillating with the HOCOCR register.
10	Wait for FLL stabilization	Wait for FLL stabilization wait time (t_{FLLWT}), or wait until the HOCO clock is measured to confirm that the frequency accuracy is stable.
11	Check the HOCO stabilization	Check that HOCOSF bit in OSCSF register is read as 1.
12	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register
13	End	HOCO Clock is available.

Note 1. If HOCO is used as the system clock source or the clock source of PLL, these clock source must be changed to another clock before HOCO is stopped.
Note 2. See section 10, Low Power Modes for detail of Software Standby mode.
Note 3. Sub-Clock Oscillator must be oscillating with the stabilization.

8.11.6 PLL Setting

8.11.6.1 PLL Initial Clock Setting

Table 8.17 shows an example of the initial setting procedure for the PLL.

Table 8.17 Example of the PLL initial setting procedure after reset release / after Deep Software Standby cancellation (PLL1 or PLL2)

No.	Step	Description
1	Start	The PLL1 is stopped after reset release / Deep software standby cancellation. The PLL2 is stopped after reset release / Deep software standby cancellation.
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 bit in PRCR register.
3	Change Operating Power Control Mode to High-Speed mode*1	Change to High-Speed mode with OPCCR register. For details, see section 10, Low Power Modes.
4	Check the PLL power supply*2	[PLL1] Check the following bit in the PLL1LDOCR register. ● LDOSTP bit is 0 (LDO is enabled). [PLL2] Check the following bit in the PLL2LDOCR register. ● LDOSTP bit is 0 (LDO is enabled).
5	Configure the PLL settings	[PLL1] Configure the following PLL1 settings in PLLCCR and PLLCCR2 registers. [PLL2] Configure the following PLL2 settings in PLL2CCR and PLL2CCR2 registers. [Setting Items] ● PLL input frequency division ratio ● PLL clock source ● Frequency Multiplication Factor ● Frequency Multiplication Fractional Factor ● PLL Output Frequency Division Ratio (P/Q/R)
6	Set PLL to operate	[PLL1] Set PLL1 to start oscillating with the PLLCR register. [PLL2] Set PLL2 to start oscillating with the PLL2CR register.
7	Wait for PLL clock oscillation to stabilize	[PLL1] Polling until PLLSF bit in OSCSF register is read as 1(Oscillation is stable). [PLL2] Polling until PLL2SF bit in OSCSF register is read as 1 (Oscillation is stable).
8	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register
9	End	PLL1 or PLL2 clock setting is completed. [PLL1] PLL1P, PLL1Q, PLL1R clocks are available. [PLL2] PLL2P, PLL2Q, PLL2R clocks are available.

Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.
Note 2. When PLL1LDOCR.LDOSTP bit is 1, do not operate PLL1.
If PLL1LDOCR.LDOSTP is switched from 1 to 0, waiting for the PLL1 power supply to stabilize is required. See section 60, Electrical Characteristics on waiting time for the PLL1 power supply to stabilize.
When PLL2LDOCR.LDOSTP bit is 1, do not operate PLL2.
If PLL2LDOCR.LDOSTP is switched from 1 to 0, waiting for the PLL2 power supply to stabilize is required. See section 60, Electrical Characteristics on waiting time for the PLL2 power supply to stabilize.

8.11.6.2 PLL Clock Setting Change

Table 8.18 shows an example of the initial setting procedure for the PLL.

Table 8.18 Example of the PLL setting change procedure (PLL1 or PLL2) (1 of 2)

No.	Step	Description
1	Start	PLL1 is oscillating and the PLL1 clocks (PLL1P, PLL1Q, PLL1R) are being used. PLL2 is oscillating and the PLL2 clocks (PLL2P, PLL2Q, PLL2R) are being used.
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 bit in PRCR register.

Table 8.18 Example of the PLL setting change procedure (PLL1 or PLL2) (2 of 2)

No.	Step	Description
3	Stop PLL*1	[PLL1] Set PLL1 to stop with PLLCR register. [PLL2] Set PLL2 to stop with PLL2CR register.
4	Change the PLL settings	[PLL1] Configure the following PLL1 settings in PLLCCR and PLLCCR2 registers. [PLL2] Configure the following PLL2 settings in PLL2CCR and PLL2CCR2 registers. [Setting Items] ● PLL input frequency division ratio ● PLL clock source ● Frequency Multiplication Factor ● Frequency Multiplication Fractional Factor ● PLL Output Frequency Division Ratio (P/Q/R)
5	Set PLL to operate	[PLL1] Set PLL1 to start oscillating with the PLLCR register. [PLL2] Set PLL2 to start oscillating with the PLL2CR register.
6	Wait for PLL clock oscillation to stabilize	[PLL1] Polling until PLLSF bit in OSCSF register is read as 1 (Oscillation is stable). [PLL2] Polling until PLL2SF bit in OSCSF register is read as 1 (Oscillation is stable).
7	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register.
8	End	PLL1 or PLL2 clock setting change is completed. [PLL1] PLL1P, PLL1Q, PLL1R clocks are available. [PLL2] PLL2P, PLL2Q, PLL2R clocks are available.

Note 1. If PLL1P clock is used as the system clock source, the system clock source must be changed to another clock source before PLL1 is stopped.

8.12 Usage Notes

8.12.1 Notes on Clock Generation Circuit

The frequency of the following clocks supplied to each module changes according to the setting of the SCKDIVCR and SCKDIVCR2 register:

- CPU clock (CPUCLK)
- System clock (ICLK)
- Peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD and PCLKE)
- FlashIF clock (FCLK)
- external bus clock (BCLK)

Each frequency must meet the following conditions:

- Each frequency must be selected within the operation-guaranteed range of the operating frequency (f) specified in the AC characteristics. See section 60, Electrical Characteristics.
- Each clock must be set according to Table 8.2.
- Do not change the clock frequency during external bus access. Furthermore, when access via the external bus is to start after a change to the clock frequency, only start access via the bus after confirming that the change to the frequencies has been completed.

To ensure correct processing after the clock frequency changes, first write to the relevant Clock Control register to change the frequency, then read the value from the register, and finally perform the subsequent processing.

8.12.2 Notes on Resonator

Because various resonator characteristics relate closely to your board design, adequate evaluation is required before use. See the resonator connection example in Figure 8.5. The circuit constants for the resonator depend on the resonator to be used and the stray capacitance of the mounting circuit. Therefore, consult the resonator manufacturer when determining the circuit constants. The voltage to be applied between the resonator pins must be within the absolute maximum rating.

8.12.3 Notes on Board Design

When using a crystal resonator, place the resonator and its load capacitors as close to the XTAL and EXTAL pins as possible. Other signal lines should be routed away from the oscillation circuit as shown in Figure 8.16 to prevent electromagnetic induction from interfering with correct oscillation. Figure 8.16 shows the case which the main clock oscillator is used. In case of sub-clock oscillator, it is also the same as Figure 8.16.

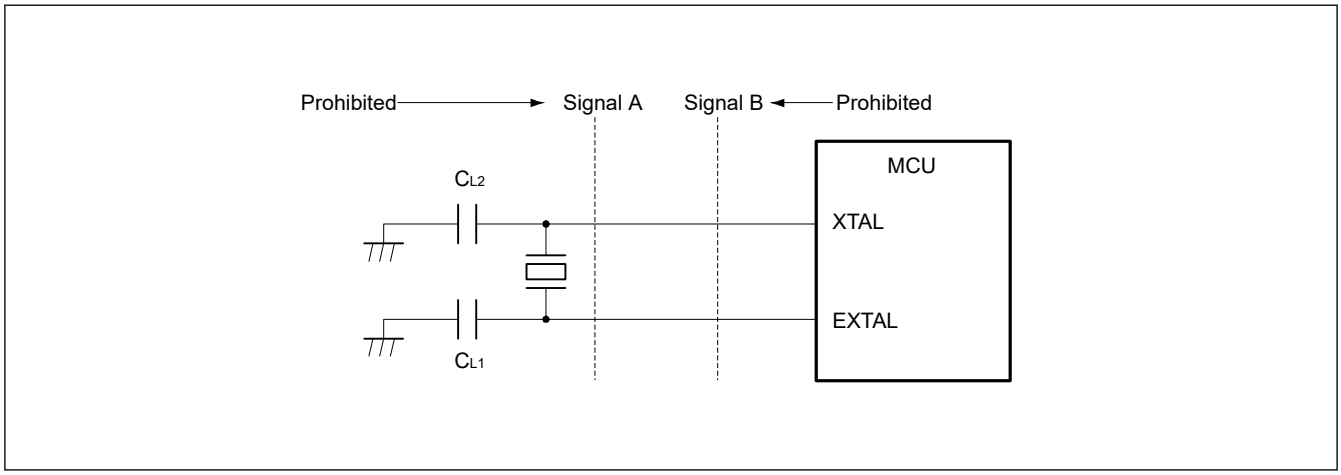


Figure 8.16 Signal routing in board design for oscillation circuit

8.12.4 Notes on Resonator Connect Pin

When the main clock is not used, the EXTAL and XTAL pins can be used as general ports. When these pins are used as general ports, the main clock must be stopped (MOSCCR.MOSTP bit should be set to 1).

8.12.5 Notes on Using Sub-Clock Oscillator

The output of the P212 (EXTAL), P213 (XTAL) and P706 pins may affect the oscillation by the sub-clock oscillator.

If the sub-clock oscillator is used, implement board design so as not to affect the oscillation. Renesas strongly recommends setting the PmnPFS.DSCR[1:0] bits to 00b or 01b when using the P212 (EXTAL), P213 (XTAL) and P706 as output pins and using the sub-clock oscillator.

In addition, when using the sub-clock oscillator in Low power mode drive capability (SOMCR.SODRV[1:0]= 01b, 10b and 11b), Renesas recommends setting the PmnPFS.DSCR[1:0] bits to 00b when using the P212 (EXTAL), P213 (XTAL) and P706 as output pins and using the sub-clock oscillator.

8.12.6 Notes on MOSCSCR.MOSCSOKP

When setting MOSCSCR.MOSCSOKP = 1 and transit to Software Standby mode, apply one of the workarounds in Workaround 1 or Workaround 2.

If the workaround below is not applicable, set MOSCSCR.MOSCSOKP = 0, because there is a potential that resuming from Software Standby mode does not work properly.

Workaround 1

Execute Steps 1. and 2. in the following procedure when reset is released.

1. Set MOSCCR.MOSTP = 1 to stop MOSC and set OSTDCR.OSTDE = 1 to enable the oscillation stop detection function.
2. Set MOSCCR. MOSTP = 0 to start MOSC and wait for the MOSC to be stabilized by checking OSCSF.MOSCSF.

Execute the following Step 3. before WFI instruction execution or Sleep-on-exit before transition to Software Standby mode.

3. Set either (a) or (b) for USBCLK, OCTACLK, CANFDCLK, USB60CLK, I3CCLK, SCICLK, SPICLK, or LCDCLK clocks.
   (a) Set the MSTP bit of each clock to 1.
   (b) Select PLL1P, PLL1Q, PLL1R, PLL2P, PLL2Q, or PLL2R as the clock source. (*CKCR.*CKSEL[3:0] = 4'b0101, 0110, 0111, 1000, 1001, 1010).

When implementing Workaround 1, the following restrictions apply:

- MOSC cannot be selected as the clock source for the system clock or peripheral IP dedicated clock.*1
- CACMCLK, CANMCLK, and MIPIMCLK cannot be used.*1
- MOSC oscillation stop detection function cannot be used.

Note 1. It is possible to select MOSC as the reference clock of the PLL, select MOSC as the clock source output from the CLKOUT pin, and use USBMCLK.

No special steps are required after resuming from Software Standby mode. When a reset occurs, it is necessary to follow the procedure from Step 1..

Workaround 2

Execute Steps 1. through 6. in the following procedure before WFI instruction execution or Sleep-on-exit before transition to Software Standby mode.

1. Set the following clock settings.
   - Select system clock source from other than MOSC or PLL1 whose clock source is MOSC.
   - Select peripheral clock source from other than MOSC, PLL1 whose clock source is MOSC, PLL2 whose clock source is MOSC.
   - Stop MOSC, PLL1, and PLL2.
   - Stop the module whose clock source is either CACMCLK or CANMCLK or MIPIMCLK.
2. Set MOSCCR.MOSTP = 1 to stop MOSC and set OSTDCR.OSTDE = 1 to enable the oscillation stop detection function.
3. Set MOSCCR. MOSTP = 0 to start MOSC and wait for the MOSC to be stabilized by checking OSCSF.MOSCSF.
4. If PLL1 whose clock source is MOSC is used as the clock source for the system clock or peripheral IP, set PLLCR.PLLSTP = 0 to start PLL1 and wait until OSCSF. PLLSF = 1 for PLL to be stabilized. Start using PLL1.
5. If PLL2 whose clock source is MOSC is used as the clock source for the system clock or peripheral IP, set PLL2CR.PLL2STP = 0 to start PLL2 and wait until OSCSF. PLL2SF = 1 for PLL to be stabilized. Start using PLL2.
6. Set either (a) or (b) for USBCLK, OCTACLK, CANFDCLK, USB60CLK, I3CCLK, SCICLK, SPICLK, or LCDCLK clocks.
   (a) Set the MSTP bit of each clock to 1.
   (b) Select PLL1P, PLL1Q, PLL1R, PLL2P, PLL2Q, or PLL2R as the clock source. (*CKCR.*CKSEL[3:0] = 4'b0101, 0110, 0111, 1000, 1001, 1010).

   Execute Steps 7. through 11. after resuming from Software Standby mode.
7. Execute either (a) or (b).
   (a) Set OSTDCR.OSTDE = 0 to disable the oscillation stop detection function.
   (b) Select the system clock source from other than MOSC or PLL1, and clear the oscillation stop detection flag to cancel the oscillation stop detection state.
8. If Step (b) in the procedure is performed, change the system clock to any clock.
9. When the MSTP bit is set to 1 in Step (a), set the MSTP bit to 0 for the module that needs to be used.

10. If you changed the clock settings of USBCLK, OCTACLK, CANFDCLK, USB60CLK, I3CCLK, SCICLK, SPICLK, or LCDCLK in Step 6., change the clock setting to be required. In this Step 10., MOSC, PLL1, and PLL2 can be used for source clock.
11. If CACMCLK, CANMCLK, or MIPIMCLK is required, cancel module stop of the required IP.

When using Workaround 2, the following limitations apply:

- MOSC cannot be selected as the clock source for the system clock or peripheral IP clock from the start of this workaround until resuming from Software Standby mode.
- CACMCLK, CANMCLK, and MIPIMCLK cannot be used from the start of this workaround until resuming from Software Standby mode.
- MOSC oscillation stop detection function cannot be used from the start of this workaround until resuming from Software Standby mode.

9. Clock Frequency Accuracy Measurement Circuit (CAC)

9.1 Overview

The Clock Frequency Accuracy Measurement Circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock selected as the measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range. When measurement is complete or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated.

Table 9.1 lists the CAC specifications, Figure 9.1 shows the CAC block diagram, and Table 9.2 lists the CAC I/O pin.

Table 9.1 CAC specifications

Parameter	Specifications
Measurement target clocks	Frequency can be measured for: • Main clock oscillator (CACMCLK) • Sub-clock oscillator (CACSCLK) • HOCO clock (CACHCLK) • MOCO clock (CACMOCLK) • LOCO clock (CACLCLK) • Peripheral module clock B (PCLKB)
Measurement reference clocks	Frequency can be referenced to: • External clock input to the CACREF pin • Main clock oscillator (CACMCLK) • Sub-clock oscillator (CACSCLK) • HOCO clock (CACHCLK) • MOCO clock (CACMOCLK) • LOCO clock (CACLCLK) • Peripheral module clock B (PCLKB)
Selectable function	Digital filter
Interrupt sources	• Measurement end • Frequency error • Overflow
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security and Privilege attribution can be set

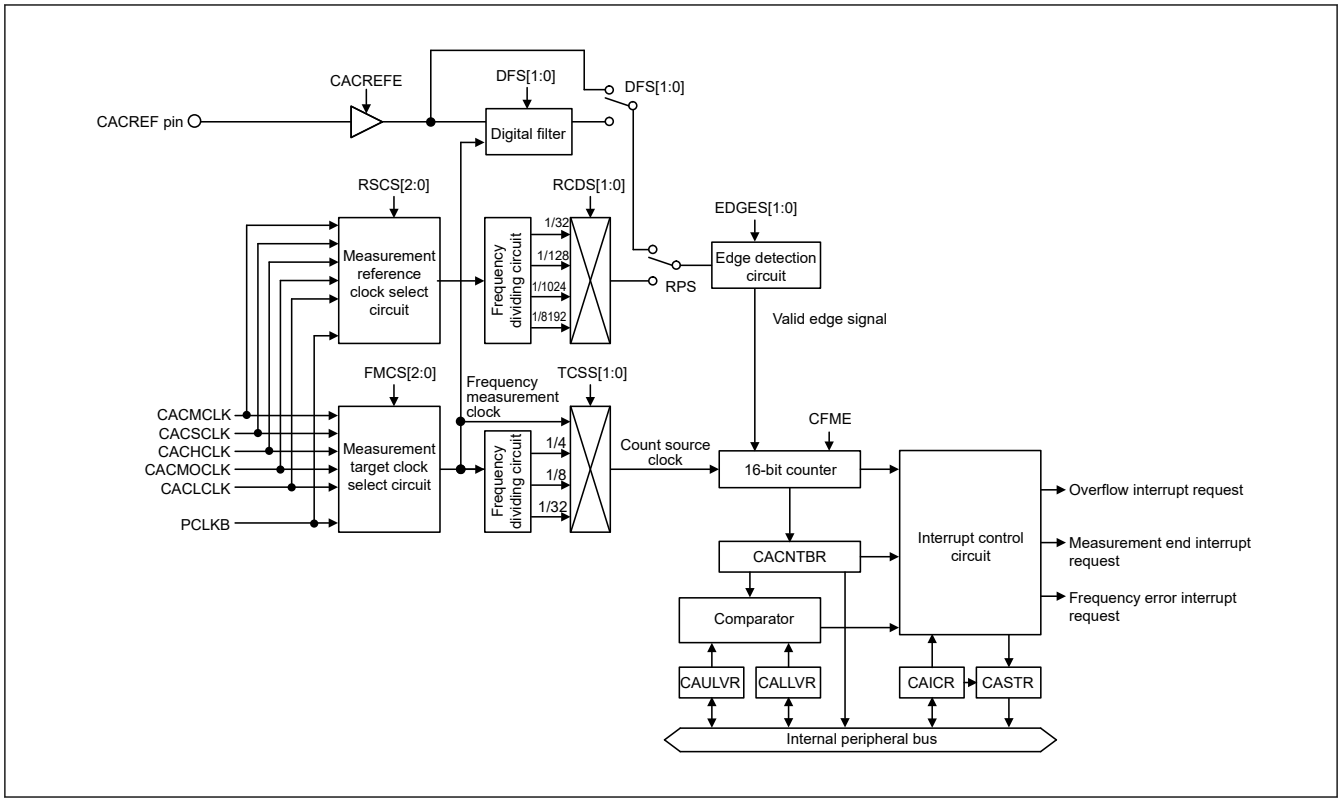


Figure 9.1 CAC block diagram

Table 9.2 CAC I/O pin

Function	Pin name	I/O	Description
CAC	CACREF	Input	Measurement reference clock input pin

9.2 Register Descriptions

9.2.1 CACR0 : CAC Control Register 0

Base address: CAC = 0x4020_2400
CAC_NS = 0x5020_2400

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFME
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CFME	Clock Frequency Measurement Enable 0: Disable 1: Enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

CFME bit (Clock Frequency Measurement Enable)

The CFME bit enables clock frequency measurement. Changes made to this bit are not immediately reflected to the internal circuit. Read the bit to confirm that the change has been reflected.

9.2.2 CACR1 : CAC Control Register 1

Base address: CAC = 0x4020_2400
CAC_NS = 0x5020_2400

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	EDGES[1:0]		TCSS[1:0]		FMCS[2:0]			CACREFE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CACREFE	CACREF Pin Input Enable 0: Disable 1: Enable	R/W
3:1	FMCS[2:0]	Measurement Target Clock Select 0 0 0: Main clock oscillator (CACMCLK) 0 0 1: Sub-clock oscillator (CACSCLK) 0 1 0: HOCO clock (CACHCLK) 0 1 1: MOCO clock (CACMOCLK) 1 0 0: LOCO clock (CACLCLK) 1 0 1: Peripheral module clock B (PCLKB) 1 1 0: Setting prohibited 1 1 1: Setting prohibited	R/W
5:4	TCSS[1:0]	Timer Count Clock Source Select 0 0: No division 0 1: × 1/4 clock 1 0: × 1/8 clock 1 1: × 1/32 clock	R/W
7:6	EDGES[1:0]	Valid Edge Select 0 0: Rising edge 0 1: Falling edge 1 0: Both rising and falling edges 1 1: Setting prohibited	R/W

Note: S-TYPE3, P-TYPE3
Note: Set the CACR1 register when the CACR0.CFME bit is 0.

CACREFE bit (CACREF Pin Input Enable)

The CACREFE bit enables the CACREF pin input.

FMCS[2:0] bits (Measurement Target Clock Select)

The FMCS[2:0] bits select the measurement target clock whose frequency is to be measured.

TCSS[1:0] bits (Timer Count Clock Source Select)

The TCSS[1:0] bits select the division ratio of the measurement target clock.

EDGES[1:0] bits (Valid Edge Select)

The EDGES[1:0] bits select the valid edge for the reference signal.

9.2.3 CACR2 : CAC Control Register 2

Base address: CAC = 0x4020_2400
CAC_NS = 0x5020_2400

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DFS[1:0]		RCDS[1:0]		RSCS[2:0]			RPS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RPS	Reference Signal Select 0: CACREF pin input 1: Internal clock (internally generated signal)	R/W
3:1	RSCS[2:0]	Measurement Reference Clock Select 0 0 0: Main clock oscillator (CACMCLK) 0 0 1: Sub-clock oscillator (CACSCLK) 0 1 0: HOCO clock (CACHCLK) 0 1 1: MOCO clock (CACMOCLK) 1 0 0: LOCO clock (CACLCLK) 1 0 1: Peripheral module clock B (PCLKB) 1 1 0: Setting prohibited 1 1 1: Setting prohibited	R/W
5:4	RCDS[1:0]	Measurement Reference Clock Frequency Division Ratio Select 0 0: × 1/32 clock 0 1: × 1/128 clock 1 0: × 1/1024 clock 1 1: × 1/8192 clock	R/W
7:6	DFS[1:0]	Digital Filter Select 0 0: Disable digital filtering 0 1: Use sampling clock for the digital filter as the frequency measuring clock 1 0: Use sampling clock for the digital filter as the frequency measuring clock divided by 4 1 1: Use sampling clock for the digital filter as the frequency measuring clock divided by 16.	R/W

Note: S-TYPE3, P-TYPE3
Note: Set the CACR2 register when the CACR0.CFME bit is 0.

RPS bit (Reference Signal Select)

The RPS bit selects whether to use the CACREF pin input or an internal clock (internally generated signal) as the reference signal.

RSCS[2:0] bits (Measurement Reference Clock Select)

The RSCS[2:0] bits select the reference clock for measurement.

RCDS[1:0] bits (Measurement Reference Clock Frequency Division Ratio Select)

The RCDS[1:0] bits select the frequency-divisor of the reference clock for measurement when an internal reference clock is selected. When RPS = 0 (CACREF pin is used as the reference clock source), the reference clock is not divided.

DFS[1:0] bits (Digital Filter Select)

The DFS[1:0] bits enable or disable the digital filter and selects its sampling clock.

9.2.4 CAICR : CAC Interrupt Control Register

Base address: CAC = 0x4020_2400
CAC_NS = 0x5020_2400

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	OVFFCL	MENDFCL	FERRFCL	—	OVFIE	MENDIE	FERRIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FERRIE	Frequency Error Interrupt Request Enable 0: Disable 1: Enable	R/W
1	MENDIE	Measurement End Interrupt Request Enable 0: Disable 1: Enable	R/W
2	OVFIE	Overflow Interrupt Request Enable 0: Disable 1: Enable	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	FERRFCL	FERRF Clear 0: No effect 1: The CASTR.FERRF flag is cleared	W
5	MENDFCL	MENDF Clear 0: No effect 1: The CASTR.MENDF flag is cleared	W
6	OVFFCL	OVFF Clear 0: No effect 1: The CASTR.OVFF flag is cleared.	W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

FERRIE bit (Frequency Error Interrupt Request Enable)

The FERRIE bit enables or disables the frequency error interrupt request.

MENDIE bit (Measurement End Interrupt Request Enable)

The MENDIE bit enables or disables the measurement end interrupt request.

OVFIE bit (Overflow Interrupt Request Enable)

The OVFIE bit enables or disables the overflow interrupt request.

FERRFCL bit (FERRF Clear)

Setting the FERRFCL bit to 1 clears the CASTR.FERRF flag.

MENDFCL bit (MENDF Clear)

Setting the MENDFCL bit to 1 clears the CASTR.MENDF flag.

OVFFCL bit (OVFF Clear)

Setting the OVFFCL bit to 1 clears the CASTR.OVFF flag.

9.2.5 CASTR : CAC Status Register

Base address: CAC = 0x4020_2400
CAC_NS = 0x5020_2400

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	OVFF	MENDF	FERRF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FERRF	Frequency Error Flag 0: Clock frequency is within the allowable range 1: Clock frequency has deviated beyond the allowable range (frequency error).	R
1	MENDF	Measurement End Flag 0: Measurement is in progress 1: Measurement ended	R
2	OVFF	Overflow Flag 0: Counter has not overflowed 1: Counter overflowed	R
7:3	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3

FERRF flag (Frequency Error Flag)

The FERRF flag indicates a deviation of the clock frequency from the set value (frequency error).

[Setting condition]

- The clock frequency is outside the allowable range defined in the CAULVR and CALLVR registers.

[Clearing condition]

- 1 is written to the FERRFCL bit.

MENDF flag (Measurement End Flag)

The MENDF flag indicates the end of measurement.

[Setting condition]

- Measurement ends.

[Clearing condition]

- 1 is written to the MENDFCL bit.

OVFF flag (Overflow Flag)

The OVFF flag indicates that the counter overflowed.

[Setting condition]

- The counter overflows.

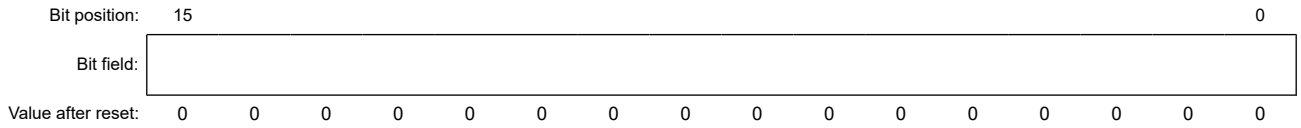
[Clearing condition]

- 1 is written to the CAICR.OVFFCL bit.

9.2.6 CAULVR : CAC Upper-Limit Value Setting Register

Base address: CAC = 0x4020_2400
CAC_NS = 0x5020_2400

Offset address: 0x06



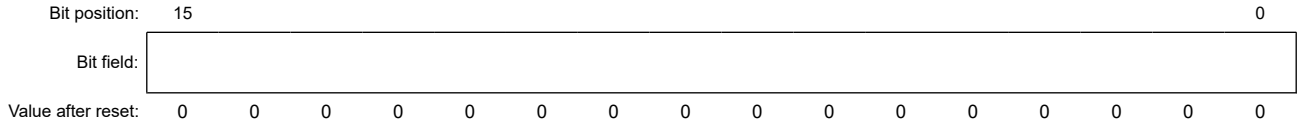
Bit	Symbol	Function	R/W
15:0	n/a	The Upper Value of the Allowable Range The CAULVR register is a 16-bit read/write register that specifies the upper value of the allowable range. When the counter value exceeds the value specified in this register, a frequency error is detected. Write to this register when the CACR0.CFME bit is 0. The counter value stored in CACNTBR can vary depending on the difference between the phases of the digital filter and edge-detection circuit, and the signal on the CACREF pin. Ensure that this setting allows an adequate margin.	R/W

Note: S-TYPE3, P-TYPE3

9.2.7 CALLVR : CAC Lower-Limit Value Setting Register

Base address: CAC = 0x4020_2400
CAC_NS = 0x5020_2400

Offset address: 0x08



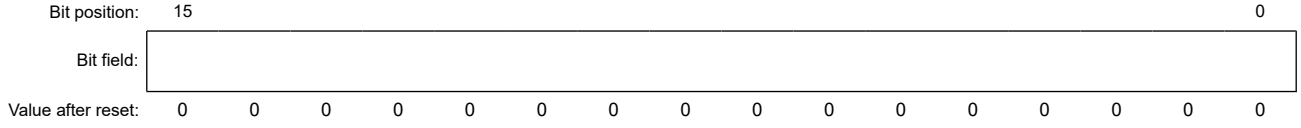
Bit	Symbol	Function	R/W
15:0	n/a	The Lower Value of the Allowable Range The CALLVR register is a 16-bit read/write register that specifies the lower value of the allowable range. When the counter value falls below the value specified in this register, a frequency error is detected. Write to this register when the CACR0.CFME bit is 0. The counter value stored in CACNTBR can vary depending on the difference between the phases of the digital filter and edge-detection circuit, and the signal on the CACREF pin. Ensure that this setting allows an adequate margin.	R/W

Note: S-TYPE3, P-TYPE3

9.2.8 CACNTBR : CAC Counter Buffer Register

Base address: CAC = 0x4020_2400
CAC_NS = 0x5020_2400

Offset address: 0x0A



Bit	Symbol	Function	R/W
15:0	n/a	The Measurement Result The CACNTBR register is a 16-bit read-only register that stores the measurement result.	R

Note: S-TYPE3, P-TYPE3

9.3 Operation

9.3.1 Measuring Clock Frequency

The CAC measures the clock frequency using the CACREF pin input or an internal clock as a reference. Figure 9.2 shows an operating example of the CAC.

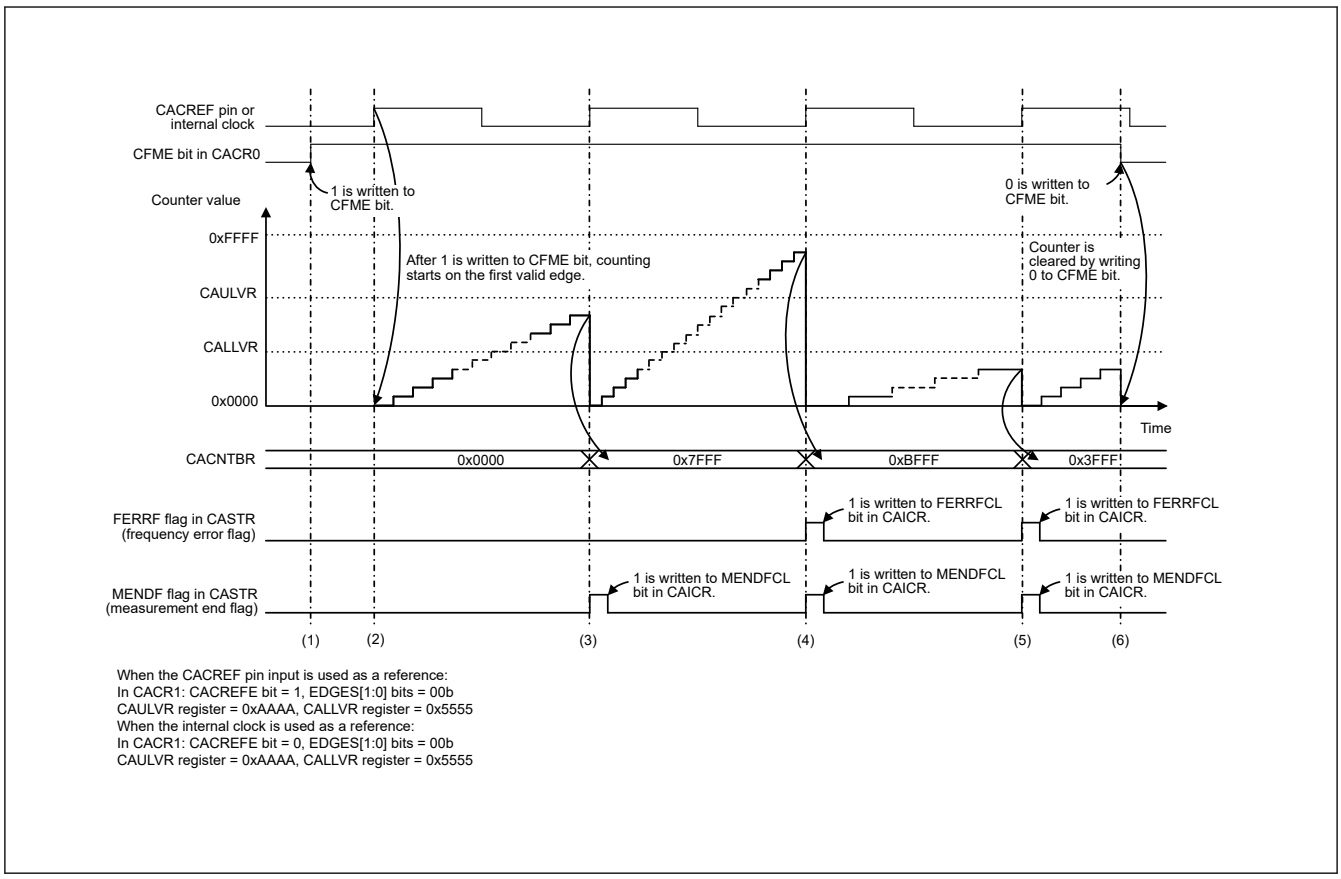


Figure 9.2 CAC operating example

The events in Figure 9.2 are:

1. When the CACREF pin input is used as reference (CACR1.CACREFE = 1), frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is set to 0 and the CACR1.CACREFE bit is set to 1. When the internal clock is used as reference (CACR1.CACREFE = 0), frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is set to 1.
2. When the CACREF pin input is used as reference, after 1 is written to the CFME bit, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits (rising edge (CACR1.EDGES[1:0] = 00b) in Figure 9.2) is input from the CACREF pin. When the internal clock is used as reference, after 1 is written to the CFME bit, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits (rising edge (CACR1.EDGES[1:0] = 00b) in Figure 9.2) is input based on the clock source selected by the CACR2.RSCS[2:0] bits.
3. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If both CACNTBR ≤ CAULVR and CACNTBR ≥ CALLVR are true, only the MENDF flag in CASTR is set to 1, because the clock frequency is correct. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
4. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If CACNTBR > CAULVR, the FERRF flag in CASTR is set to 1, because the clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. The MENDF flag in CASTR is set to 1 at the end of measurement. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
5. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If CACNTBR < CALLVR, the FERRF flag in CASTR is set to 1, because the clock frequency

is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. The MENDF flag in CASTR is set to 1 at the end of measurement. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.

6. When the CFME bit in CACR0 is 1, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR every time a valid edge is input. Writing 0 to the CFME bit in CACR0 clears the counter and stops up-counting.

9.3.2 Digital Filtering of Signals on CACREF Pin

The CACREF pin has a digital filter, and levels on the CACREF pin are transmitted to the internal circuitry after three consecutive matches in the selected sampling interval. The same level continues to be transmitted internally until the level on the pin has three consecutive matches again. Enabling or disabling of the digital filter and its sampling clock are selectable.

The counter value transferred to CACNTBR might be in error by up to 1 cycle of the sampling clock because of the difference between the phases of the digital filter and the signal input to the CACREF pin. When a frequency dividing clock is selected as a count source clock, the counter value error is obtained using the following formula:

$$\text{Counter value error} = (1 \text{ cycle of the count source clock}) / (1 \text{ cycle of the sampling clock})$$

9.4 Interrupt Requests

The CAC generates three types of interrupt requests:

- Frequency error interrupt
- Measurement end interrupt
- Overflow interrupt

When an interrupt source is generated, the associated status flag is set to 1. Table 9.3 provides information on the CAC interrupt requests.

Table 9.3 CAC interrupt requests

Interrupt request	Interrupt enable bit	Status flag	Interrupt sources
Frequency error interrupt	CAICR.FERRIE	CASTR.FERRF	The result of comparing CACNTBR with CAULVR and CALLVR is either CACNTBR > CAULVR or CACNTBR < CALLVR
Measurement end interrupt	CAICR.MENDIE	CASTR.MENDF	• Valid edge is input from the CACREF pin or internal clock • Measurement end interrupt does not occur at the first valid edge after writing 1 to the CACR0.CFME bit
Overflow interrupt	CAICR.OVFIE	CASTR.OVFF	Counter overflows

9.5 Usage Notes

9.5.1 Settings for the Module-Stop Function

The Module Stop Control Register C (MSTPCRC) can enable or disable CAC operation. The CAC module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

10. Low Power Modes

10.1 Overview

The MCU has several functions for reducing power consumption, such as setting clock dividers, EBCLK output control, SDCLK output control, stopping modules, power gating control, selecting operating power control mode in Normal mode, and transitioning to low power modes.

Table 10.1 lists the specifications of the low power mode functions. Table 10.2 to Table 10.4 lists the conditions to transition to low power modes, the states of the CPU and peripheral modules, and the method for canceling each mode. After a reset, the MCU enters the program execution state, but only the DMAC, DTC and SRAM operate.

Table 10.1 Specifications of the low power mode functions

Item	Specification
Reducing power consumption by switching clock signals	The frequency division ratio can be selected independently for the CPU clock (CPUCLK), system clock (ICLK), peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD, PCLKE), external bus clock (BCLK), and flash interface clock (FCLK). *1
EBCLK output control	BCLK output or high-level output can be selected.*1
SDCLK output control	SDCLK output or high-level output can be selected.*1
Module stop	Functions can be stopped independently for each peripheral module
Power gating control	This function can be controlled the power state of the power domain. ● Control the turning On/OFF for the power domain ● Control the retention of specific circuits during power gating
Processor low power modes	● CPU Sleep mode ● CPU Deep Sleep mode
Low-power modes	● Software Standby mode*2 ● Deep Software Standby mode 1, 2, 3*2
Operating power control modes	● Power consumption can be reduced in Normal and Processor low power mode by selecting an appropriate operating power control mode according to the operating frequency. ● Two operating power control modes are available: High-speed mode Low-speed mode*2
TrustZone Filter	Security and Privilege attribution can be set

Note 1. For details, see section 8, Clock Generation Circuit.
Note 2. This mode is not supported in external VDD mode.

Table 10.2 Operating state of processor low power mode (1 of 2)

Item	CPU Sleep mode	CPU Deep Sleep mode
Transition condition	When [Condition 1] or [Condition 2] while CPU.SCR.SLEEPDEEP = 0 [Condition 1] ● WFI instruction ● A valid interrupt request *4 cannot be accepted to CPU (including a transition from the time WFI instruction is executed to the time the transition to CPU Sleep mode is completed). [Condition 2] ● CPU.SCR.SLEEPONEXIT = 1 ● Complete execution of all exception handlers ● A valid interrupt request*4 cannot be accepted to CPU (including a transition from the time WFI instruction is executed to the time the transition to CPU Sleep mode is completed).	When [Condition 1] or [Condition 2] while CPU.SCR.SLEEPDEEP = 0 [Condition 1] ● WFI instruction ● A valid interrupt request *4 cannot be accepted to CPU (including a transition from the time WFI instruction is executed to the time the transition to CPU Sleep mode is completed). [Condition 2] ● CPU.SCR.SLEEPONEXIT = 1 ● Complete execution of all exception handlers ● A valid interrupt request*4 cannot be accepted to CPU (including a transition from the time WFI instruction is executed to the time the transition to CPU Sleep mode is completed).
Canceling method	All interrupts. Any reset available in the mode.	Interrupts shown in Table 10.4 Any reset available in the mode.
State after cancellation by an interrupt	Program execution state	Program execution state
State after cancellation by a reset	Reset state	Reset state
CPU	Stop (Retained)	Stop (Retained)

Table 10.2 Operating state of processor low power mode (2 of 2)

Item	CPU Sleep mode	CPU Deep Sleep mode
DMA Controller (DMAC)	Selectable	Selectable
Data Transfer Controller (DTC)	Selectable	Selectable
Watchdog Timer (WDT)	Selectable*1	Selectable*1
Independent Watchdog Timer (IWDT)	Selectable*1	Selectable*1
ARM Debug function	Stop*2	Stop*2
Trace function	Stop*3	Stop*3
Other peripheral modules	—	—

Note: Selectable means that operating or not operating can be selected by setting the control registers before entering Processor Low Power Mode.
Stop (Retained) means that the contents of the internal registers are retained but the operations are suspended.
Stop (Undefined) means that the contents of the internal registers are undefined and power to the internal circuit is cut off.
" — " means that the operation is not affected.

Note 1. In IWDT, operating or stopping is selected by setting the IWDT Stop Control bit (IWDTSTPCTL) in Option Function Select register 0 (OFS0) in IWDT auto start mode.
In WDT, operating or stopping is selected by setting the WDT Stop Control bit (WDTSTPCTL) in Option Function Select Register 0 (OFS0) in WDT auto start mode. Also, operating or stopping is selected by setting the WDT.WDTCSTPR.SLCSTP bit in WDT register start mode.

Note 2. This function is operating if debugger is connecting (DBGSTR.CDBGPWRUPREQ = 1 and SYOCDCR.DBGEN = 1) before entering this mode.

Note 3. This function is operating if debugger is connecting (DBGSTR.CDBGPWRUPREQ = 1 and SYOCDCR.DBGEN = 1) and TRCKCR.TRCKEN = 1 before entering this mode.

Note 4. Valid interrupt requests are any interrupt/exception that are not masked by the priority level of current exception and the priority level set by BASEPRI. In addition, if the interrupt request is based on IELSRn, the interrupt must be enabled by NVIC_ISERn.

Table 10.3 Operating state of each low power mode (1 of 3)

Item	Software Standby Mode(SSTBY)	Deep Software Standby mode(DSTBY)		
	SSTBY	DSTBY1	DSTBY2	DSTBY3
Transition condition	When [Condition 1] or [Condition 2] while CPU.SCR.SLEEPDEEP = 1 and LPSCR.LPMD is 0x4 [Condition 1] • WFI instruction • A valid interrupt request*22 cannot be accepted to CPU (including a transition from the time WFI instruction is executed to the time the transition to CPU Deep Sleep mode is completed). [Condition 2] • CPU.SCR.SLEEPONEXIT = 1 • Complete execution of all exception handlers • A valid interrupt request*22 cannot be accepted to CPU (including a transition from the time WFI instruction is executed to the time the transition to CPU Deep Sleep mode is completed).	When [Condition 1] or [Condition 2] while CPU.SCR.SLEEPDEEP = 1 and LPSCR.LPMD is either 0x8 or, 0x9 or 0xA. [Condition 1] • WFI instruction • A valid interrupt request*22 cannot be accepted to CPU (including a transition from the time WFI instruction is executed to the time the transition to CPU Deep Sleep mode is completed). [Condition 2] • CPU.SCR.SLEEPONEXIT = 1 • Complete execution of all exception handlers • A valid interrupt request*22 cannot be accepted to CPU (including a transition from the time WFI instruction is executed to the time the transition to CPU Deep Sleep mode is completed).		
Canceling method	Interrupts shown in Table 10.4 Any reset available in the mode	Interrupts shown in Table 10.4 Any reset available in the mode		
State after cancellation by an interrupt	Program execution state	Reset state		
State after cancellation by a reset	Reset state	Reset state		
Main clock oscillator	Selectable*10	Stop		
Sub-clock oscillator	Selectable	Selectable		
High-speed on-chip oscillator	Selectable*11	Stop		
Middle-speed on-chip oscillator	Stop*19	Stop		

Table 10.3 Operating state of each low power mode (2 of 3)

Item	Software Standby Mode(SSTBY)	Deep Software Standby mode(DSTBY)		
	SSTBY	DSTBY1	DSTBY2	DSTBY3
Low-speed on-chip oscillator	Selectable*2	Selectable*2	Stop	
PLL1	Stop	Stop		
PLL2	Stop	Stop		
Oscillation stop detection function	Selectable*12	Stop		
Clock/buzzer output function	Selectable*3	Stop (Undefined)		
External Bus (EBCLK)	Stop (Retained)	Stop (Retained)		
CPU	Stop (Retained)	Stop (Undefined)		
TCM (SRAM)	Stop (Retained)*15	Stop (Undefined)		
User SRAM	Stop (Retained)*13	Stop (Undefined)		
Standby SRAM	Stop (Retained)*14	Stop (Retained)*14	Stop (Undefined)	
Backup register	Stop (Retained)	Stop (Retained)		
Flash memory	Stop (Retained)	Stop (Retained)		
Memory Protection Unit (MPU)	Stop (Retained)	Stop (Undefined)		
DMA Controller (DMAC)	Stop (Retained)	Stop (Undefined)		
Data Transfer Controller (DTC)	Stop (Retained)	Stop (Undefined)		
Watchdog Timer (WDT)	Stop (Retained)	Stop (Undefined)		
Independent Watchdog Timer (IWDT)	Selectable*1	Selectable*1	Stop (Undefined)	
ARM Debug function	Stop*16	Stop*16		
Trace function	Stop*17	Stop*17		
Clock Frequency Accuracy Measurement Circuit (CAC)	Stop (Undefined)	Stop (Undefined)		
Ethernet MAC Controller (ETHERC)	Stop (Undefined)	Stop (Undefined)		
Ethernet DMA Controller (EDMAC)	Stop (Undefined)	Stop (Undefined)		
USB 2.0 Full-Speed Module (USBFS)	Stop (Retained) Detection of USB resumption is possible.	Stop (Retained) Detection of USB resumption is possible.	Stop (Undefined)	
USB 2.0 High-Speed Module (USBHS)	Stop (Retained) Detection of USB resumption is possible.	Stop (Retained) Detection of USB resumption is possible.	Stop (Undefined)	
Realtime clock (RTC)	Selectable	Selectable	Selectable*4	
CAN-FD	Stop (Undefined)	Stop (Undefined)		
CANFD ECC (CNECC)	Stop (Undefined)	Stop (Undefined)		
Serial Peripheral Interface (SPI0)	Stop (Retained)	Stop (Undefined)		
Serial Peripheral Interface (SPI1)	Stop (Undefined)	Stop (Undefined)		
Octa Serial Peripheral Interface (OSPI)	Stop (Retained)	Stop (Undefined)		
Decryption On The Fly (DOTF)	Stop (Retained)	Stop (Undefined)		
Serial Sound Interface Enhanced (SSIE0)	Stop (Retained)	Stop (Undefined)		
Serial Sound Interface Enhanced (SSIE1)	Stop (Undefined)	Stop (Undefined)		
SD/MMC Host Interface (SDHI0)	Stop (Retained)	Stop (Undefined)		

Table 10.3 Operating state of each low power mode (3 of 3)

Item	Software Standby Mode(SSTBY)	Deep Software Standby mode(DSTBY)		
	SSTBY	DSTBY1	DSTBY2	DSTBY3
SD/MMC Host Interface (SDHI1)	Stop (Undefined)	Stop (Undefined)		
Cyclic Redundancy Check (CRC) Calculator	Stop (Undefined)	Stop (Undefined)		
Port Output Enable for GPT (POEG)	Stop (Undefined)	Stop (Undefined)		
General PWM Timer (GPT)	Stop (Undefined)	Stop (Undefined)		
Ultra low power Timer (ULPTn, n = 0, 1)	Selectable	Selectable	Stop (Undefined)	
Low Power Asynchronous General Purpose Timer (AGTn, n = 0, 1)	Selectable*5	Stop (Undefined)		
12-Bit A/D Converter (ADC12)	Stop (Undefined)	Stop (Undefined)		
12-Bit D/A Converter (DAC12)	Stop (Retained)	Stop (Undefined)		
Data Operation Circuit (DOC)	Stop (Undefined)	Stop (Undefined)		
Serial Communications Interface (SCI0)	Stop (Retained)	Stop (Undefined)		
Serial Communications Interface (SCIn, n = 1 to 4, 9)	Stop (Undefined)	Stop (Undefined)		
I2C Bus Interface (IIC0)	Selectable*6	Stop (Undefined)		
I2C Bus Interface (IIC1)	Stop (Undefined)	Stop (Undefined)		
I3C Bus Interface (I3C)	Selectable*21	Stop (Undefined)		
Event Link Controller (ELC)	Stop (Undefined)	Stop (Undefined)		
Renesas Secure IP (RSIP-E51A)	Stop (Retained)	Stop (Undefined)		
Capture Engine Unit(CEU)	Stop (Undefined)	Stop (Undefined)		
2D Drawing Engine (DRW)	Stop (Undefined)	Stop (Undefined)		
Graphics LCD Controller (GLCDC)	Stop (Undefined)	Stop (Undefined)		
MIPI DSI	Stop (Undefined)	Stop (Undefined)		
Temperature Sensor (TSN)	Stop (Undefined)	Stop (Undefined)		
High-Speed Analog Comparator 0 (ACMPHS0)	Selectable	Stop (Undefined)		
High-Speed Analog Comparator 1 (ACMPHS1)	Selectable *20	Stop (Undefined)		
IRQn (n = 0 to 15) pin interrupt	Selectable	Stop (Undefined)		
NMI, IRQn-DS (n = 0 to 15) pin interrupt	Selectable	Selectable		
Programmable Voltage Detect (PVD)	Selectable	Selectable*18	Selectable*18	Stop (Undefined)*7
VBATT_R voltage drop detection	Selectable	Selectable	Selectable	Stop (Undefined)
Power-on reset circuit	Operating	Operating	Operating	Operating*8
I/O Ports	Retained*9	Retained*9		

Note: Selectable means that operating or not operating can be selected by setting the control registers before low power mode.
Stop (Retained) means that the contents of the internal registers are retained but the operations are suspended.
Stop (Undefined) means that the contents of the internal registers are undefined and power to the internal circuit is cut off. After canceling from each low power mode, the internal registers are initialized.

Note 1. In IWDT, operating or stopping is selected by setting the IWDT Stop Control bit (IWDTSTPCTL) in Option Function Select register 0 (OFS0) in IWDT auto start mode. Also, operating or stopping is selected by setting the IWDT.IWDTCSTPR.SLCSTP bit in IWDT register start mode.

Note 2. If IWDT is not used and LOCOCR.LCSTP = 0, LOCO is not stopped in Software Standby mode or Deep Software Standby mode 1. If IWDT is used and IWDT Stop Control bit is 0 (OFS.IWDTSTPCTL = 0 or IWDT.IWDTCSTPR.SLCSTP = 0), LOCO is not stopped regardless the value of LOCOCR.LCSTP in Software Standby mode or Deep Software Standby mode 1.

If IWDT is used and IWDT Stop Control bit is 1 (OFS.IWDTSTPCTL = 1 or IWDT.IWDTCSTPR.SLCSTP = 1), LOCO is not stopped with LOCOCR.LCSTP = 0 in Software Standby mode or Deep Software Standby mode 1.
In other cases, LOCO is stopped in Software Standby mode or Deep Software Standby mode 1.

Note 3. Stopped when the clock output source select bits (CKOCR.CKOSEL[2:0]) are set to a value other than 010b (LOCO) and 100b (SOSC).

Note 4. Only sub clock oscillator can be selected as the count source clock for RTC. When the RCR4.RCKSEL bit set to 1 (LOCO), the LPSCR must set to Deep Software Standby mode 1 before entering Deep Software Standby mode.

Note 5. AGT0 operation is possible when 100b (AGTLCLK) or 110b (AGTSCLK) is selected by the AGT0.AGTMR1.TCK[2:0] bits.
AGT1 operation is possible when 100b (AGTLCLK), 110b (AGTSCLK) or 101b (Underflow event signal from AGT0) is selected by the AGT1.AGTMR1.TCK[2:0] bits.
keeping the output state or changing to the

Note 6. Only IIC0 wakeup function is available.

Note 7. When using PVD in Deep Software Standby mode, LPSCR must be set to DSTBY1 or DSTBY2 before entering Deep Software Standby mode.

Note 8. When the MCU enters Deep Software Standby mode 3, the PVD circuit stops and the low-power function of the power-on reset circuit is enabled.

Note 9. For SRAM: [A00 to A23, CS0 to CS7, RD, WR0 to WR3, WR, BC0 to BC3 and ALE], and for SDRAM: [A00 to A16, DQM0 to DQM3, SDCS, RAS, CAS, WE and CKE] .

Note 10. When MOSCSCR.MOSCSOKP = 1 while MOSC is oscillated, MOSC continues to oscillate in Software Standby mode.

Note 11. When HOCOSCR.HOCOSOKP = 1 while HOCO is oscillated, HOCO continues to oscillate in Software Standby mode.

Note 12. The function is depending on the setting value of the MOSCSCR.MOSCSOKP bit.

Note 13. If PDRAMSCR0.RKEEPn bit is set to 0, the contents of the target User SRAM is not retained.

Note 14. If DPSBYCR.SRKEEP bit is set to 0, the contents of Standby RAM is not retained.

Note 15. If PDRAMSCR1.RKEEPn bit is set to 0, the contents of the target TCM is not retained.

Note 16. This function is operating if debugger is connecting before entering this mode.

Note 17. This function is operating if debugger is connecting and TRCKCR.TRCKEN = 1 before entering this mode.

Note 18. When OFS1(_SEC).PVDAS = 0 and OFS1(_SEC).PVDLPSEL = 0, low power consumption function of PVD0 is enabled during DSTBY1 and DSTBY2. When OFS1(_SEC).PVDAS = 0 and OFS1(_SEC).PVDLPSEL = 1, low power consumption function of PVD0 is disabled during DSTBY1 and DSTBY2. See section 60, Electrical Characteristics.

Note 19. When the on-chip debugger function is valid, MOCO does not stop in this mode.

Note 20. Only VCOUT function is permitted. The VCOUT pin operates when ACMPHS uses no digital filter. For details on digital filter, see section 48, High-Speed Analog Comparator (ACMPHS).

Note 21. Only I3C wakeup function is available.

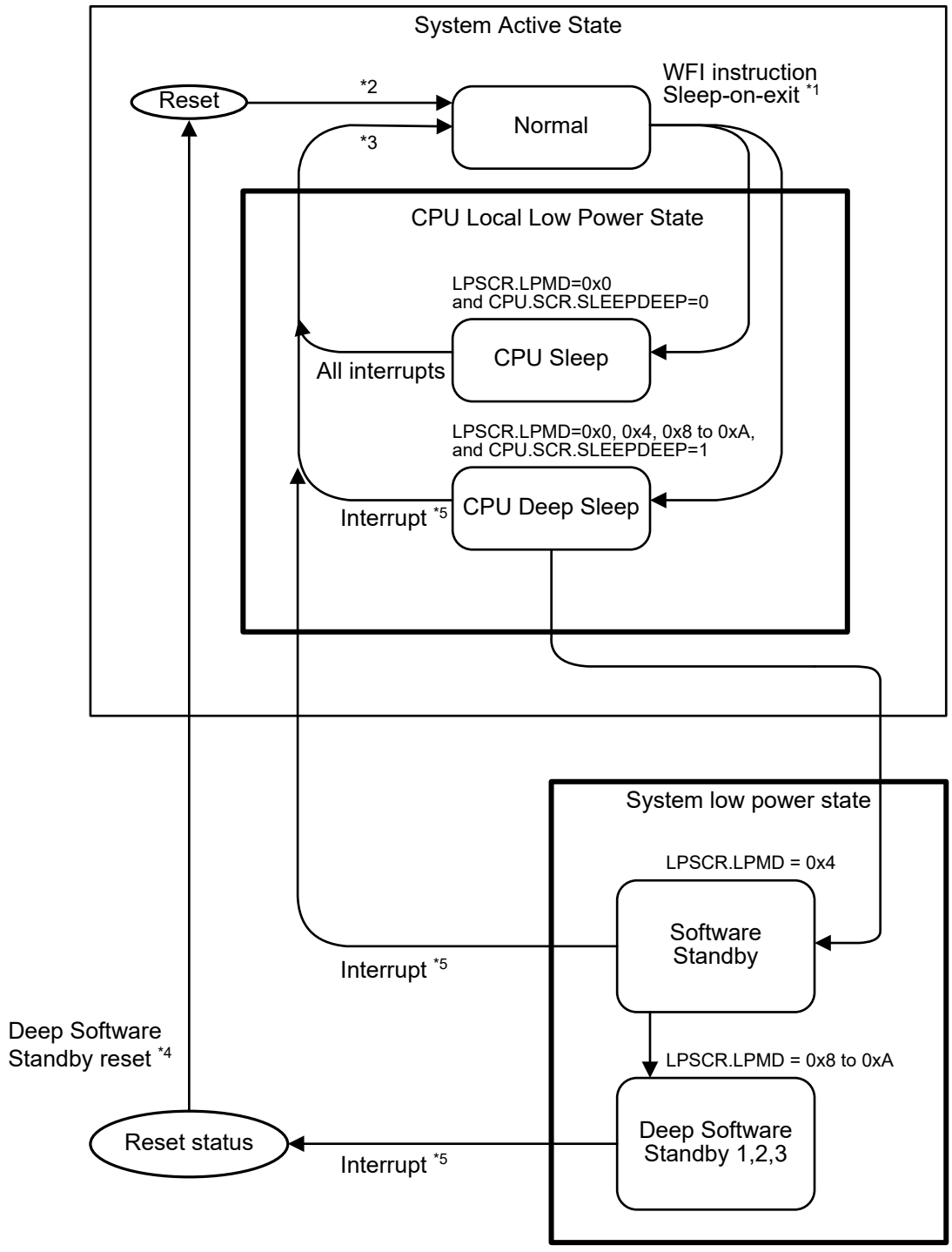
Note 22. Valid interrupt requests are any interrupt/exception that are not masked by the priority level of current exception and the priority level set by BASEPRI. In addition, if the interrupt request is based on IELSRn, the interrupt must be enabled by NVIC_ISERn.

Table 10.4 Interrupt source for canceling CPU Deep Sleep, Software Standby and Deep Software Standby modes (1 of 2)

Interrupt source	Name	CPU Deep Sleep Mode	Software Standby Mode	Deep Software Standby mode		
				DSTBY1	DSTBY2	DSTBY3
NMI		Yes	Yes	Yes	Yes	Yes
Port	PORT_IRQn (n = 0 to 15)	Yes	Yes	No	No	No
	PORT_IRQn-DS (n = 0 to 15)	Yes	Yes	Yes	Yes	Yes
PVD	PVD_PVDm (m = 1, 2)	Yes	Yes	Yes	Yes	No
IWDT	IWDT_NMIUNDF	Yes	Yes	Yes	No	No
USBFS	USBFS_USBR	Yes	Yes	Yes	No	No
USBHS	USBHS_USBIR	Yes	Yes	Yes	No	No
RTC	RTC_ALM	Yes	Yes	Yes	Yes	Yes
	RTC_PRD	Yes	Yes	Yes	Yes	Yes
ULPT0	ULPT0_ULPTI	Yes	Yes	Yes	No	No
	ULPT0_ULPTCMAI	Yes	Yes	No	No	No
	ULPT0_ULPTCMBI	Yes	Yes	No	No	No
ULPT1	ULPT1_ULPTI	Yes	Yes	Yes	No	No
	ULPT1_ULPTCMAI	Yes	Yes	No	No	No
	ULPT1_ULPTCMBI	Yes	Yes	No	No	No

Table 10.4 Interrupt source for canceling CPU Deep Sleep, Software Standby and Deep Software Standby modes (2 of 2)

Interrupt source	Name	CPU Deep Sleep Mode	Software Standby Mode	Deep Software Standby mode		
				DSTBY1	DSTBY2	DSTBY3
AGT1	AGT1_AGTI	Yes	Yes	No	No	No
	AGT1_AGTCMAI	Yes	Yes	No	No	No
	AGT1_AGTCMBI	Yes	Yes	No	No	No
ACMPHS0	ACMP_HS0	Yes	Yes	No	No	No
IIC0	IIC0_WUI	Yes	Yes	No	No	No
I3C	I3C_WU	Yes	Yes	No	No	No
BBF (Battery Backup Function)	VBATT_TADI	Yes	Yes	Yes	Yes	Yes



Note 1. When an interrupt that acts as a trigger for cancellation of CPU Sleep mode state is received during a transition to the CPU Sleep mode state or CPU Deep Sleep mode state, the MCU does not transit to CPU Sleep mode state or CPU Deep Sleep mode state but goes back to Normal mode state.
Note 2. The MOCO clock is the source of the operating clock following a transition from the reset state to Normal mode.
Note 3. When the transition to Normal mode is done because of an interrupt from CPU Sleep mode, CPU Deep Sleep mode or Software Standby mode, the clock source is the same as before entering the each low power modes.
Note 4. When an available interrupt request is generated, an internal reset (Deep Software Standby reset) is generated over a fixed period. Canceling of Deep Software Standby mode accompanies release from the internal reset state, and then the MCU transitions to Normal mode and executes reset exception processing with the MOCO clock as the source of the operating clock.
Note 5. For interrupt sources for canceling each low power modes, see Table 10.4

Figure 10.1 Mode transitions

10.2 Register Descriptions

10.2.1 LPMSAR : Low Power Mode Security Attribution Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x3C8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	NONSEC21	—	NONSEC19	NONSEC18	NONSEC17	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	NONSEC8	—	—	—	—	—	NONSEC2	NONSEC1	NONSEC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NONSEC0	Non-secure Attribute bit 0 Target register: OPCCR 0: Secure 1: Non-secure	R/W
1	NONSEC1	Non-secure Attribute bit 1 Target register: LPSCR, DPSWCR 0: Secure 1: Non-secure	R/W
2	NONSEC2	Non-secure Attribute bit 2 Target register: SBYCR, SSCR1 0: Secure 1: Non-secure	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	NONSEC8	Non-secure Attribute bit 8 Target register: DPSBYCR 0: Secure 1: Non-secure	R/W
16:9	—	These bits are read as 0. The write value should be 0.	R/W
17	NONSEC17	Non-secure Attribute bit 17 Target register: HOCOLDOCR 0: Secure 1: Non-secure	R/W
18	NONSEC18	Non-secure Attribute bit 18 Target register: PLL1LDOCR 0: Secure 1: Non-secure	R/W
19	NONSEC19	Non-secure Attribute bit 19 Target register: PLL2LDOCR 0: Secure 1: Non-secure	R/W
20	—	This bit is read as 0. The write value should be 0.	R/W
21	NONSEC21	Non-secure Attribute bit 21 Target register: LVOCR 0: Secure 1: Non-secure	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1
Note: Set the PRCR.PRC4 bit to 1 (write enabled) before rewriting this register.

The LPMSAR register controls the secure attribute of Low Power Mode registers.

NONSEC0 bit (Non-secure Attribute bit 0)

This bit controls the security attribute of OPCCR.

NONSEC1 bit (Non-secure Attribute bit 1)

This bit controls the security attribute of LPSCR, DPSECR.

NONSEC2 bit (Non-secure Attribute bit 2)

This bit controls the security attribute of SBYCR, SSCR1.

NONSEC8 bit (Non-secure Attribute bit 8)

This bit controls the security attribute of DPSBYCR.

NONSEC17 bit (Non-secure Attribute bit 17)

This bit controls the security attribute of HOCOLDOCR.

NONSEC18 bit (Non-secure Attribute bit 18)

This bit controls the security attribute of PLL1LDOCR.

NONSEC19 bit (Non-secure Attribute bit 19)

This bit controls the security attribute of PLL2LDOCR.

NONSEC21 bit (Non-secure Attribute bit 21)

This bit controls the security attribute of LVOCR.

10.2.2 PGCSAR : Power Gating Control Security Attribution Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x3D8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	NONSEC1	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	NONSEC1	Non-secure Attribute bit 1 Target register: PDCTRGD 0: Secure 1: Non-secure	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1
Note: Set the PRCR.PRC4 bit to 1 (write enabled) before rewriting this register.

The PGCSAR register controls the secure attribute of power gating registers.

NONSEC1 bit (Non-secure Attribute bit 1)

This bit controls the security attribute of PDCTRGD.

10.2.3 DPFSAR : Deep Software Standby Interrupt Factor Security Attribution Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x3E0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DPFSA31	—	DPFSA29	—	DPFSA27	DPFSA26	DPFSA25	DPFSA24	—	—	—	DPFSA20	DPFSA19	DPFSA18	DPFSA17	DPFSA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DPFSA15	DPFSA14	DPFSA13	DPFSA12	DPFSA11	DPFSA10	DPFSA9	DPFSA8	DPFSA7	DPFSA6	DPFSA5	DPFSA4	DPFSA3	DPFSA2	DPFSA1	DPFSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DPFSA7 to DPFSA0	Deep Software Standby Interrupt Factor Security Attribute bit n (n = 0 to 7) Target register: DPSIER0.bn, DPSIFR0.bn, DPSIEGR0.bn (n = 0 to 7) Target factor : IRQn-DS Pin (n = 0 to 7) 0: Secure 1: Non-secure	R/W
15:8	DPFSA15 to DPFSA8	Deep Software Standby Interrupt Factor Security Attribute bit n (n = 8 to 15) Target register: DPSIER1.bn, DPSIFR1.bn, DPSIEGR1.bn (n = 0 to 7) Target factor : IRQn-DS Pin (n = 8 to 15) 0: Secure 1: Non-secure	R/W
16	DPFSA16	Deep Software Standby Interrupt Factor Security Attribute bit 16 Target register: DPSIER2.b0, DPSIFR2.b0, DPSIEGR2.b0 Target factor : PVD1 0: Secure 1: Non-secure	R/W
17	DPFSA17	Deep Software Standby Interrupt Factor Security Attribute bit 17 Target register: DPSIER2.b1, DPSIFR2.b1, DPSIEGR2.b1 Target factor : PVD2 0: Secure 1: Non-secure	R/W
18	DPFSA18	Deep Software Standby Interrupt Factor Security Attribute bit 18 Target register: DPSIER2.b2, DPSIFR2.b2 Target factor : RTC Interval 0: Secure 1: Non-secure	R/W
19	DPFSA19	Deep Software Standby Interrupt Factor Security Attribute bit 19 Target register: DPSIER2.b3, DPSIFR2.b3 Target factor : RTC Alarm 0: Secure 1: Non-secure	R/W
20	DPFSA20	Deep Software Standby Interrupt Factor Security Attribute bit 20 Target register: DPSIER2.b4, DPSIFR2.b4, DPSIEGR2.b4 Target factor : NMI Pin 0: Secure 1: Non-secure	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	DPFSA24	Deep Software Standby Interrupt Factor Security Attribute bit 24 Target register: DPSIER3.b0, DPSIFR3.b0 Target factor : USBFS Suspend/Resume 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
25	DPFSA25	Deep Software Standby Interrupt Factor Security Attribute bit 25 Target register: DPSIER3.b1, DPSIFR3.b1 Target factor : USBHS Suspend/Resume 0: Secure 1: Non-secure	R/W
26	DPFSA26	Deep Software Standby Interrupt Factor Security Attribute bit 26 Target register: DPSIER3.b2, DPSIFR3.b2 Target factor : ULPT0 0: Secure 1: Non-secure	R/W
27	DPFSA27	Deep Software Standby Interrupt Factor Security Attribute bit 27 Target register: DPSIER3.b3, DPSIFR3.b3 Target factor : ULPT1 0: Secure 1: Non-secure	R/W
28	—	This bit is read as 0. The write value should be 0.	R/W
29	DPFSA29	Deep Software Standby Interrupt Factor Security Attribute bit 29 Target register: DPSIER3.b5, DPSIFR3.b5 Target factor : IWDT Underflow 0: Secure 1: Non-secure	R/W
30	—	This bit is read as 0. The write value should be 0.	R/W
31	DPFSA31	Deep Software Standby Interrupt Factor Security Attribute bit 31 Target register: DPSIER3.b7, DPSIFR3.b7 Target factor : Tamper Detection 0: Secure 1: Non-secure	R/W

Note: S-TYPE-1, P-TYPE-1
Note: Set the PRCR.PRC4 bit to 1 (write enabled) before rewriting this register.

The DPFSAR register controls the secure attribute of Deep Software Standby Interrupt Factor control registers.

DPFSA bit (Deep Software Standby Interrupt Factor Security Attribute bit n (n = 0 to 7))

This bit controls the security attribute of DPSIER0.bn, DPSIFR0.bn, DPSIEGR0.bn (n = 0 to 7) .

Target factor is IRQn-DS Pin (n = 0 to 7).

DPFSA bit (Deep Software Standby Interrupt Factor Security Attribute bit n (n = 8 to 15))

This bit controls the security attribute of DPSIER1.bn, DPSIFR1.bn, DPSIEGR1.bn (n = 0 to 7) .

Target factor is IRQn-DS Pin (n = 8 to 15).

DPFSA16 bit (Deep Software Standby Interrupt Factor Security Attribute bit 16)

This bit controls the security attribute of DPSIER2.b0, DPSIFR2.b0, DPSIEGR2.b0 .

Target factor is PVD1.

DPFSA17 bit (Deep Software Standby Interrupt Factor Security Attribute bit 17)

This bit controls the security attribute of DPSIER2.b1, DPSIFR2.b1, DPSIEGR2.b1.

Target factor is PVD2.

DPFSA18 bit (Deep Software Standby Interrupt Factor Security Attribute bit 18)

This bit controls the security attribute of DPSIER2.b2, DPSIFR2.b2.

Target factor is RTC Interval.

DPFSA19 bit (Deep Software Standby Interrupt Factor Security Attribute bit 19)

This bit controls the security attribute of DPSIER2.b3, DPSIFR2.b3.

Target factor is RTC Alarm.

DPFSA20 bit (Deep Software Standby Interrupt Factor Security Attribute bit 20)

This bit controls the security attribute of DPSIER2.b4, DPSIFR2.b4, DPSIEGR2.b4.

Target factor is NMI Pin.

DPFSA24 bit (Deep Software Standby Interrupt Factor Security Attribute bit 24)

This bit controls the security attribute of DPSIER3.b0, DPSIFR3.b0.

Target factor is USBFS Suspend/Resume.

DPFSA25 bit (Deep Software Standby Interrupt Factor Security Attribute bit 25)

This bit controls the security attribute of DPSIER3.b1, DPSIFR3.b1.

Target factor is USBHS Suspend/Resume.

DPFSA26 bit (Deep Software Standby Interrupt Factor Security Attribute bit 26)

This bit controls the security attribute of DPSIER3.b2, DPSIFR3.b2.

Target factor is ULPT0.

DPFSA27 bit (Deep Software Standby Interrupt Factor Security Attribute bit 27)

This bit controls the security attribute of DPSIER3.b3, DPSIFR3.b3.

Target factor is ULPT1.

DPFSA29 bit (Deep Software Standby Interrupt Factor Security Attribute bit 29)

This bit controls the security attribute of DPSIER3.b5, DPSIFR3.b5.

Target factor is IWDT Underflow.

DPFSA31 bit (Deep Software Standby Interrupt Factor Security Attribute bit 31)

This bit controls the security attribute of DPSIER3.b7, DPSIFR3.b7.

Target factor is Tamper Detection.

10.2.4 RSCSAR : RAM Standby Control Security Attribution Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x3E4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RSCSA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	RSCSA6	RSCSA5	RSCSA4	RSCSA3	RSCSA2	RSCSA1	RSCSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	RSCSA6 to RSCSA0	RAM Standby Control Security Attribute bit n (n = 0 to 6) Target register: PDRAMSCR0.bn (n = 0 to 6) 0: Secure 1: Non-secure	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W
16	RSCSA16	RAM Standby Control Security Attribute bit n (n = 16) Target register: PDRAMSCR1.bn (n = 0) 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1
Note: Set the PRCR.PRC4 bit to 1 (write enabled) before rewriting this register.

The RSCSAR register controls the secure attribute of RAM Standby Control registers.

RSCSAn (n = 0 to 6) bit (RAM Standby Control Security Attribute bit n)

This bit controls the security attribute of PDRAMSCR0.bn (n = 0 to 6).

RSCSAn (n = 16) bit (RAM Standby Control Security Attribute bit n)

This bit controls the security attribute of PDRAMSCR1.bn (n = 0).

10.2.5 MSTPCRA : Module Stop Control Register A

Base address: MSTP = 0x4020_3000
MSTP_NS = 0x5020_3000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	MSTPA22	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MSTPA15	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTPA1	MSTPA0
Value after reset:	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0

Bit	Symbol	Function	R/W
0	MSTPA0	SRAM0 Module Stop*2 Target module: SRAM0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
1	MSTPA1	SRAM1 Module Stop*2 Target module: SRAM1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
14:2	—	These bits are read as 1. The write value should be 1.	R/W
15	MSTPA15	Standby SRAM Module Stop Target module: Standby SRAM 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21:16	—	These bits are read as 1. The write value should be 1.	R/W
22	MSTPA22	DMA Controller and Data Transfer Controller Module Stop*1 Target module: DMAC, DTC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31:23	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-4, P-TYPE-2
Note 1. When rewriting the MSTPA22 bit from 0 to 1, disable the DMAC and DTC before setting the MSTPA22 bit.
Note 2. When changing the value of this bit, only execute subsequent instructions after reading this bit to check that the value was updated.

10.2.6 MSTPCRB : Module Stop Control Register B

Base address: MSTP = 0x4020_3000
MSTP_NS = 0x5020_3000

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSTPB31	MSTPB30	MSTPB29	MSTPB28	MSTPB27	—	—	—	—	MSTPB22	—	—	MSTPB19	MSTPB18	—	MSTPB16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MSTPB15	—	—	MSTPB12	MSTPB11	—	MSTPB9	MSTPB8	—	—	—	MSTPB4	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 1. The write value should be 1.	R/W
4	MSTPB4	I3C Bus Interface Module Stop[*1] Target module: I3C 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
7:5	—	These bits are read as 1. The write value should be 1.	R/W
8	MSTPB8	I^2C Bus Interface 1 Module Stop Target module: IIC1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
9	MSTPB9	I^2C Bus Interface 0 Module Stop Target module: IIC0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
10	—	This bit is read as 1. The write value should be 1.	R/W
11	MSTPB11	Universal Serial Bus 2.0 FS Interface Module Stop[*2] Target module: USBFS 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
12	MSTPB12	Universal Serial Bus 2.0 HS Interface Module Stop[*2] Target module: USBHS 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
14:13	—	These bits are read as 1. The write value should be 1.	R/W
15	MSTPB15	ETHERC0 and EDMAC0 Controller Module Stop Target module: ETHERC0 and EDMAC0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
16	MSTPB16	Octal Serial Peripheral Interface and Decryption On The Fly Module Stop[*3] Target module: OSPI and DOTF 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
17	—	This bit is read as 1. The write value should be 1.	R/W
18	MSTPB18	Serial Peripheral Interface 1 Module Stop[*4] Target module: SPI1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Bit	Symbol	Function	R/W
19	MSTPB19	Serial Peripheral Interface 0 Module Stop*4 Target module: SPI0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21:20	—	These bits are read as 1. The write value should be 1.	R/W
22	MSTPB22	Serial Communication Interface 9 Module Stop*5 Target module: SCI9 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
26:23	—	These bits are read as 1. The write value should be 1.	R/W
27	MSTPB27	Serial Communication Interface 4 Module Stop*5 Target module: SCI4 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
28	MSTPB28	Serial Communication Interface 3 Module Stop*5 Target module: SCI3 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
29	MSTPB29	Serial Communication Interface 2 Module Stop*5 Target module: SCI2 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
30	MSTPB30	Serial Communication Interface 1 Module Stop*5 Target module: SCI1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31	MSTPB31	Serial Communication Interface 0 Module Stop*5 Target module: SCI0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Note: S-TYPE-4, P-TYPE-4

Note 1. The MSTPB4 bit should be written after the I3C clock (I3CCLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of I3CCLK following the change of MSTPB4.

Note 2. The MSTPBi bit should be written after the USB clock (USBCLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of USBCLK following the change of MSTPBi (i = 11 to 12).

Note 3. The MSTPB16 bit should be written after the Octal-SPI clock (OCTACLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of OCTACLK following the change of MSTPB16. When changing the value of this bit, only execute subsequent instructions after reading this bit to check that the value was updated.

Note 4. The MSTPBi bit should be written after the SPI clock (SPICLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of SPICLK following the change of MSTPBi (i = 18 to 19).

Note 5. The MSTPBi bit should be written after the SCI clock (SCICLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of SCICLK following the change of MSTPBi (i = 22, 27 to 31).

10.2.7 MSTPCRC : Module Stop Control Register C

Base address: MSTP = 0x4020_3000
MSTP_NS = 0x5020_3000

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSTPC31	—	—	—	MSTPC27	MSTPC26	—	—	—	—	—	—	—	—	—	MSTPC16
Value after reset:	1/0*4	1	1	1	1/0*4	1/0*4	1	1	1	1	1	1	1	1	1	1/0*4

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MSTPC14	MSTPC13	MSTPC12	MSTPC11	MSTPC10	—	MSTPC8	MSTPC7	MSTPC6	—	MSTPC4	—	—	MSTPC1	MSTPC0
Value after reset:	1	1/0*4	1/0*4	1/0*4	1/0*4	1/0*5	1	1/0*4	1/0*4	1/0*4	1	1	1	1	1/0*4	1/0*4

Bit	Symbol	Function	R/W
0	MSTPC0	Clock Frequency Accuracy Measurement Circuit Module Stop[*1] Target module: CAC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
1	MSTPC1	Cyclic Redundancy Check Calculator Module Stop Target module: CRC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
3:2	—	These bits are read as 1. The write value should be 1.	R/W
4	MSTPC4	Graphics LCD Controller Module Stop[*2] Target module: GLCDC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
5	—	This bit is read as 1. The write value should be 1.	R/W
6	MSTPC6	2D Drawing Engine Module Stop Target module: DRW 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
7	MSTPC7	Serial Sound Interface Enhanced 1 Module Stop Target module: SSIE1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
8	MSTPC8	Serial Sound Interface Enhanced 0 Module Stop Target module: SSIE0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
9	—	This bit is read as 1. The write value should be 1.	R/W
10	MSTPC10	MIPI Display Serial Interface Module Stop Target module: MIPI DSI 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
11	MSTPC11	Secure Digital Host IF/Multi Media Card 1 Module Stop Target module: SDHI/MMC1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
12	MSTPC12	Secure Digital Host IF/Multi Media Card 0 Module Stop Target module: SDHI/MMC0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
13	MSTPC13	Data Operation Circuit Module Stop Target module: DOC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
14	MSTPC14	Event Link Controller Module Stop Target module: ELC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
15	—	This bit is read as 1. The write value should be 1.	R/W
16	MSTPC16	Capture Engine Unit Module Stop Target module: CEU 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
25:17	—	These bits are read as 1. The write value should be 1.	R/W
26	MSTPC26	Controller Area Network with Flexible Data-Rate 1 Module Stop[*3] Target module: CANFD1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Bit	Symbol	Function	R/W
27	MSTPC27	Controller Area Network with Flexible Data-Rate 0 Module Stop*3 Target module: CANFD0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
30:28	—	These bits are read as 1. The write value should be 1.	R/W
31	MSTPC31	Renesas Secure IP Module Stop Target module: RSIP-E51A 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Note: S-TYPE-4, P-TYPE-4
Note 1. The MSTPC0 bit should be written after the CAC clock (CACMCLK/CACSCLK/CACLCLK/CACMOCLK/CACHCLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of CACMCLK/CACSCLK/CACLCLK/CACMOCLK/CACHCLK following the change of MSTPC0.
Note 2. The MSTPC4 bit should be written after the GLCDC clock (GLCDCCLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of GLCDCCLK following the change of MSTPC4.
Note 3. The MSTPCi bit should be written after the CANFD clock (CANFDCLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of CANFDCLK following the change of MSTPCi (i = 26 to 27).
Note 4. If FSBL is skipped in reset, then value after reset is 1, if FSBL is executed in reset, then value after reset is 0.
Note 5. If FSBL is skipped in reset, then MSTPC10 value after reset is 1. If FSBL is executed in reset, then the value after reset depends on the parts number which supports MIPI DSI or not. If FSBL is executed in reset, then value after reset is 0 for R7FA8D1BHECBD, R7FA8D1BHECFC, R7FA8D1BFECBD, R7FA8D1BFECFC, which supports MIPI DSI. The value after reset is 1 for R7FA8D1AHECBD, R7FA8D1AHECFC, R7FA8D1AFECBD, R7FA8D1AFECFC which does not support MIPI DSI.

10.2.8 MSTPCRD : Module Stop Control Register D

Base address: MSTP = 0x4020_3000
MSTP_NS = 0x5020_3000

Offset address: 0x00C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	MSTPD28	MSTPD27	—	—	—	—	MSTPD22	—	MSTPD20	—	—	—	MSTPD16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MSTPD15	MSTPD14	MSTPD13	MSTPD12	MSTPD11	—	—	—	—	—	MSTPD5	MSTPD4	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 1. The write value should be 1.	R/W
4	MSTPD4	Low Power Asynchronous General Purpose Timer 1 Module Stop*1 Target module: AGT1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
5	MSTPD5	Low Power Asynchronous General Purpose Timer 0 Module Stop*1 Target module: AGT0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
10:6	—	These bits are read as 1. The write value should be 1.	R/W
11	MSTPD11	Port Output Enable for GPT Group D Module Stop Target module: POEG Group D 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
12	MSTPD12	Port Output Enable for GPT Group C Module Stop Target module: POEG Group C 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Bit	Symbol	Function	R/W
13	MSTPD13	Port Output Enable for GPT Group B Module Stop Target module: POEG Group B 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
14	MSTPD14	Port Output Enable for GPT Group A Module Stop Target module: POEG Group A 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
15	MSTPD15	12-bit A/D Converter 1 Module Stop Target module: ADC12_1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
16	MSTPD16	12-bit A/D Converter 0 Module Stop Target module: ADC12_0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
19:17	—	These bits are read as 1. The write value should be 1.	R/W
20	MSTPD20	12-bit D/A Converter Module Stop Target module: DAC12 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21	—	This bit is read as 1. The write value should be 1.	R/W
22	MSTPD22	Temperature Sensor Module Stop Target module: TSN 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
26:23	—	These bits are read as 1. The write value should be 1.	R/W
27	MSTPD27	High-Speed Analog Comparator 1 Module Stop Target module: ACMPHS1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
28	MSTPD28	High-Speed Analog Comparator 0 Module Stop Target module: ACMPHS0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31:29	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-4, P-TYPE-4

Note 1. When the count source is sub-clock oscillator or LOCO, AGTn counting does not stop even if MSTPDi is set to 1. If the count source is the sub-clock oscillator or LOCO, this bit must be set to 1 except when accessing the AGTn registers. (n =0,1) (i = 4, 5)

10.2.9 MSTPCRE : Module Stop Control Register E

Base address: MSTP = 0x4020_3000
MSTP_NS = 0x5020_3000

Offset address: 0x010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSTPE31	MSTPE30	MSTPE29	MSTPE28	MSTPE27	MSTPE26	MSTPE25	MSTPE24	MSTPE23	MSTPE22	MSTPE21	MSTPE20	MSTPE19	MSTPE18	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	MSTPE8	MSTPE8	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 1. The write value should be 1.	R/W
8	MSTPE8	Ultra-Low Power Timer 1 Module Stop[*1] Target module: ULPT1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
9	MSTPE9	Ultra-Low Power Timer 0 Module Stop[*1] Target module: ULPT0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
17:10	—	These bits are read as 1. The write value should be 1.	R/W
18	MSTPE18	General PWM Timer 13 Module[*2] Target module: GPT13 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
19	MSTPE19	General PWM Timer 12 Module[*2] Target module: GPT12 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
20	MSTPE20	General PWM Timer 11 Module[*2] Target module: GPT11 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21	MSTPE21	General PWM Timer 10 Module[*2] Target module: GPT10 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
22	MSTPE22	General PWM Timer 9 Module[*2] Target module: GPT9 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
23	MSTPE23	General PWM Timer 8 Module[*2] Target module: GPT8 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
24	MSTPE24	General PWM Timer 7 Module[*2] Target module: GPT7 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
25	MSTPE25	General PWM Timer 6 Module[*2] Target module: GPT6 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
26	MSTPE26	General PWM Timer 5 Module[*2] Target module: GPT5 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
27	MSTPE27	General PWM Timer 4 Module[*2] Target module: GPT4 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
28	MSTPE28	General PWM Timer 3 Module[*2] Target module: GPT3 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Bit	Symbol	Function	R/W
29	MSTPE29	General PWM Timer 2 Module*2 Target module: GPT2 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
30	MSTPE30	General PWM Timer 1 Module*2 Target module: GPT1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31	MSTPE31	General PWM Timer 0 Module*2 Target module: GPT0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Note: S-TYPE-4, P-TYPE-4
Note 1. When the count source is sub-clock oscillator or LOCO, ULPTn counting does not stop even if MSTPEi is set to 1. If the count source is the sub-clock oscillator or LOCO, this bit must be set to 1 except when accessing the ULPTn registers. (n = 1, 0) (i = 8, 9)
Note 2. The MSTPEi bit should be written after the GPT clock (GPTCLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of GPTCLK following the change of MSTPEi (i = 18 to 31).

10.2.10 SBYCR : Standby Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x00C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	OPE	—	—	—	—	—	—
Value after reset:	0	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
6	OPE	Output Port Enable 0: In Software Standby mode or Deep Software Standby mode, set the address bus and other bus control signal to the high-impedance state. 1: In Software Standby mode or Deep Software Standby mode, address bus and other bus control signal retain the output state.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The SBYCR register controls the bus signal output in Software Standby mode and Deep Software Standby mode.

OPE bit (Output Port Enable)

The OPE bit specifies whether to set to the high-impedance state or to retain the output of the address bus and bus control signals. For CSC: A00 to A23, CS0 to CS7, RD, WR0 to WR3, WR, BC0 to BC3 and ALE, and for SDRAMC: A00 to A16, DQM0 to DQM3, SDCS, RAS, CAS, WE and CKE in Software Standby mode or Deep Software Standby mode.

10.2.11 OPCCR : Operating Power Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0A0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	OPCM TSF	—	—	OPCM[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	OPCM[1:0]	Operating Power Control Mode Select 0 0: High-speed mode 0 1: Setting prohibited 1 0: Setting prohibited 1 1: Low-speed mode	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	OPCMTSF	Operating Power Control Mode Transition Status Flag 0: Transition completed 1: During transition	R
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The OPCCR register is used to reduce power consumption in Normal and Processor low power mode by specifying a lower operating frequency. For the procedure to change the operating power control modes, see section 10.5. Function for Lower Operating Power Consumption.

When transitioning from Software Standby mode to Normal mode, the setting in the OPCCR.OPCM[1:0] bits are as follows, regardless of the setting before entering Software Standby mode:

- OPCCR.OPCM[1:0] = 00b (High-speed mode)

If Software Standby mode is canceled by interrupt before the transition to Software Standby completes, the OPCCR.OPCM[1:0] bits retain the setting from before the WFI instruction is executed. If this causes any problem, set the MCU to High-speed mode during the exception handling procedure when canceling Software Standby mode.

OPCM[1:0] bits (Operating Power Control Mode Select)

The OPCM[1:0] bits select the operating power control mode in Normal and Processor low power mode. Table 10.5 shows the relationship between the operating power control modes and the OPCM[1:0] settings.

OPCMTSF flag (Operating Power Control Mode Transition Status Flag)

The OPCMTSF flag indicates the switching control state when the operating power control mode is switched. This flag becomes 1 when the OPCM bit is written, and 0 when mode transition completes. Read this flag and confirm that it is 0 before proceeding.

Table 10.5 The operating power control modes

Operating power control mode	OPCM[1:0] bits	Power consumption
High-speed mode	00b	High
Low-speed mode	11b	Low

Operating frequency range and voltage range are shown in Electrical characteristics.

Each operating power control mode is described below.

- High-speed mode
  After a reset cancellation, the MCU is activated in this mode.
- Low-speed mode
  The following restrictions apply in Low-speed mode:
  - P/E operations for flash memory are prohibited.
  - Using the PLL1/PLL2 is prohibited. See section 10.8.1. Register Access
  - Writing to PLL1LDOCR, PLL2LDOCR and HOCOLDOCR is prohibited.

  In this mode, lower power consumption is possible than in high-speed mode when the same operation is performed under the same conditions (operating frequency, operating voltage).

10.2.12 PDCTRGD : Graphics Power Domain Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x110

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PDPGSF	PDCSF	—	—	—	—	—	PDDE
Value after reset:	1	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	PDDE	Power control enable 0: Power on the target domain 1: Power off the target domain	R/W
5:1	—	These bits are read as 0. The write value should be 0.	R/W
6	PDCSF	Power control status flag 0: Power gating control is not executed (idle) 1: Power gating control is in progress	R
7	PDPGSF	Power gating status flag 0: Target domain is power on (not gating) 1: Target domain is power off (during Gating)	R

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The PDCTRGD register controls power gating for Graphics power domain.

For restrictions on this register, not only in this section , please also refer to section 10.6.1. Power Gating Control Using Power Domain Control Register.

PDDE bit (Power control enable)

The PDDE bit controls the power gating of the target domain. The PDDE bit should be set from 0 to 1, after confirmed that the PDCSF=0 and PDPGSF=0. On the other hand, the PDDE bit should be set from 1 to 0, after confirmed that the PDCSF=0 and PDPGSF=1.

PDCSF bit (Power control status flag)

The PDCSF bit indicates the power control status of the target domain.

PDPGSF bit (Power gating status flag)

The PDPGSF bit indicates whether the target domain is power gated.

10.2.13 PDRAMSCR0 : SRAM Power Domain Standby Control Register 0

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x140

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	RKEEP6	RKEEP5	RKEEP4	RKEEP3	RKEEP2	RKEEP1	RKEEP0
Value after reset:	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
6:0	RKEEP6 to RKEEP0	RAM Retention 0: When entering the Software Standby mode, the contents of the target RAM are not kept. 1: When entering the Software Standby mode, the contents of the target RAM are kept.	R/W

Bit	Symbol	Function	R/W
14:7	—	These bits are read as 1. The write value should be 1.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-4, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The PDRAMSCR0 register controls the target memory in Software Standby mode.

RKEEPn bit (RAM Retention) (n = 0 to 6)

The RKEEPn (n = 0 to 6) 7 bits controls whether to retain the contents of the target memory in Software Standby mode.

Table 10.6 shows the relationship between each bit and the target memory.

Table 10.6 Relationship between each bit and the target memory

PDRAMSCR0 register bit	Target memory
PDRAM.b0	SRAM0: 0x2200_0000 to 0x2201_FFFF (Secure alias) 0x3200_0000 to 0x3201_FFFF (Non-secure alias)
PDRAM.b1	SRAM0: 0x2202_0000 to 0x2203_FFFF (Secure alias) 0x3202_0000 to 0x3203_FFFF (Non-secure alias)
PDRAM.b2	SRAM0: 0x2204_0000 to 0x2205_FFFF (Secure alias) 0x3204_0000 to 0x3205_FFFF (Non-secure alias)
PDRAM.b3	SRAM1: 0x2206_0000 to 0x2207_FFFF (Secure alias) 0x3206_0000 to 0x3207_FFFF (Non-secure alias)
PDRAM.b4	SRAM1: 0x2208_0000 to 0x2209_FFFF (Secure alias) 0x3208_0000 to 0x3209_FFFF (Non-secure alias)
PDRAM.b5	SRAM1: 0x220A_0000 to 0x220B_FFFF (Secure alias) 0x320A_0000 to 0x320B_FFFF (Non-secure alias)
PDRAM.b6	SRAM1: 0x220C_0000 to 0x220D_FFFF (Secure alias) 0x320C_0000 to 0x320D_FFFF (Non-secure alias)

10.2.14 PDRAMSCR1 : SRAM Power Domain Standby Control Register 1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x142

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	RKEEP0
Value after reset:	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	RKEEP0	RAM Retention 0: When entering the CPU Deep Sleep and Software Standby mode, the contents of the target RAM are not kept. 1: When entering the CPU Deep Sleep and Software Standby mode, the contents of the target RAM are kept.	R/W
1	—	This bit is read as 1. The write value should be 1.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-4, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The PDRAMSCR1 register controls the target memory in CPU Deep Sleep and Software Standby mode.

RKEEP0 bit (RAM Retention)

The RKEEP0 bit controls whether to retain the contents of the target memory in CPU Deep Sleep and Software Standby mode.

Table 10.7 shows the relationship between each bit and the target memory.

Table 10.7 Relationship between each bit and the target memory

PDRAM register bit	Target memory
PDRAM.b0	ITCM and DTCM

10.2.15 SSCR1 : Software Standby Control Register 1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA98

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SS1FR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SS1FR	Software Standby Fast Return 0: When returning from Software Standby mode, fast return function is disabled 1: When returning from Software Standby mode, fast return function is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

SS1FR bit (Software Standby Fast Return)

When the SS1FR bit is 1, the recovery time from Software Standby mode is shorter.

Refer to the chapter of electrical characteristics on the detail of the recovery time.

Note: It is recommended to set this bit to 1. Otherwise it just increases the recovery time.

10.2.16 LPSCR : Low Power State Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA90

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	LPMD[3:0]			
Value after reset:	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
3:0	LPMD[3:0]	Low power mode setting bit 0x0: System Active 0x4: Software Standby mode 0x8: Deep Software Standby mode 1 0x9: Deep Software Standby mode 2 0xA: Deep Software Standby mode 3 Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The LPSCR register controls the transition destination of low power mode.

LPSCR is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 5, Resets.

LPMD[3:0] bit (Low power mode setting bit)

The LPMD[3:0] bit indicate the destination of low power mode. Actual low power mode transition is executed by WFI instruction in CPU.

The transition of Software Standby mode and Deep Software Standby mode does not occur unless the WFI instruction of each CPU is executed.

When the low power mode is canceled by an interrupt, The LPMD[3:0] bit remains the value before transition. Writing 0 to this bit clears the bit itself.

While FENTRYR.FENTRYi (i = 0 to 3) is 1 or FENTRYR.FENTRYD bit is 1, the setting to Software Standby and Deep Software Standby mode is ineffective. Even if LPMD is any of 0x4, 0x8 to 0xA, the MCU is keep to System active state on execution of a WFI instruction. Refer to Table 10.10 for the detail.

While OFS0.IWDTSTPCTL bit is 0 (counting continues), the setting to Deep software Standby 2 and 3 mode is ineffective. Even if LPMD is any of 0x9 or 0xA, the MCU enters Deep Software Standby 1 mode on execution of a WFI instruction.

While IWDTCSTPR.SLCSTP is 0 (counting continues), the setting to Deep software Standby 2 and 3 mode is ineffective. Even if LPMD is any of 0x9 or 0xA, the MCU enters Deep Software Standby 1 mode on execution of a WFI instruction.

Refer to Table 10.10 for the detail.

While the voltage monitor n (n=1, 2) reset is enabled (PVDnCR0.RI=1), the setting to Deep Software Standby mode 2, 3 is ineffective. Even if LPMD is any of 0x9 or 0xA the MCU enters Deep Software Standby 1 mode on execution of a WFI instruction. Refer to Table 10.10 for the detail.

10.2.17 DPSBYCR : Deep Software Standby Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	IOKEEP	—	SRKEEP	—	DCSSMODE	—	—
Value after reset:	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	DCSSMODE	DCDC Soft Start Mode 0: When the Deep Software Standby mode is canceled, the time required to recover is the standard time. 1: When the Deep Software Standby mode is canceled, the time required to recover is shortened.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	SRKEEP	Standby SRAM Retention 0: When entering the Software Standby mode or the Deep Software Standby mode 1, the contents of Standby SRAM are not kept. 1: When entering the Software Standby mode or the Deep Software Standby mode 1, the contents of Standby SRAM are kept.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6	IOKEEP	I/O Port Retention 0: When the Deep Software Standby mode is canceled, the I/O ports are in the reset state. 1: When the Deep Software Standby mode is canceled, the I/O ports are in the same state as in the Deep Software Standby mode.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The DPSBYCR register controls the Deep Software Standby mode.

DPSBYCR is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 5, Resets.

DCSSMODE bit (DCDC Soft Start Mode)

The DCSSMODE bit specifies whether to shorten the time required to recover from the Deep Software Standby mode. For details, see section 60, Electrical Characteristics.

SRKEEP bit (Standby SRAM Retention)

The SRKEEP bit specifies whether to keep the contents of Standby SRAM or not when entering the Software Standby mode or the Deep Software Standby mode 1.

IOKEEP bit (I/O Port Retention)

In Deep Software Standby mode, I/O ports keep the same states as in the Software Standby mode. The IOKEEP bit specifies whether to reset the state of the I/O ports or not when the Deep Software Standby mode is canceled.

10.2.18 DPSWCR : Deep Software Standby Wait Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	WTSTS[7:0]							
Value after reset:	0	0	0	0	1	0	1	1

Bit	Symbol	Function	R/W
7:0	WTSTS[7:0]	Deep Software Wait Standby Time Setting Bit 0x0B: Wait cycle for fast recovery 0x9A: Wait cycle for slow recovery*1 Others: Setting prohibited	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. Only Deep Software Standby mode 2 is supported.

The DPSWCR register is a register appointing waiting stabilization time when a Deep Software Standby mode is canceled by certain pins which are sources of external pin interrupts or peripheral interrupts.

During a waiting stabilization period appointed in this register, Deep Software Standby reset occurs, and this MCU is initialized.

The DPSWCR is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 5, Resets

WTSTS[7:0] bits (Deep Software Wait Standby Time Setting Bit)

Set only 2 following values to WTSTS.

1. WTSTS = 0x0B:

The DSTBY recovery time is set to be fast. In this case, after returning from Deep Software Standby mode 2, it is necessary to be secured 700µs in the program before transition to next Software Standby mode or next Deep Software Standby mode.
For details of this fast recovery time, see section 60, Electrical Characteristics

2. WTSTS = 0x9A:
The DSTBY recovery time is set to be slow for Deep Software Standby mode 2. In this case, there is not the time constraint as the case of WTSTS = 0x0B.
For details of this slow recovery time, see section 60, Electrical Characteristics

Note: When you use Deep Software Standby mode 1 or Deep Software Standby mode 3, you should set 0x0B to WTSTS to avoid extra waiting time.

10.2.19 DPSIER0 : Deep Software Standby Interrupt Enable Register 0

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ7E	DIRQ6E	DIRQ5E	DIRQ4E	DIRQ3E	DIRQ2E	DIRQ1E	DIRQ0E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ0E	IRQ0-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
1	DIRQ1E	IRQ1-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
2	DIRQ2E	IRQ2-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
3	DIRQ3E	IRQ3-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
4	DIRQ4E	IRQ4-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
5	DIRQ5E	IRQ5-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
6	DIRQ6E	IRQ6-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
7	DIRQ7E	IRQ7-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W

Note: S-TYPE-4, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIER0 is not initialized by the internal reset signal used as Deep Software Standby mode Canceling source. For details, see section 5, Resets.

After the setting of DPSIER0 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR0 being set to 1. Therefore, DPSIFR0 should be cleared to 0 before entering Deep Software Standby mode.

Refer to Table 10.4 to know the relation between this register and interrupt source for canceling Deep Software Standby mode.

10.2.20 DPSIER1 : Deep Software Standby Interrupt Enable Register 1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ15E	DIRQ14E	DIRQ13E	DIRQ12E	DIRQ11E	DIRQ10E	DIRQ9E	DIRQ8E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ8E	IRQ8-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
1	DIRQ9E	IRQ9-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
2	DIRQ10E	IRQ10-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
3	DIRQ11E	IRQ11-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
4	DIRQ12E	IRQ12-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
5	DIRQ13E	IRQ13-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
6	DIRQ14E	IRQ14-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
7	DIRQ15E	IRQ15-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W

Note: S-TYPE-4, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIER1 is not initialized by the internal reset signal used as Deep Software Standby mode Canceling source. For details, see section 5, Resets.

After the setting of DPSIER1 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR1 being set to 1. Therefore, DPSIFR1 should be cleared to 0 before entering Deep Software Standby mode.

Refer to Table 10.4 to know the relation between this register and interrupt source for canceling Deep Software Standby mode.

10.2.21 DPSIER2 : Deep Software Standby Interrupt Enable Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DNMIE	DRTCAIE	DRTCIIE	DPVD2IE	DPVD1IE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DPVD1IE	PVD1 Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
1	DPVD2IE	PVD2 Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
2	DRTCIIE	RTC Interval interrupt Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
3	DRTCAIE	RTC Alarm interrupt Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
4	DNMIE	NMI Pin Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W*1
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-4, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. 1 can be written only once. Once 1 is written to this bit, subsequent write accesses are disabled.

DPSIER2 is not initialized by the internal reset signal used as Deep Software Standby mode Canceling source. For details, see section 5, Resets.

After the setting of DPSIER2 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR2 being set to 1. Therefore, DPSIFR2 should be cleared to 0 before entering Deep Software Standby mode.

Refer to Table 10.4 to know the relation between this register and interrupt source for canceling Deep Software Standby mode.

10.2.22 DPSIER3 : Deep Software Standby Interrupt Enable Register 3

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA14

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DVBATTADIE	—	DIWDTIE	—	DULPT1IE	DULPT0IE	DUSBHSIE	DUSBFSIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DUSBFSIE	USBFS Suspend/Resume Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
1	DUSBHSIE	USBHS Suspend/Resume Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
2	DULPT0IE	ULPT0 Overflow Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
3	DULPT1IE	ULPT1 Overflow Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	DIWDTIE	IWDT Underflow Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W

Bit	Symbol	Function	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	DVBATTADIE	VBATT Tamper Detection Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W

Note: S-TYPE-4, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIER3 is not initialized by the internal reset signal used as Deep Software Standby mode Canceling source. For details, see section 5, Resets.

After the setting of DPSIER3 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR3 being set to 1. Therefore, DPSIFR3 should be cleared to 0 before entering Deep Software Standby mode.

Refer to Table 10.4 to know the relation between this register and interrupt source for canceling Deep Software Standby mode.

10.2.23 DPSIFR0 : Deep Software Standby Interrupt Flag Register 0

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA18

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ7F	DIRQ6F	DIRQ5F	DIRQ4F	DIRQ3F	DIRQ2F	DIRQ1F	DIRQ0F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ0F	IRQ0-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
1	DIRQ1F	IRQ1-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
2	DIRQ2F	IRQ2-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
3	DIRQ3F	IRQ3-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
4	DIRQ4F	IRQ4-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
5	DIRQ5F	IRQ5-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
6	DIRQ6F	IRQ6-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
7	DIRQ7F	IRQ7-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1

Note: S-TYPE-4, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to clear flag. The flag must be cleared by writing 0 after 1 is read.

Each flag is set to 1 when a cancel request specified by DPSIEGR0 is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not only in Deep Software Standby mode) or when the setting of DPSIER0 is modified. Therefore, a transition to Deep Software Standby mode should be made after DPSIFR0 is cleared to 0x00.

To clear DPSIFR0 to 0x00 after modifying DPSIER0, wait for at least 6 PCLKB cycles, read DPSIFR0, and then write 0 to DPSIFR0. Six or more PCLKB cycles can be secured, for example, by reading DPSIER0.

DPSIFR0 is not initialized by the internal reset signal used as Deep Software Standby mode Canceling source.

For details, see section 5, Resets.

DIRQnF flag (IRQn-DS Pin Deep Software Standby Cancel Flag) (n = 0 to 7)

The DIRQnF flag indicates that a cancel request by the IRQn-DS pin has been generated.

[Setting condition]

A cancel request by the IRQn-DS pin specified by DPSIEGR0 is generated.

[Clearing condition]

Writing 0 to each flag after 1 is read.

10.2.24 DPSIFR1 : Deep Software Standby Interrupt Flag Register 1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA1C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ15F	DIRQ14F	DIRQ13F	DIRQ12F	DIRQ11F	DIRQ10F	DIRQ9F	DIRQ8F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ8F	IRQ8-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
1	DIRQ9F	IRQ9-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
2	DIRQ10F	IRQ10-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
3	DIRQ11F	IRQ11-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
4	DIRQ12F	IRQ12-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
5	DIRQ13F	IRQ13-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
6	DIRQ14F	IRQ14-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
7	DIRQ15F	IRQ15-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1

Note: S-TYPE-4, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to clear flag. The flag must be cleared by writing 0 after 1 is read.

Each flag is set to 1 when a cancel request specified by DPSIEGR1 is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not only in Deep Software Standby mode) or when the setting of DPSIER1 is modified. Therefore, a transition to Deep Software Standby mode should be made after DPSIFR1 is cleared to 0x00.

To clear DPSIFR1 to 0x00 after modifying DPSIER1, wait for at least 6 PCLKB cycles, read DPSIFR1, and then write 0 to DPSIFR1. Six or more PCLKB cycles can be secured, for example, by reading DPSIER1.

DPSIFR1 is not initialized by the internal reset signal used as Deep Software Standby mode Canceling source. For details, see section 5, Resets.

DIRQnF flag (IRQn-DS Pin Deep Software Standby Cancel Flag) (n = 8 to 15)

The DIRQnF flag indicates that a cancel request by the IRQn-DS pin has been generated.

[Setting condition]

A cancel request by the IRQn-DS pin specified by DPSIEGR1 is generated.

[Clearing condition]

Writing 0 to each flag after 1 is read.

10.2.25 DPSIFR2 : Deep Software Standby Interrupt Flag Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA20

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DNMIF	DRTCAIF	DRTCIIF	DPVD2IF	DPVD1IF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DPVD1IF	PVD1 Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
1	DPVD2IF	PVD2 Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
2	DRTCIIF	RTC Interval Interrupt Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
3	DRTCAIF	RTC Alarm Interrupt Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
4	DNMIF	NMI Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-4, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to clear flag. The flag must be cleared by writing 0 after 1 is read.

Each flag is set to 1 when a cancel request specified by DPSIEGR2 is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not only in Deep Software Standby mode) or when the setting of DPSIER2 is modified. Therefore, a transition to Deep Software Standby mode should be made after DPSIFR2 is cleared to 0x00.

To clear DPSIFR2 to 0x00 after modifying DPSIER2, wait for at least 6 PCLKB cycles, read DPSIFR2, and then write 0 to DPSIFR2. Six or more PCLKB cycles can be secured, for example, by reading DPSIER2.

DPSIFR2 is not initialized by the internal reset signal used as Deep Software Standby mode Canceling source. For details, see section 5, Resets.

DPVDmIF flag (PVDm Deep Software Standby Cancel Flag) (m = 1 to 2)

The DPVDmIF flag indicates that a cancel request by the voltage monitor m signal has been generated.

[Setting condition]

A cancel request is generated by the voltage monitor m signal that is selected in DPSIEGR2.

[Clearing condition]

Writing 0 to each flag after 1 is read.

DRTCIIF flag (RTC Interval Interrupt Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the RTC interval interrupt signal has been generated.

[Setting condition]

A cancel request by the RTC interval interrupt signal is generated

[Clearing condition]

Writing 0 to each flag after 1 is read.

DRTCAIF flag (RTC Alarm Interrupt Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the RTC alarm interrupt signal has been generated.

[Setting condition]

A cancel request by the RTC alarm interrupt signal is generated

[Clearing condition]

Writing 0 to each flag after 1 is read.

DNMIF flag (NMI Pin Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the NMI pin has been generated.

[Setting condition]

A cancel request by the NMI pin specified by DPSIEGR2 is generated

[Clearing condition]

Writing 0 to each flag after 1 is read.

10.2.26 DPSIFR3 : Deep Software Standby Interrupt Flag Register 3

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA24

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DVBATTADIF	—	DIWDTIF	—	DULPT1IF	DULPT0IF	DUSBHSIF	DUSBFSIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DUSBFSIF	USBFS Suspend/Resume Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
1	DUSBHSIF	USBHS Suspend/Resume Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
2	DULPT0IF	ULPT0 Overflow Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1

Bit	Symbol	Function	R/W
3	DULPT1IF	ULPT1 Overflow Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
4	—	This bit is read as 0. The write value should be 0.	R/W
5	DIWDTIF	IWDT Underflow Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
6	—	This bit is read as 0. The write value should be 0.	R/W
7	DVBATTADIF	VBATT Tamper Detection Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1

Note: S-TYPE-4, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to clear flag. The flag must be cleared by writing 0 after 1 is read.

Each flag is set to 1 when the corresponding cancel request is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not only in Deep Software Standby mode) or when the setting of DPSIER3 is modified. Therefore, a transition to Deep Software Standby mode should be made after DPSIFR3 is cleared to 0x00.

To clear DPSIFR3 to 0x00 after modifying DPSIER3, wait for at least 6 PCLKB cycles, read DPSIFR3, and then write 0 to DPSIFR3. Six or more PCLKB cycles can be secured, for example, by reading DPSIER3.

DPSIFR3 is not initialized by the internal reset signal used as Deep Software Standby mode Canceling source. For details, see section 5, Resets.

DUSBFSIF flag (USBFS Suspend/Resume Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the USBFS suspend/resume has been generated.

The DUSBFSIF flag is a flag for USBFS.

[Setting condition]

A cancel request by the USBFS suspend/resume is generated.

[Clearing condition]

Writing 0 to each bit after reading the bit as 1.

DUSBHSIF bit (USBHS Suspend/Resume Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the USBHS suspend/resume has been generated.

The DUSBHSIF flag is a flag for USBHS.

[Setting condition]

A cancel request by the USBHS suspend/resume is generated.

[Clearing condition]

Writing 0 to each bit after reading the bit as 1.

DULPT0IF flag (ULPT0 Overflow Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the ULPT0 overflow has been generated.

[Setting condition]

A cancel request by the ULPT0 is generated.

[Clearing condition]

Writing 0 to each bit after reading the bit as 1.

DULPT1IF flag (ULPT1 Overflow Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the ULPT1 overflow has been generated.

[Setting condition]

A cancel request by the ULPT1 is generated.

[Clearing condition]

Writing 0 to each bit after reading the bit as 1.

DIWDTIF bit (IWDT Underflow Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the IWDT underflow has been generated.

[Setting condition]

A cancel request by the IWDT is generated.

[Clearing condition]

Writing 0 to each bit after reading the bit as 1.

DVBATTADIF bit (VBATT Tamper Detection Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the VBATT tamper detection has been generated.

[Setting condition]

A cancel request by the VBATT tamper detection is generated.

[Clearing condition]

Writing 0 to each bit after reading the bit as 1.

10.2.27 DPSIEGR0 : Deep Software Standby Interrupt Edge Register 0

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA28

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ7EG	DIRQ6EG	DIRQ5EG	DIRQ4EG	DIRQ3EG	DIRQ2EG	DIRQ1EG	DIRQ0EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ0EG	IRQ0-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
1	DIRQ1EG	IRQ1-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
2	DIRQ2EG	IRQ2-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
3	DIRQ3EG	IRQ3-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
4	DIRQ4EG	IRQ4-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
5	DIRQ5EG	IRQ5-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
6	DIRQ6EG	IRQ6-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W

Bit	Symbol	Function	R/W
7	DIRQ7EG	IRQ7-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W

Note: S-TYPE-4, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIEGR0 is not initialized by the internal reset signal that is the source to cancel the Deep Software Standby mode. For details, see section 5, Resets.

10.2.28 DPSIEGR1 : Deep Software Standby Interrupt Edge Register 1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA2C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ15EG	DIRQ14EG	DIRQ13EG	DIRQ12EG	DIRQ11EG	DIRQ10EG	DIRQ9EG	DIRQ8EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ8EG	IRQ8-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W
1	DIRQ9EG	IRQ9-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W
2	DIRQ10EG	IRQ10-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge	R/W
3	DIRQ11EG	IRQ11-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W
4	DIRQ12EG	IRQ12-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W
5	DIRQ13EG	IRQ13-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W
6	DIRQ14EG	IRQ14-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W
7	DIRQ15EG	IRQ15-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W

Note: S-TYPE-4, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIEGR1 is not initialized by the internal reset signal that is the source to cancel the Deep Software Standby mode. For details, see section 5, Resets.

10.2.29 DPSIEGR2 : Deep Software Standby Interrupt Edge Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA30

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DNMIEG	—	—	DPVD2EG	DPVD1EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DPVD1EG	PVD1 Edge Select 0: A cancel request is generated when VCC < V_{det1} (fall) is detected 1: A cancel request is generated when VCC ≥ V_{det1} (rise) is detected	R/W
1	DPVD2EG	PVD2 Edge Select 0: A cancel request is generated when VCC < V_{det2} (fall) is detected 1: A cancel request is generated when VCC ≥ V_{det2} (rise) is detected	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	DNMIEG	NMI Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-4, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIEGR2 is not initialized by the internal reset signal that is the source to cancel the Deep Software Standby mode. For details, see section 5, Resets.

10.2.30 SYOCDCR : System Control OCD Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA38

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DBGEN	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	x

Bit	Symbol	Function	R/W
0	—	The read value is undefined. The write value should be 0.	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	DBGEN	Debugger Enable bit 0: On-chip debugger is disabled 1: On-chip debugger is enabled*1	R/W

Note: S-TYPE-5, P-TYPE-2
Note 1. The value of DBGEN can be set by software. However, it is not until the debugger authentication passes that the on-chip debugger function is valid. That is, even if the DBGEN is 1, the on-chip debugger function is invalid unless the debugger authentication passes. For detail, see section 2, CPU.

This register is not controlled by any security attribute register (e.g. LPMSAR, DPFSAR).

DBGEN bit (Debugger Enable bit)

The DBGEN bit enables the on-chip debugger function.

[Setting condition]

- Power-on reset is generated

- Voltage Monitor 0 reset is generated.
- Writing 1 to the bit.

[Clearing condition]

- Writing 0 to the bit.

Note: Certain restrictions apply in terms of the MCU states in which the DBGEN bit can be set to 1. For details, see section 2.13.2. Restriction in OCD Emulator Connecting.

10.2.31 PLL1LDOCR : PLL1-LDO Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xB04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SKEEP	LDOSTP
Value after reset:	0	0	0	0	0	x	0	0

Bit	Symbol	Function	R/W
0	LDOSTP	LDO Stop 0: PLL1-LDO is enabled 1: PLL1-LDO is stopped	R/W
1	SKEEP	STBY Keep 0: PLL1-LDO is stopped during Software Standby mode. 1: PLL1-LDO state before Software Standby mode is retained during Software Standby mode.	R/W
2	—	The read value is undefined.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

It can be controlled the operation of the LDO dedicated for PLL1(PLL1-LDO) by using this register.

Writing to this register is prohibited in low-speed mode.

LDOSTP bit (LDO Stop)

The PLL1-LDO state during Normal mode is controlled by this bit. After setting 1 to this bit, do not set 0 to this bit for at least 1 µs. This bit can be set 0 to this bit only when 1us has elapsed after setting 1 to this bit.

When this bit is changed from 1 to 0, it takes 25 µs for the power supply to stabilize.

Allow the power supply to stabilize before operating the PLL1.

SKEEP bit (STBY Keep)

The PLL1-LDO state during Software Standby mode is controlled by this bit. If the PLL1-LDO operation is stopped during Software Standby mode, the power consumption during Software Standby mode can be reduced.

10.2.32 PLL2LDOCR : PLL2-LDO Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xB08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SKEEP	LDOSTP
Value after reset:	0	0	0	0	0	x	0	0

Bit	Symbol	Function	R/W
0	LDOSTP	LDO Stop 0: PLL2-LDO is enabled 1: PLL2-LDO is stopped	R/W
1	SKEEP	STBY Keep 0: PLL2-LDO is stopped during Software Standby mode. 1: PLL2-LDO state before Software Standby mode is retained during Software Standby mode.	R/W
2	—	The read value is undefined.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

It can be controlled the operation of the LDO dedicated for PLL2 (PLL2-LDO) by using this register.

Writing to this register is prohibited in low-speed mode.

LDOSTP bit (LDO Stop)

The PLL2-LDO state during Normal mode is controlled by this bit. After setting 1 to this bit, do not set 0 to this bit for at least 1 µs. This bit can be set 0 to this bit only when 1us has elapsed after setting 1 to this bit.

When this bit is changed from 1 to 0, it takes 25 µs for the power supply to stabilize.

Allow the power supply to stabilize before operating the PLL2.

SKEEP bit (STBY Keep)

The PLL2-LDO state during Software Standby mode is controlled by this bit. If the PLL2-LDO operation is stopped during Software Standby mode, the power consumption during Software Standby mode can be reduced.

10.2.33 HOCOLDOCR : HOCO-LDO Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xB0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SKEEP	LDOSTP
Value after reset:	0	0	0	0	0	x	0	0

Bit	Symbol	Function	R/W
0	LDOSTP	LDO Stop 0: HOCO-LDO is enabled 1: HOCO-LDO is stopped	R/W
1	SKEEP	STBY Keep 0: HOCO-LDO is stopped during Software Standby mode. 1: HOCO-LDO state before Software Standby mode is retained during Software Standby mode.	R/W
2	—	The read value is undefined.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

It can be controlled the operation of the LDO dedicated for HOCO (HOCO-LDO) by using this register.

Writing to this register is prohibited in low-speed mode.

LDOSTP bit (LDO Stop)

The HOCO-LDO state during Normal mode is controlled by this bit. After setting 1 to this bit, do not set 0 to this bit for at least 2 µs. This bit can be set 0 to this bit only when 2 µs has elapsed after setting 1 to this bit.

When this bit is changed from 1 to 0, it takes 5 µs for the power supply to stabilize.

Allow the power supply to stabilize before operating the HOCO.

SKEEP bit (STBY Keep)

The HOCO-LDO state during Software Standby mode is controlled by this bit. If the HOCO-LDO operation is stopped during Software Standby mode, the power consumption during Software Standby mode can be reduced.

10.3 Reducing Power Consumption by Switching Clock Signals

The clock frequency changes when the SCKDIVCR and SCKDIVCR2 registers are set.

For information on module and clock associations, see section 8.2.2. SCKDIVCR : System Clock Division Control Register and section 8.2.3. SCKDIVCR2 : System Clock Division Control Register 2.

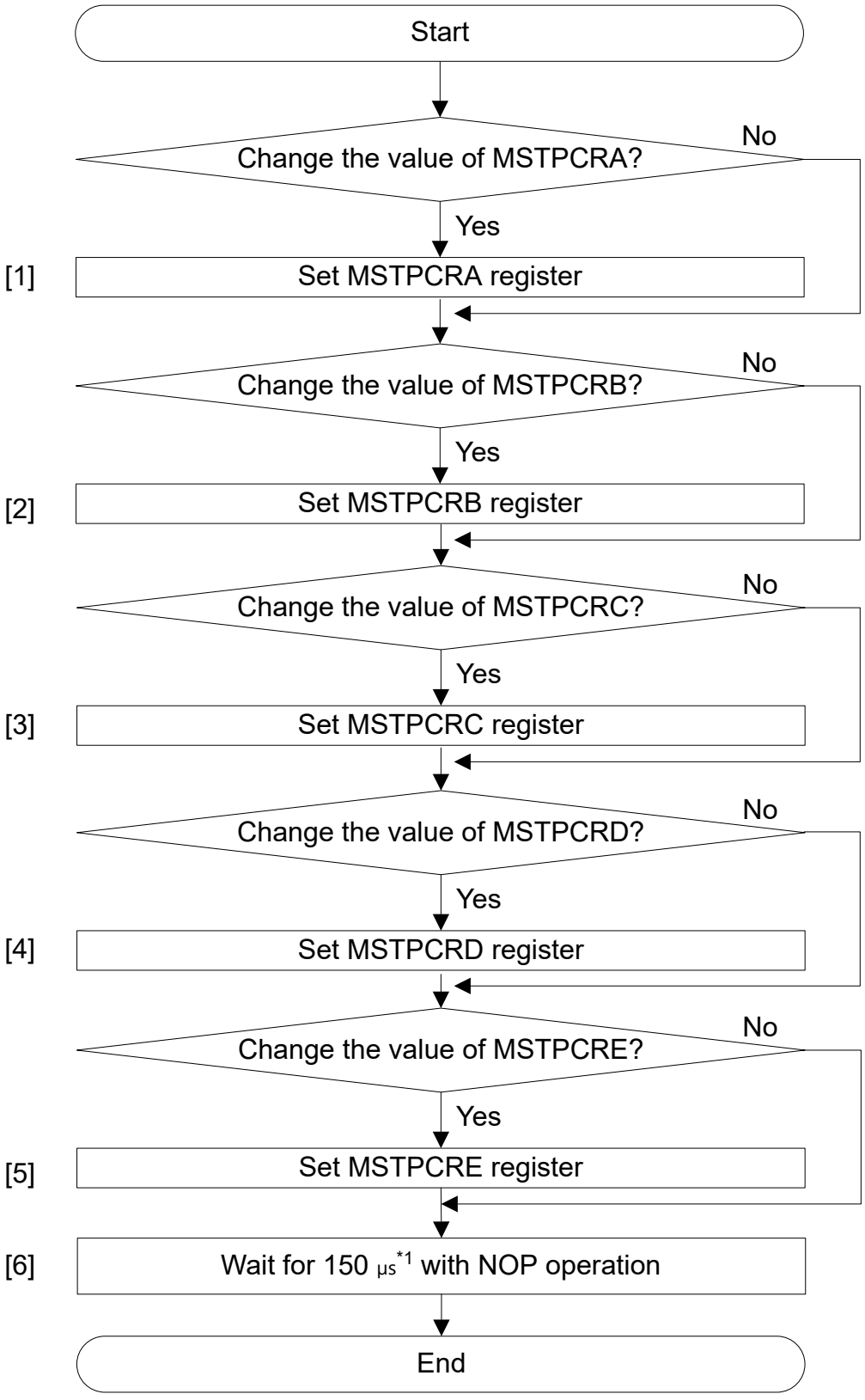
10.4 Module-Stop Function

The module stop function can stop the clock supply set for each peripheral module.

When the MSTPmi bit (m = A to E, i = 31 to 0) in MSTPCRn (n = A to E) is set to 1, the specified module stops operating and enters the module-stop state, but the CPU continues to operate independently. Setting the MSTPmi bit to 0 cancels the module-stop state, allowing the module to resume operation at the end of the bus cycle.

After a reset is canceled, all modules other than the DMAC, DTC and SRAMn (n = 0, 1) modules are placed in the module-stop state. Do not access the module while the corresponding MSTPmi bit is 1. Additionally, do not set 1 to the MSTPmi bit while the corresponding module is accessed.

If the CPU clock is faster than 120 MHz, after the MSTPmi bit is changed, a wait time should be provided according to the flow shown in Figure 10.2. It is recommended to use software to measure the wait time. Be sure to consider the worst-case conditions to ensure that the required wait time elapses. If an interrupt is unavoidably generated during the wait time, retry the measurement after return from the interrupt.



Note 1. 150µs is in DCDC mode, 10µs in External VDD mode

Figure 10.2 Setting flow of module stop control registers in case of CPUCLK is higher than 120MHz

10.5 Function for Lower Operating Power Consumption

By selecting an appropriate operating power consumption control mode according to the operating frequency, power consumption can be reduced in Normal mode and Processor low power mode.

10.5.1 Setting Operating Power Control Mode

Ensure the operating condition such as the frequency range is always within the specified range before and after switching the operating power control modes.

In External VDD mode, make sure to set OPCCR.OPCM is 00b.

This section provides example procedures for switching operating power control modes.

Table 10.8 Available oscillators in each mode

Mode	Oscillator					
	PLL1, PLL2	High-speed on-chip oscillator	Middle-speed on-chip oscillator	Low-speed on-chip oscillator	Main clock oscillator	Sub-clock oscillator
High-speed	Available	Available	Available	Available	Available	Available
Low-speed	Not Available	Available	Available	Available	Available	Available

(1) Switching from a higher power mode to a lower power mode

Example : From High-speed mode to Low-speed mode:

(Operation begins in High-speed mode)

1. Change the oscillator to what is used in Low-speed mode. Set the frequency of each clock lower than or equal to the maximum operating frequency in Low-speed mode.
2. Turn off the oscillator that is not required in Low-speed mode.
3. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
4. Set the OPCCR.OPCM[1:0] bits to 11b (Low-speed mode).
5. Confirm that OPCCR.OPCMTSF flag is 0 (indicates transition completed).

(2) Switching from a lower power mode to a higher power mode

Example : From Low-speed mode to High-speed mode

(Operation begins in Low-speed mode)

1. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
2. Set the OPCCR.OPCM[1:0] bits to 00b (High-speed mode).
3. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
4. Turn on any required oscillator in High-speed mode.
5. Set the frequency of each clock lower than or equal to the maximum operating frequency for High-speed mode.

(Operation is now in High-speed mode)

10.6 Power Gating Control

The MCU supports the power gating function of the power domain.

User can individually control the power on/off of the divided power domains.

The power domain is consist of normal operation-area and retention-area.

10.6.1 Power Gating Control Using Power Domain Control Register

Table 10.9 shows the relationship Power gating control register and the target power domain.

When the Power gating control register is set, power gating control is executed.

In addition, when using the power gating function, it should be set MOCOCR.MCSTP to 0 (MOCO is operated) in advance.

Table 10.9 Relationship between the Power gating control register and the target power domain

Control register				Target Power Domain name	Implemented module name
Register name	PDDE	PDCSF	PDPGSF		
PDCTRGD	Available	Available	Available	Graphic domain	MIPI DSI, DRW, GLCDC

Examples of the procedures for power gating control:

(1) Power OFF by PDCTRGD.PDDE*1

STEP1 : Confirm that the target domain is not performing the Power gating process

Check PDCTRGD.PDCSF = 0

STEP2 : Set the module stop control register to stop.

Write MSTPCR* = 1 (*:all register for target domain)

STEP3 : Set PDCTRGD.PDDE bit and start the power control.

Write PDCTRGD.PDDE = 1

STEP4 : Confirm that the power gating process is completed and the powered off

Check PDCTRGD.PDCSF = 0 & PDCTRGD.PDPGSF = 1

(2) Power ON by PDCTRGD.PDDE*1

STEP1 : Confirm that the target domain is not performing the Power gating process

Check PDCTRGD.PDCSF = 0

STEP2 : Set PDCTRGD.PDDE bit and start the power control.

Write PDCTRGD.PDDE = 0

STEP3 : Confirm that the power gating process is completed and the powered on

Check PDCTRGD.PDCSF = 0 & PDCTRGD.PDPGSF = 0

STEP4 : Set the module stop control register to operate.

Write MSTPCR* = 0 (*:all register for target domain)

Note 1. When PDCTRGD.PDCSF bit is 1 (during power gating control processing) , PDCTRGD register must not be changed.

10.7 Low Power Modes

10.7.1 CPU Sleep Mode

10.7.1.1 Transition to CPU Sleep Mode

When a WFI instruction is executed while SCR.SLEEPDEEP bit is 0, the MCU enters CPU Sleep mode. In this mode, the CPU stops operating but the contents of its internal registers are retained. Other peripheral functions do not stop. Available resets or interrupts in CPU Sleep mode cause the MCU to cancel CPU Sleep mode. All interrupt sources are available. If using an interrupt to cancel CPU Sleep mode, you must set the associated IELSRn register before executing a WFI instruction. For details, see section 13, Interrupt Controller Unit (ICU).

Counting by IWDT stops when the MCU enters CPU Sleep mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 1 (IWDT stops in CPU Sleep mode or CPU Deep Sleep mode). Similarly, counting by IWDT stops when the MCU enters CPU Sleep mode while the IWDT is in register start mode and the SLCSTP bit in IWDTCSTPR is 1 (IWDT stops in CPU Sleep mode or CPU Deep Sleep mode).

Counting by IWDT continues when the MCU enters CPU Sleep mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 0 (IWDT does not stop in CPU Sleep mode or CPU Deep Sleep mode). Similarly, counting by IWDT continues when the MCU enters CPU Sleep mode while the IWDT is in register start mode and the SLCSTP bit in IWDTCSTPR is 0 (IWDT does not stop in CPU Sleep mode or CPU Deep Sleep mode).

Counting by WDT stops when the MCU enters CPU Sleep mode while the WDT is in auto start mode and the OFS0.WDTSTPCTL bit is 1 (WDT stops in CPU Sleep mode or CPU Deep Sleep mode). Similarly, counting by WDT

stops when the MCU enters CPU Sleep mode while the WDT is in register start mode and the SLCSTP bit in WDTCSTPR is 1 (WDT stops in CPU Sleep mode or CPU Deep Sleep mode).

Counting by WDT continues when the MCU enters CPU Sleep mode while the WDT is in auto start mode and the OFS0.WDTSTPCTL bit is 0 (WDT does not stop in CPU Sleep mode or CPU Deep Sleep mode). Similarly, counting by WDT continues when the MCU enters CPU Sleep mode while the WDT is in register start mode and the SLCSTP bit in WDTCSTPR is 0 (WDT does not stop in CPU Sleep mode or CPU Deep Sleep mode).

WFI instruction and sleep-on-exit function for entering CPU Sleep mode purpose.

For the transition to CPU Sleep mode, there is a procedure according to the CPUCLK frequency. For details, see section 10.8.10. Notes on Transitioning to or Canceling Low Power State.

10.7.1.2 Canceling CPU Sleep Mode

CPU Sleep mode is canceled by:

- An interrupt
- A reset:
  - RES pin reset
  - Power-on reset
  - Voltage Monitor 0 reset
  - Independent watchdog timer reset
  - Watchdog timer reset
  - Voltage Monitor m reset (m = 1, 2)
  - Common memory error reset
  - Bus error reset

The operations are as follows:

1. Canceling by an interrupt
   When an available interrupt request is generated, CPU Sleep mode is canceled and the MCU starts the interrupt handling.
2. Canceling by RES pin reset
   When the RES pin is driven low, the MCU enters the reset state. Be sure to keep the RES pin low for the time period specified in section 60, Electrical Characteristicssection 50, Electrical Characteristics. When the RES pin is driven high after the specified time period, the CPU starts the reset exception handling.
3. Canceling by IWDT reset
   CPU Sleep mode is canceled by an internal reset generated by an IWDT underflow and the MCU starts the reset exception handling. However, IWDT stops in CPU Sleep mode and an internal reset for canceling CPU Sleep mode is not generated in the following conditions:
   - OFS0.IWDTSTRT = 0 (auto start mode) and OFS0.IWDTSTPCTL = 1
   - OFS0.IWDTSTRT = 1 (register start mode) and OFS0.IWDTSTPCTL = 1
4. Canceling by WDT reset
   CPU Sleep mode is canceled by an internal reset generated by a WDT underflow and the MCU starts the reset exception handling. However, WDT stops in CPU Sleep mode even when counting in Normal mode and an internal reset for canceling CPU Sleep mode is not generated in the following conditions:
   - OFS0.WDTSTRT = 0 (auto start mode) and OFS0.WDTSTPCTL = 1
   - OFS0.WDTSTRT = 1 (register start mode) and WDTCSTPR.SLCSTP = 1
5. Canceling by other resets available in CPU Sleep mode
   CPU Sleep mode is canceled by other resets and the MCU starts the reset exception handling.

Note: For details on proper setting of the interrupts, see section 13, Interrupt Controller Unit (ICU)

10.7.2 CPU Deep Sleep Mode

10.7.2.1 Transition to CPU Deep Sleep Mode

When a WFI instruction is executed while SCR.SLEEPDEEP is set as 1, the MCU enters CPU Deep Sleep mode. In this mode, the CPU stops operating but the contents of its internal registers are retained. Other peripheral functions do not stop. However, accessing to TCM is not available and SysTick also stops in this mode. Available resets or wakeup enabled interrupts in CPU Deep Sleep mode cause the MCU to cancel CPU Deep Sleep mode. Not all interrupt sources are available, in detail user must also set WUPEN0 and WUPEN1 registers located in ICU to enable wakeup from CPU Deep Sleep mode. For details, see section 13, Interrupt Controller Unit (ICU).

IWDT and WDT operation in CPU Deep Sleep mode is same with CPU Sleep mode.

For the transition to CPU Deep Sleep mode, there is a procedure according to the CPUCLK frequency. For details, see section 10.8.10. Notes on Transitioning to or Canceling Low Power State.

10.7.2.2 Canceling CPU Deep Sleep Mode

CPU Deep Sleep mode is canceled by:

- An available interrupt shown in Table 10.4.
- A reset:
  - RES pin reset
  - Power-on reset
  - Voltage Monitor 0 reset
  - Independent watchdog timer reset
  - Watchdog timer reset
  - Voltage Monitor m reset (m = 1, 2)
  - Common memory error reset
  - Bus error reset

The operations are as follows:

1. Canceling by an interrupt
   When an available interrupt request (for available interrupt, See Table 10.4) is generated, each oscillator which was operating before the transition to CPU Deep Sleep mode restarts. After all these oscillators have been stabilized, the MCU returns from CPU Deep Sleep mode to Normal mode and starts interrupt handling. When canceling the CPU Deep Sleep mode by interrupt, each power domain returns to the same power state as the state before entering the Deep Sleep state
   (For available interrupt, See Table 10.4)
2. Canceling by RES pin reset
   When the RES pin is driven low, the MCU enters the reset state. Be sure to keep the RES pin low for the time period specified in section 60, Electrical Characteristics. When the RES pin is driven high after the specified time period, the CPU starts the reset exception handling.
3. Canceling by IWDT reset
   CPU Deep Sleep mode is canceled by an internal reset generated by an IWDT underflow and the MCU starts the reset exception handling. However, IWDT stops in CPU Deep Sleep mode and an internal reset for canceling CPU Deep Sleep mode is not generated in the following conditions:
   - OFS0.IWDTSTRT = 0 (auto start mode) and OFS0.IWDTSTPCTL = 1
   - OFS0.IWDTSTRT = 1 (register start mode) and IWDTCSTPR.SLCSTP = 1
4. Canceling by WDT reset
   CPU Deep Sleep mode is canceled by an internal reset generated by a WDT underflow and the MCU starts the reset exception handling. However, WDT stops in CPU Deep Sleep mode even when counting in Normal mode and an internal reset for canceling CPU Deep Sleep mode is not generated in the following conditions:
   - OFS0.WDTSTRT = 0 (auto start mode) and OFS0.WDTSTPCTL = 1

- OFS0.WDTSTRT = 1 (register start mode) and WDTCSTPR.SLCSTP = 1

5. Canceling by other resets available in CPU Deep Sleep mode
   CPU Deep Sleep mode is canceled by other resets and the MCU starts the reset exception handling.

   Note: For details on proper setting of the interrupts, see section 13, Interrupt Controller Unit (ICU)

For CPU Deep Sleep mode cancellation, there is a procedure according to the CPUCLK frequency. For details, see section 10.8.10. Notes on Transitioning to or Canceling Low Power State.

10.7.3 Software Standby Mode

10.7.3.1 Transition to Software Standby Mode

When a WFI instruction is executed while LPSCR.LPMD bit is 0x4 and CPU0.SCR.SLEEPDEEP bit is 1, the MCU enters Software Standby mode depending on LPSCR.LPMD setting. It must be set MOCOCR.MCSTP to 0 (MOCO is operated) before executed the WFI instruction to entry this mode.

In this mode, the CPU, most of the on-chip peripheral functions and most of the oscillators stop*1. However, the contents of the CPU internal registers and SRAM data, the states of on-chip peripheral functions and the I/O ports are retained. Software Standby mode allows significant reduction in power consumption because most of the oscillators stops*1 in this mode. Table 10.3 shows the status of each on-chip peripheral functions and oscillators. Available resets or interrupts in Software Standby mode make the MCU to return to Normal mode. See Table 10.4 for available interrupt sources for each mode. In case of using an interrupt to return to Normal mode, corresponding IELSRn register must be set before executing a WFI instruction. For the detail, see section 13, Interrupt Controller Unit (ICU).

Note 1. The state of the oscillators depends on the setting of the control register for each oscillator. For details, refer to the chapter of clock generation circuit.

The status of address bus and bus control signals in Software Standby mode can be selected by SBYCR.OPE bit. The SBYCR.OPE bit specifies whether to set to the high-impedance state or to retain the output of the address bus and bus control signals in Software Standby mode.

Clear DMAST.DMST bit and DTCST.DTCST bit to 0 before executing WFI instruction.

Counting by IWDT stops when the MCU enters Software Standby mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 1 (IWDT stops in Software Standby mode). Similarly, counting by IWDT stops when the MCU enters Software Standby mode while the IWDT is in register start mode and the SLCSTP bit in IWDTCSTPR is 1 (IWDT stops in Software Standby mode).

Counting by IWDT continues when the MCU enters Software Standby mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 0 (IWDT does not stop in Software Standby mode). Similarly, counting by IWDT continues when the MCU enters Software Standby mode while the IWDT is in register start mode and the SLCSTP bit in IWDTCSTPR is 0 (IWDT does not stop in Software Standby mode).

WDT stops counting when the MCU enters Software Standby mode because the PCLK stops.

Do not enter Software Standby mode while the flash memory is in programming or erasing procedure. In order to enter Software Standby mode, execute a WFI instruction after programming or erasing procedure completes.

In External VDD mode, do not enter Software Standby Mode.

Table 10.10 shows the setting of the related control bits and the modes to enter after executing WFI instruction.

Table 10.10 Setting of the bits what affect the mode after executing WFI instruction

		Setting mode of LPSCR.LPMD bit and CPU.SCR.SLEEPDEEP bit			
		Software Standby	**Deep Software Standby 1**	**Deep Software Standby 2**	**Deep Software Standby 3**
		(LPSCR.LPMD = 0x4, CPU0.SCR. SLEEPDEEP = 1)	**(LPSCR.LPMD = 0x8, CPU0.SCR. SLEEPDEEP = 1)**	**(LPSCR.LPMD = 0x9, CPU0.SCR. SLEEPDEEP = 1)**	**(LPSCR.LPMD = 0xA, CPU0.SCR. SLEEPDEEP = 1)**
FENTRYR.FENTRYC FENTRYR.FENTRYD	0	Software Standby	Deep Software Standby 1	Deep Software Standby 2	Deep Software Standby 3
	1	CPU Deep Sleep mode and Keep System Active	CPU Deep Sleep mode and Keep System Active	CPU Deep Sleep mode and Keep System Active	CPU Deep Sleep mode and Keep System Active
OFS0.IWDTSTPCTL (auto-start mode) IWDTCSTPR.SLCSTP (register start mode)	0	Software Standby	Deep Software Standby 1	Deep Software Standby 1	Deep Software Standby 1
	1			Deep Software Standby 2	Deep Software Standby 3
PVD1CR0.RI PVD2CR0.RI	0	Software Standby	Deep Software Standby 1	Deep Software Standby 2	Deep Software Standby 3
	1			Deep Software Standby 1	Deep Software Standby 1

Note: When multiple mode transition conditions conflict, mode transition is performed according to the following priority order.
CPU Deep Sleep mode and Keep System Active > Software Standby > Deep Software Standby 1 > Deep Software Standby 2 > Deep Software Standby 3.

For the transition to Software Standby mode, there is a procedure according to the CPUCLK frequency. For details, see section 10.8.10. Notes on Transitioning to or Canceling Low Power State.

10.7.3.2 Canceling Software Standby Mode

Software Standby mode is canceled by:

- An available interrupt shown in Table 10.4
- A reset:
  - RES pin reset
  - Power-on reset
  - Voltage monitor reset
  - Reset caused by an IWDT underflow

On exiting Software Standby mode, the oscillators that operate before the transition to the mode restart. After all the oscillators are stabilized, the MCU returns to Normal mode from Software Standby mode. See section 13.2.15. WUPEN0 : Wake Up Interrupt Enable Register 0, section 13.2.16. WUPEN1 : Wake Up interrupt enable register 1 for information on how to wake up the MCU from Software Standby mode.

You can cancel Software Standby mode in any of the following ways:

1. Canceling by an interrupt
   When an available interrupt request (see Table 10.4) is generated, an oscillator that operates before the transition to Software Standby mode restarts. After all the oscillators are stabilized, the MCU returns to Normal mode from Software Standby mode and starts the interrupt handling.
   When canceling the software standby mode by interrupt , each power domain returns to the same power state as the state before entering the software standby state.
2. Canceling by a RES pin reset
   When the RES pin is driven low, the MCU enters the reset state, and the oscillators whose default status is operating, start the oscillation. Be sure to keep the RES pin low for the time period specified in section 60, Electrical Characteristics. When the RES pin is driven high after the specified time period, the CPU starts the reset exception handling.
3. Canceling by a power-on reset

Software Standby mode is canceled by a power-on reset and the MCU starts the reset exception handling.

4. Canceling by a voltage monitor reset
   Software Standby mode is canceled by a voltage monitor reset from the voltage detection circuit and the MCU starts the reset exception handling.
5. Canceling by IWDT reset
   Software Standby mode is canceled by an internal reset generated by an IWDT underflow and the MCU starts the reset exception handling. However, IWDT stops in Software Standby mode and an internal reset for canceling Software Standby mode is not generated in the following condition:
   - OFS0.IWDTSTRT = 0 (auto start mode) and OFS0.IWDTSTPCTL = 1
   - OFS0.IWDTSTRT = 1 (register start mode) and IWDTCSTPR.SLCSTP = 1

For Software Standby mode cancellation, there is a procedure according to the CPUCLK frequency. For details, see section 10.8.10. Notes on Transitioning to or Canceling Low Power State

10.7.3.3 Example of Software Standby Mode Application

Figure 10.3 shows an example of entry to Software Standby mode on detection of a falling edge of the IRQn pin, and exit from Software Standby mode by a rising edge of the IRQn pin.

In this example, an IRQn pin interrupt is accepted with the IRQCRi.IRQMD[1:0] (i = 0 to 15) bits of the ICU set to 00b (falling edge) in Normal mode, and the IRQCRi.IRQMD[1:0] bits are set to 01b (rising edge). Then, if MOCO is not operated, it must set MOCOCR.MCSTP to 0 (MOCO is operated). After that, the LPSCR.LPMD bit is set to 0x4 and CPU.SCR.SLEEPDEEP bit is set to 1, and then a WFI instruction is executed. As a result, entry to Software Standby mode completes and exit from Software Standby mode is initiated by a rising edge of the IRQn pin.

Setting the ICU is also required to exit Software Standby mode. For details, see section 13, Interrupt Controller Unit (ICU). The oscillation stabilization time in Figure 10.3 is specified in section 60, Electrical Characteristics.

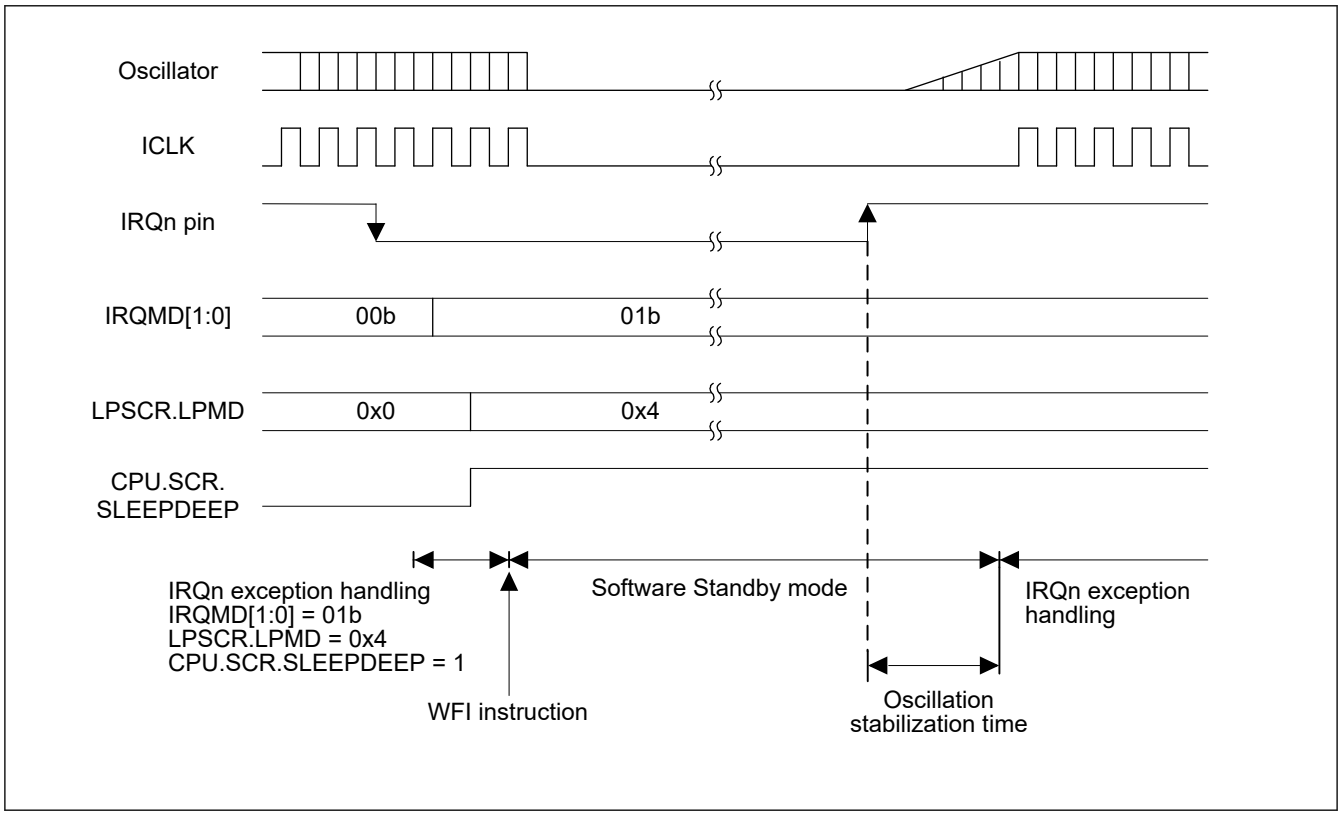


Figure 10.3 Example of Software Standby mode application

10.7.4 Deep Software Standby Mode

10.7.4.1 Transition to Deep Software Standby Mode

When a WFI instruction is are executed while LPSCR.LPMD bit is 0x8, 0x9, 0xA and CPU.SCR.SLEEPDEEP bit is 1, the MCU enters Deep Software Standby mode 1 or 2 or 3 depending on LPSCR.LPMD setting. It must be set MOCOCR.MCSTP to 0 (MOCO is operated) before executed the WFI instruction to entry this mode. See Table 10.10 for the setting of the related control bits.

In these mode , the CPU, on-chip peripheral functions, SRAM, and oscillators are almost stopped as shown in Table 10.3. Furthermore, since the internal power supply to these inactive modules is stopped, power consumption is remarkably reduced. The contents of all the registers of the CPU and specific internal peripheral modules become undefined. Table 10.3. shows the status of each on-chip peripheral functions and oscillators.

Data in the standby SRAM are preserved in Deep Software Standby mode 1. When OFS1(_SEC).PVDAS = 0 and OFS1(_SEC).PVDLPSEL = 0, the low-power-consumption function of PVD0 is enabled, so power consumption is further reduced. In Deep Software Standby mode 2, the internal supply of power to the standby SRAM, the USB resume detecting unit and IWDT are cut off, reducing power consumption. Data in the standby SRAM become undefined at this time. When OFS1(_SEC).PVDAS = 0 and OFS1(_SEC).PVDLPSEL = 0, the low-power-consumption function of PVD0 is enabled, so power consumption is further reduced. In Deep Software Standby mode 3, the PVD is also stopped. For details, see section 60, Electrical Characteristics.

Clear DMAST.DMST bit and DTCST.DTCST bit to 0 before executing WFI instruction.

When the MCU enters Deep Software Standby mode 2 or 3 while the IWDT is in auto-start mode and the OFS0.IWDTSTPCTL bit is 1, power supply to the LOCO and the IWDT is cut off. Counting by the IWDT also stops.

When the IWDT is in auto-start mode and the OFS0.IWDTSTPCTL bit is 0, the MCU enters Deep Software Standby mode 1 instead of Deep Software Standby mode 2 or 3, regardless of the setting in the LPSCR.LPMD.

When the IWDT is in register start mode and the IWDTCSTPR.SLCSTP is 0, the MCU enters Deep Software Standby mode 1 instead of Deep Software Standby mode 2 or 3, regardless of the setting in the LPSCR.LPMD.

Do not enter Deep Software Standby mode while the flash memory is in programming or erasing procedure. In order to enter Deep Software Standby mode, execute a WFI instruction after programming or erasing procedure completes.

In External VDD mode, do not enter Deep Software Standby mode.

When PVDnCR0.RI (n = 1, 2) = 1 (selecting the voltage monitor n reset), the MCU enters Deep Software Standby mode 1 instead of Deep Software Standby mode 2 or 3, regardless of the setting in the LPSCR.LPMD. The I/O port states are same as Software Standby mode.

For the transition to Deep Software Standby mode, there is a procedure according to the CPUCLK frequency. For details, see section 10.8.10. Notes on Transitioning to or Canceling Low Power State

10.7.4.2 Canceling Deep Software Standby Mode

Deep Software Standby mode is canceled by:

- An available interrupt shown in Table 10.4
- A reset:
  - RES pin reset
  - Power-on reset
  - Voltage monitor reset
  - Reset caused by an IWDT underflow

1. Canceling by an interrupt
   Canceling by interrupts is controlled by DPSIERn (n = 0 to 3) and DPSIFRn (n = 0 to 3). When a Deep Software Standby Canceling interrupt is generated, the corresponding flag in DPSIFRn is set to 1. If the interrupt is enabled in DPSIERn, Deep Software Standby mode is canceled. Rising edge or falling edge can be selected by DPSIEGRn (n = 0 to 2). The interrupts for which an edge can be selected are the NMI, IRQ0-DS to IRQ15-DS, voltage monitor n (n = 1, 2) interrupts. When a Deep Software Standby mode canceling request occurs, the internal power is supplied and MOCO starts oscillating, and then the internal reset (Deep Software Standby reset) is generated for the entire MCU.

The stable MOCO clock is then supplied to the entire MCU and Deep Software Standby reset is canceled. And the MCU starts reset exception handling.
When Deep Software Standby mode is canceled by an external interrupt pin or internal interrupt signal, the RSTSR0.DPSRSTF flag is set to 1.

2. Canceling by RES pin reset
When the RES pin is driven low, the MCU cancels Deep Software Standby mode and enters the reset state. Make sure to keep RES pin low for the time period specified in the chapter of Electrical Specification. When RES pin is driven high after the specified time period, the CPU starts the reset exception handling.
3. Canceling by a power-on reset
Deep Software Standby mode is canceled by a power-on reset and the MCU starts the reset exception handling.
4. Canceling by a voltage monitor 0 reset
Deep Software Standby mode is canceled by a voltage monitor 0 reset from the voltage detection circuit and the MCU starts the reset exception handling.
5. Canceling by a voltage monitor n reset (n = 1, 2)
Deep Software Standby mode 1 is canceled by a voltage monitor n reset (n= 1,2))from the voltage detection circuit and the MCU starts the reset exception handling.
6. Canceling by a IWDT reset
Deep Software Standby mode 1 is canceled by a IWDT reset (an internal reset generated by IWDT underflow) and the MCU starts the reset exception handling.
However, IWDT stops in Deep Software Standby mode 1 and a IWDT reset for Canceling Deep Software Standby mode 1 is not generated in the following conditions:
   - OFS0.IWDTSTRT = 0 (auto start mode) and OFS0.IWDTSTPCTL = 1
   - OFS0.IWDTSTRT = 1 (register start mode) and IWDTCSTPR.SLCSTP = 1

10.7.4.3 Pin States When Deep Software Standby Mode is Canceled

In Deep Software Standby mode, the I/O ports retain the same states as before transition to the Deep Software Standby mode. The MCU is initialized by an internal reset generated when Deep Software Standby mode is canceled. Upon cancellation of Deep Software Standby mode, the reset exception handling starts immediately.

Whether to initialize the I/O ports or to retain the I/O port states as before transition to the Deep Software Standby mode can be selected by the DPSBYCR.IOKEEP bit.

- When the DPSBYCR.IOKEEP bit = 0
I/O ports are initialized by an internal reset generated when Deep Software Standby mode is canceled.
- When the DPSBYCR.IOKEEP bit = 1
Although the MCU is initialized by an internal reset generated when Deep Software Standby mode is canceled, I/O ports retain their states as before transition to the Deep Software Standby mode regardless of the MCU internal state. At this time, the I/O port states remain unchanged as before transition to the Deep Software Standby mode even if settings of I/O ports or peripheral modules are made. Then, the retained I/O port states are released by clearing the DPSBYCR.IOKEEP bit to 0, and the MCU operates according to the internal state.

The DPSBYCR.IOKEEP bit is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode.

10.7.4.4 Example of Deep Software Standby Mode Application

(1) Entering and exiting Deep Software Standby mode

Figure 10.4 shows an example where a transition to Deep Software Standby mode is made at the falling edge of the IRQn-DS pin, and exiting Deep Software Standby mode is made at the rising edge of the IRQn-DS pin. In this example, an IRQn interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 00b (falling edge). If MOCO is not operated, it must set MOCOCR.MCSTP to 0 (MOCO is operated).

Then, after the DPSIEGRy.DIRQnEG (y = 0 or 1, n = 0 to 15) bit is set to 1 (rising edge) and the LPSCR.LPMD bit is set to 0x8 or 0x9 or 0xA and CPU.SCR.SLEEPDEEP bit is set to 1, the WFI instruction is executed. Thus a transition to Deep Software Standby mode is made. After that, exiting Deep Software Standby mode is made at the rising edge of the IRQn-DS pin.

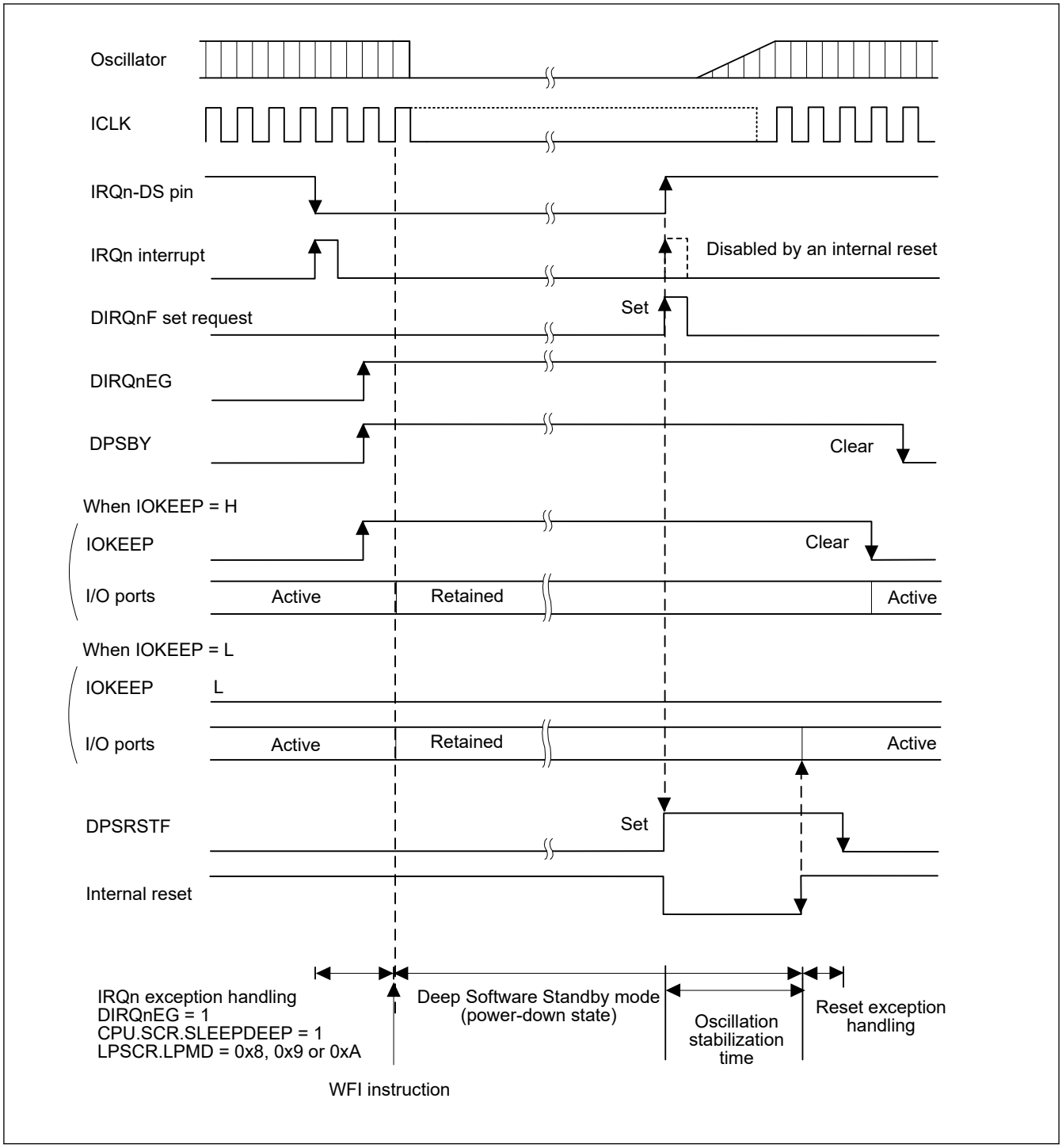


Figure 10.4 Example of Deep Software Standby mode application

10.7.4.5 Usage Flow for Deep Software Standby Mode

Figure 10.5 shows an example flow for using Deep Software Standby mode.

In this example, the RSTSR0.DPSRSTF flag of the reset function is read after the reset exception handling to determine whether a reset was generated by the RES pin or by the cancellation of Deep Software Standby mode.

For a reset by the RES pin, the MCU transitions to Deep Software Standby mode after the required register settings are made.

For a reset by cancellation of Deep Software Standby mode, the DPSBYCR.IOKEEP bit is cleared to 0 after the I/O port settings are made.

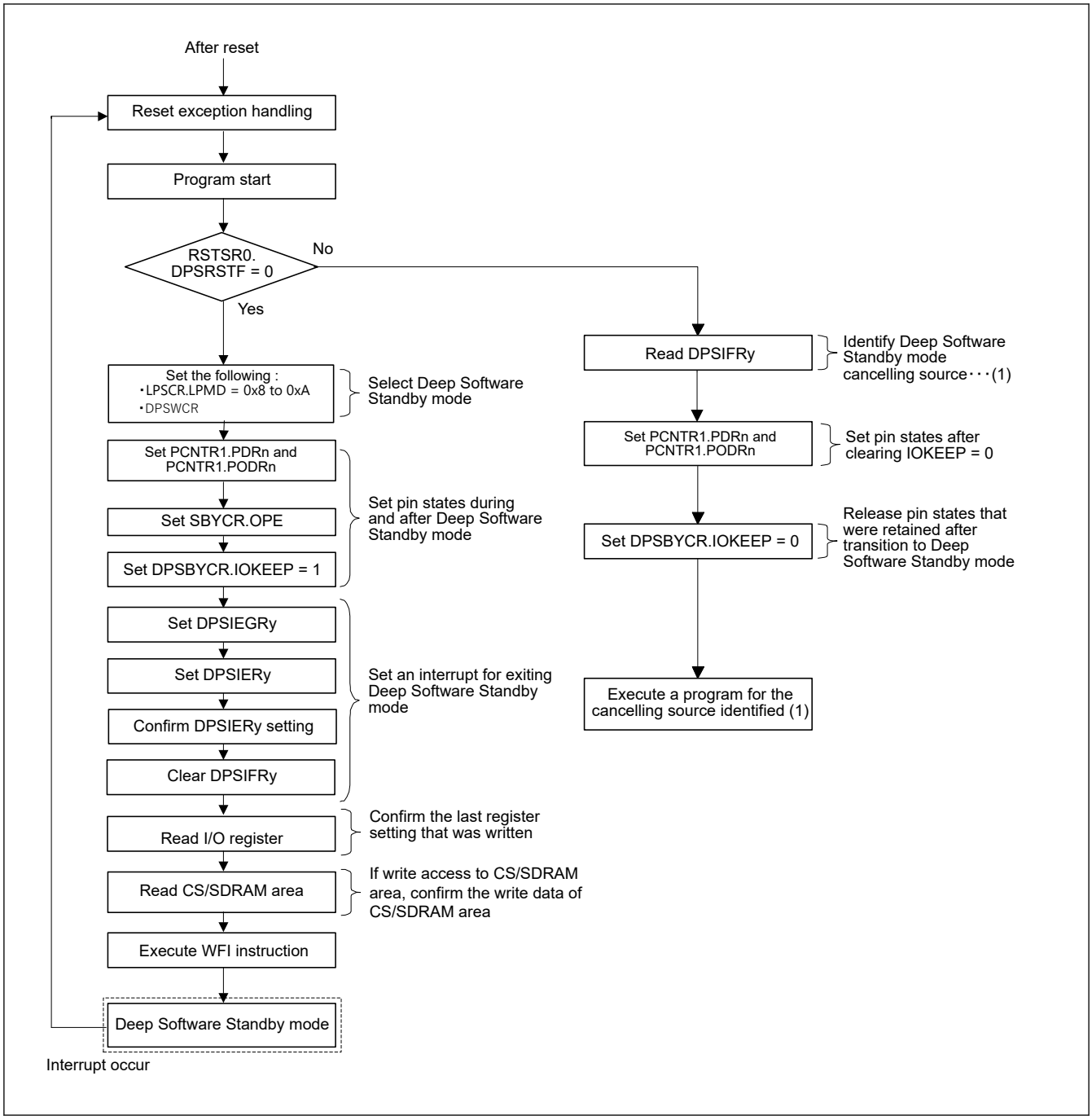


Figure 10.5 Example flow for using Deep Software Standby mode

10.8 Usage Notes

10.8.1 Register Access

(1) Invalid register write accesses during specific modes or transitions:

Do not write to registers under any of the conditions listed in this section.

[Registers]

- All registers with a peripheral name of “SYSTEM”.

[Conditions]

- OPCCR.OPCMTSF = 1 (during transition of operating power control mode)

- FENTRYR.FENTRYi = 1 (i = 0 to 3) or FENTRYR.FENTRYD = 1 (Flash P/E mode, dataFlash P/E mode)
- During the time period from executing a WFI instruction to returning to Normal mode

(2) Valid setting for the clock-related registers

Table 10.11 and Table 10.12 shows the valid setting of the clock related registers in each operating power control mode. Do not write any value other than the valid setting. Also, each register has some prohibited settings under certain condition other than the operating power control modes. See section 8, Clock Generation Circuit for another condition of each register.

Table 10.11 Valid setting of the clock related registers(1)

Mode	Valid settings		
	SCKSCR.CKSEL[2:0] CKOCR.CKOSEL[2:0]	PLLCR.PLLSTP	PLL2CR.PLL2STP
High-speed	000b (HOCO) 001b (MOCO) 010b (LOCO)*2 011b (MOSC) 100b (SOSC) 101b (PLL1P) *1	0 (operating) 1 (stop)	0 (operating) 1 (stop)
Low-speed	000b (HOCO) 001b (MOCO) 010b (LOCO)*2 011b (MOSC) 100b (SOSC)	1 (stop)	1 (stop)

Note 1. SCKSCR.CKSEL[2:0] only
Note 2. CKOCR.CKOSEL[2:0] only

Table 10.12 Valid setting of the clock related registers (2)

Operating oscillator	Valid settings
	OPCCR.OPCM[1:0]
PLL1, PLL2	00b
High-speed on-chip oscillator	00b, 11b
Middle-speed on-chip oscillator	
Main clock oscillator	
Low-speed on-chip oscillator	
Sub-clock oscillator	

(3) Invalid register write accesses by the DTC or DMAC

Do not write to registers listed in this section by the DTC or DMAC.

[Registers]

- MSTPCRA
- MSTPCRB
- MSTPCRC
- MSTPCRD
- MSTPCRE

(4) Invalid write access when PRCR.PRC1 is 0

Write access to the registers listed below is invalid when PRCR.PRC1 bit is 0.

[Registers]

- OPCCR
- PDCTRGD
- PDRAMSCRn

- SBYCR
- SSCR1
- LPSCR
- DPSBYCR
- DPSIERn
- DPSIFRn
- DPSIEGRn
- PLL1LDOSCR
- PLL2LDOSCR
- HOCOLDOSCR
- LVOCR

(5) Invalid write access when PRCR.PRC4 is 0

Write access to the registers listed below is invalid when PRCR.PRC4 bit is 0.

[Registers]

- LPMSAR
- PGCSAR
- DPFSAR
- RSCSAR

10.8.2 I/O Port States

I/O port states in Software Standby mode, Deep Software Standby mode are as same as before entering the modes. Therefore, the supply current is not reduced while output signals are held high.

10.8.3 Module-Stop State of DMAC and DTC

Before writing 1 to MSTPCRA.MSTPA22, clear the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC to 0. For details, see section 16, DMA Controller (DMAC) and section 17, Data Transfer Controller (DTC)

10.8.4 Internal Interrupt Sources

Interrupts do not operate in the module-stop state. If setting module-stop bit while an interrupt request is generated, a CPU interrupt source or a DMAC or DTC startup source cannot be cleared. For this reason, make sure to disable the corresponding interrupts before setting the module-stop bits.

10.8.5 Input Buffer Control by DIRQnE Bit

Setting the DPSIERy.DIRQnE (y = 0 or 1, n = 0 to 15) bit to 1 enables the input buffer of the IRQ0-DS to IRQ15-DS pins. Note that, although inputs to these pins are sent to the DPSIFRy.DIRQnF (y = 0 or 1, n = 0 to 15) bits, they are not sent to the interrupt controller, peripheral modules, and I/O ports.

10.8.6 Transition to Low Power Modes

Since the MCU does not support wakeup by event, do not enter low power modes (CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode, or Deep Software Standby mode) by executing a WFE instruction.

10.8.7 Timing of WFI Instructions

It is possible for the WFI instruction to be executed before I/O register and CS/SDRAM area writes are complete, in which case operation may not be as intended. This can happen if the WFI is placed immediately after a write to a I/O register and CS/SDRAM area. To avoid this problem it is advisable to read back the register and CS/SDRAM area that was written to confirm that the write has completed.

10.8.8 Writing WDT/IWDT Registers by DMAC or DTC

Do not write registers in WDT or IWDT by DMAC or DTC while WDT or IWDT stops by entering processor Low power mode.

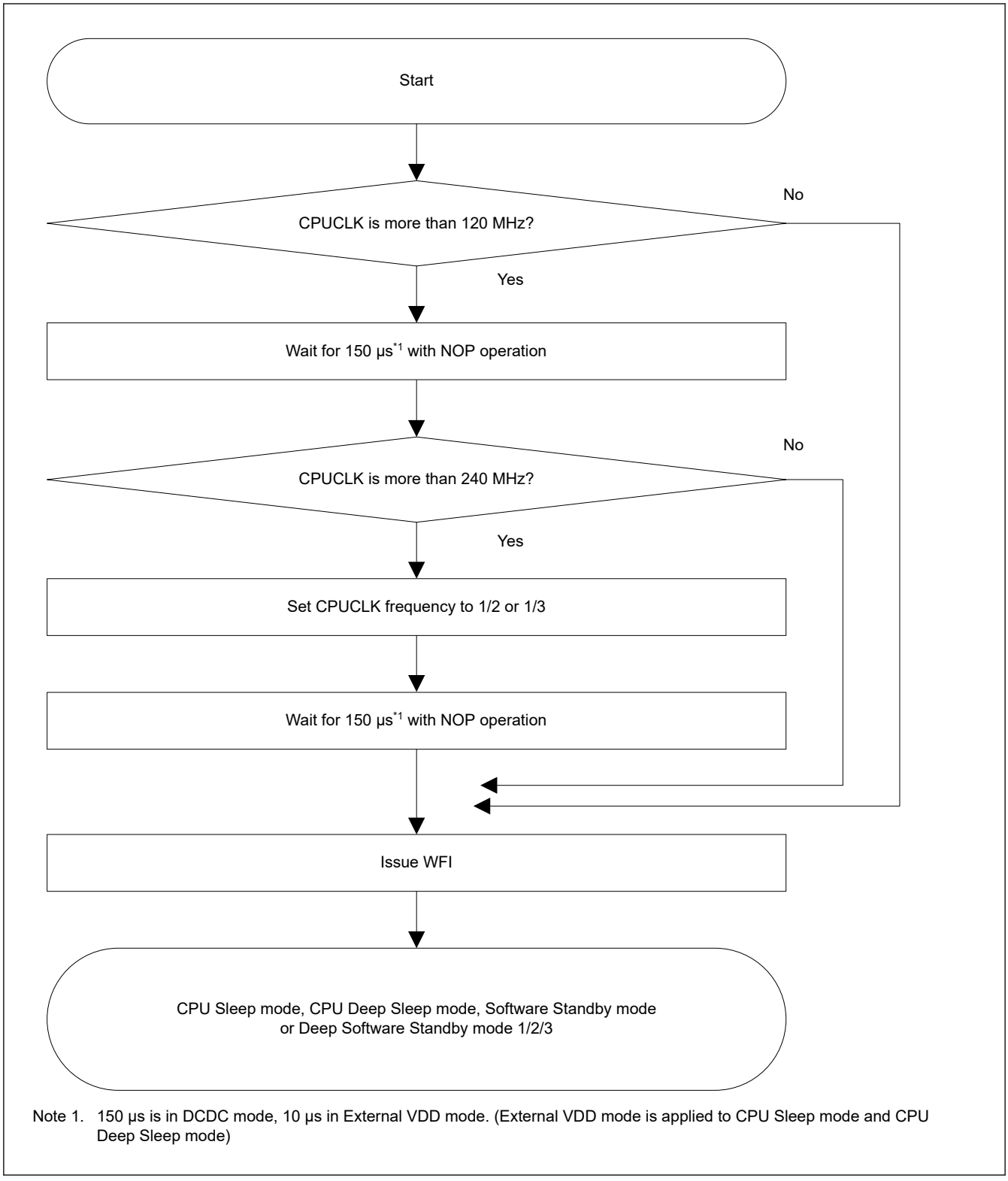
10.8.9 Module Stop Bit Write Timing

It is possible that the access to I/O register may be executed before the corresponding module stop bit write is completed. In this case, the access to I/O register may not proceed as intended. To avoid this issue, before accessing I/O register, read back the module stop bit that was written to confirm that the write completed.

10.8.10 Notes on Transitioning to or Canceling Low Power State

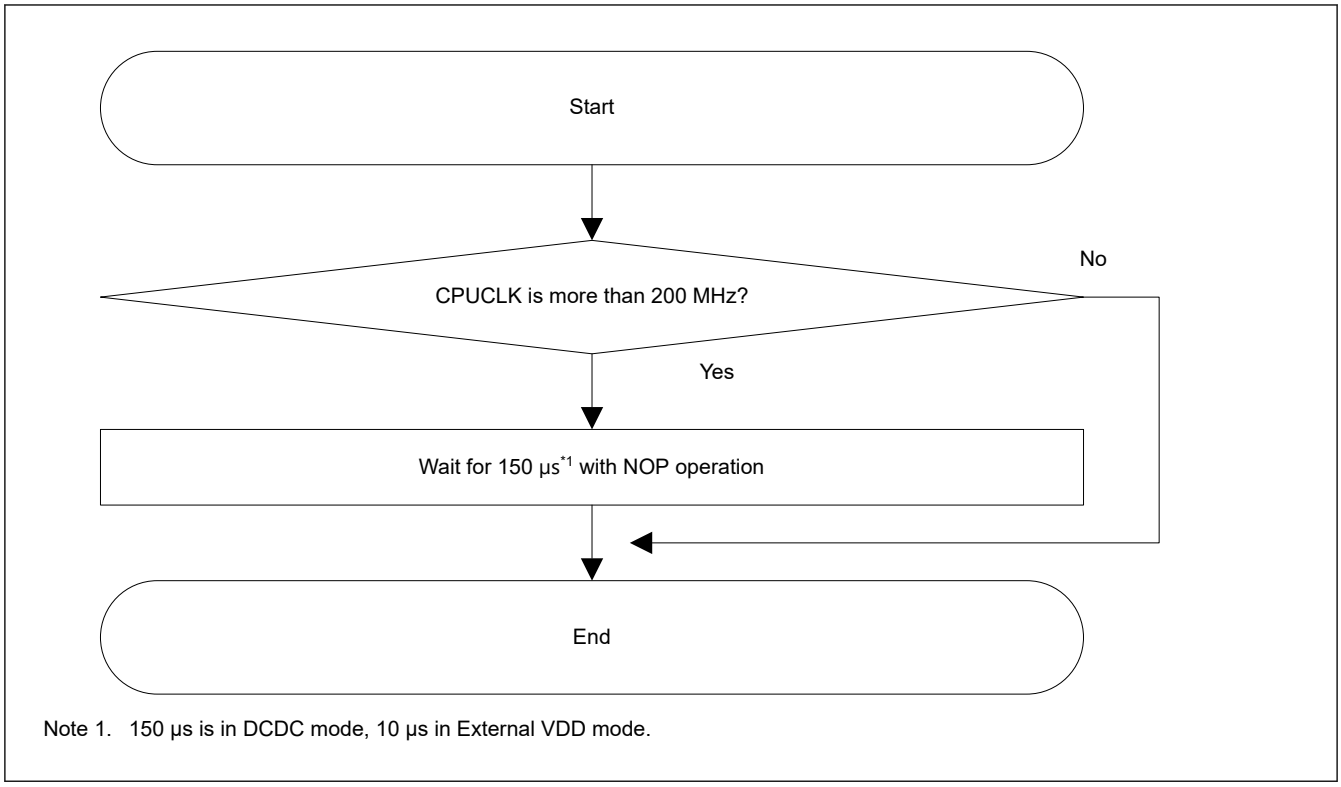
Transitioning to/returning from the low power state requires procedures according to the CPUCLK frequency.

These flows are shown in Figure 10.6, Figure 10.7 and Figure 10.8. It is recommended to use software to measure the wait time. Be sure to consider the worst-case conditions to ensure that the required wait time elapses. If an interrupt is unavoidably generated during the wait time, retry the measurement after return from the interrupt



Note 1. 150 µs is in DCDC mode, 10 µs in External VDD mode. (External VDD mode is applied to CPU Sleep mode and CPU Deep Sleep mode)

Figure 10.6 Additional flow for the transition to CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode and Deep Software Standby mode 1/2/3 according to CPUCLK frequency



Note 1. 150 µs is in DCDC mode, 10 µs in External VDD mode.

Figure 10.7 Additional flow for the return from CPU Deep Sleep mode according to CPUCLK frequency

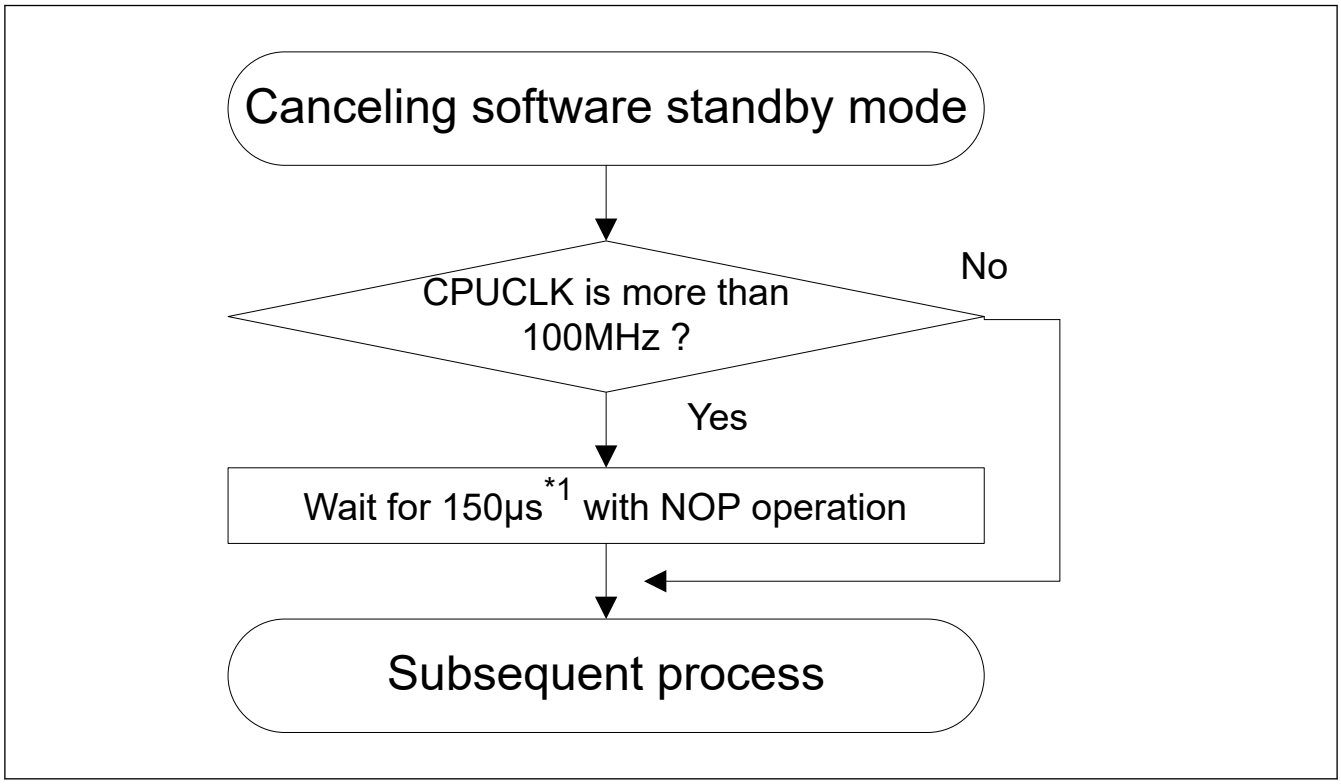


Figure 10.8 Additional flow for the canceling Software Standby mode

10.8.11 Interruption During Transition to Low Power Modes

10.8.11.1 Notes About the Sleep-on-exit Function

Transitioning to low power modes can be done through either WFI instruction or Sleep-on-exit.

When Sleep-on-exit is used for transition to low power modes, the comments described in the User's Manual Hardware for WFI instruction is also applicable to Sleep-on-exit.

10.8.11.2 Applicable Conditions and Notes

(1) Applicable Conditions

When all three conditions are met.

1. Transition to CPU Deep Sleep mode is started by a trigger (WFI instruction or SLEEPONEXIT) with CPU.SCR.SLEEPDEEP = 1 and LPSCR.LPMD is either 0x0 or 0x4 to use Software Standby mode, or CPU Deep Sleep mode.
2. When transition to CPU Deep Sleep mode, ICLK frequency is over 90 MHz and CPUCLK:ICLK = 2:1.
3. During the specified interval (one ICLK cycle) of transitioning to CPU Deep Sleep mode, one of the following interrupt requests that is not an interrupt source to return from CPU Deep Sleep mode is accepted by CPU.
   (a) SysTick interrupt (all the following are applicable):
      - Exception number 15 of the Interrupt vector table.
      - Interrupt requests are not masked by Base Priority Mask Register (BASEPRI) (BASEPARI = 0 or BASEPRI > SHPR3.PRI_15).
   (b) Maskable interrupt requests that are not interrupt sources to return from CPU Deep Sleep mode (all the following are applicable):
      - By WUPEN in exception numbers 16 to 111 in Table 13.3 (those not permitted to return from CPU Deep Sleep mode)
      - Interrupt requests are enabled by Interrupt Set-Enable Register (NVIC_ISERn)
      - Interrupt requests are not masked by Base Priority Mask Register (BASEPRI) (BASEPARI = 0 or BASEPRI > NVIC_IPRn.PRI_N).

(2) Notes

If the above conditions are met, the MCU rarely transits to following unintended states.

These unintended states can be resolved by a reset. If an unintended state occurs, then the next transition to CPU Deep Sleep mode, Software Standby mode, and Deep Software Standby mode will fail.

Adapt a workaround if these unintended states are not acceptable.

1. When transitioning to CPU Deep Sleep mode (LPSCR.LPMD = 0x0 and CPU.SCR.SLEEPDEEP = 1), CLK continues to operate the same as before transitioning to CPU Deep Sleep mode.
   - Only CPU stop operating, current consumption becomes the same level as CPU Sleep mode
   - As for the interrupt source settings to return from CPU Deep Sleep mode, these interrupt requests can return to Normal mode.
2. When transitioning to Software Standby mode (LPSCR.LPMD = 0x4 and CPU.SCR.SLEEPDEEP = 1), CLK continues to operate the same as before transitioning to CPU Deep Sleep mode.
   - Only CPU stop operating, current consumption becomes the same level as CPU Sleep mode
   - As before the transition to Software Standby mode is started and depending on the setting, timer or other peripherals continue to operate, and an interrupt request related to the peripheral is generated
   - Because WDT clock does not stop, a reset or an interrupt for WDT is generated depending on the settings before starting the transition to Software Standby mode.
   - Interrupt requests are held in IR flag (IELSRn)
   - As for the interrupt source settings to return from CPU Deep Sleep mode, these interrupt requests can return to Normal mode.
3. When transitioning to Deep Software Standby mode (LPSCR.LPMD = 0x8 or 0x9 or 0xA and CPU.SCR.SLEEPDEEP = 1).

The transition to Deep Software Standby mode is not possible, and the state shown in "2. When transitioning to Software Standby mode" continues.
As for the interrupt source (DPSIERn) settings to returning from Deep Software Standby mode, these interrupt requests can return to Normal mode only when the interrupt source (WUPENn) to return from CPU Deep Sleep mode is set.

10.8.11.3 Workaround

(1) Workaround

To avoid the unintended states as described, apply either 1. or 2. from the following before the condition to transit to CPU Deep Sleep mode or Software Standby mode are met:

1. Set the frequency setting as CPUCLK: ICLK=1:1.
2. Disable interrupt requests that are not interrupt sources to return from CPU Deep Sleep mode. (For the setting procedure, see (2) Setting procedure for transition to CPU Deep Sleep mode, or Software Standby mode).
   (a) Disable SysTick interrupt requests (exception number 15 of Table 13.3).
   (b) Disable maskable interrupt requests that are not interrupt sources to return from CPU Deep Sleep mode (exception number 16 through 111 of Table 13.3 that WUPENn does not allow to return from CPU Deep Sleep mode).

(2) Setting procedure for transition to CPU Deep Sleep mode, or Software Standby mode

This section describes procedures for avoiding unintended states.

The handling of interrupt requests after returning from CPU Deep Sleep mode or Software Standby mode varies depending on the method used to disable the maskable interrupt request. Either one or the other should be applied.

[Procedure A: Disable maskable interrupt request acceptance]

Any interrupt request that occurs while interrupt request acceptance is disabled, is discarded.

Table 10.13 Before transitioning to CPU Deep Sleep mode, or Software Standby mode

Step	Description
1	Disable the SysTick interrupt request*1.
2	Clear IELSRn in ICU to disable acceptance of maskable interrupt requests that are not interrupt sources to return from CPU Deep Sleep mode.
3	Read IELSRn in ICU to confirm that IELSRn in ICU has been cleared.
4	Transition to Software Standby mode (WFI instruction, SLEEPONEXIT).

Note 1. Disabling a SysTick interrupt request may cause SysTick interrupt request to be delayed by one cycle of SysTick timer without generating the latest SysTick interrupt request.

Table 10.14 After returning from CPU Deep Sleep mode or Software Standby mode

Step	Description
1	Enable the SysTick interrupt request.
2	Set IELSRn in ICU to enable acceptance of maskable interrupt requests that are not interrupt sources to return from CPU Deep Sleep mode.

[Procedure B: Disable the maskable interrupt request]

The interrupt request generated while the interrupt request is disabled, is retained in IELSRn.IR flag. Therefore, after returning from CPU Deep Sleep mode, Software Standby and enabling the maskable interrupt, it is possible to process the interrupt.

Table 10.15 Before transitioning to Software Standby, Snooze, or Deep Software Standby mode (1 of 2)

Step	Description
1	Disable the SysTick interrupt request*1.

Table 10.15 Before transitioning to Software Standby, Snooze, or Deep Software Standby mode (2 of 2)

Step	Description
2	Write 1 to the corresponding bit in NVIC_ICERn in CPU to disable maskable interrupt requests that are not interrupt sources to return from CPU Deep Sleep mode.
3	Execute Data Synchronization Barrier (DSB) instruction.
4	Transition Software Standby mode (WFI instruction, SLEEPONEXIT).

Note 1. Disabling a SysTick interrupt request may cause SysTick interrupt request to be delayed by one cycle of SysTick timer without generating the latest SysTick interrupt request.

Table 10.16 After returning from CPU Deep Sleep mode or Software Standby mode

Step	Description
1	Enable the SysTick interrupt request.
2	Write 1 to the corresponding bit in NVIC_ISERn in CPU to enable maskable interrupt requests that are not interrupt sources to return from CPU Deep Sleep mode.

11. Battery Backup Function

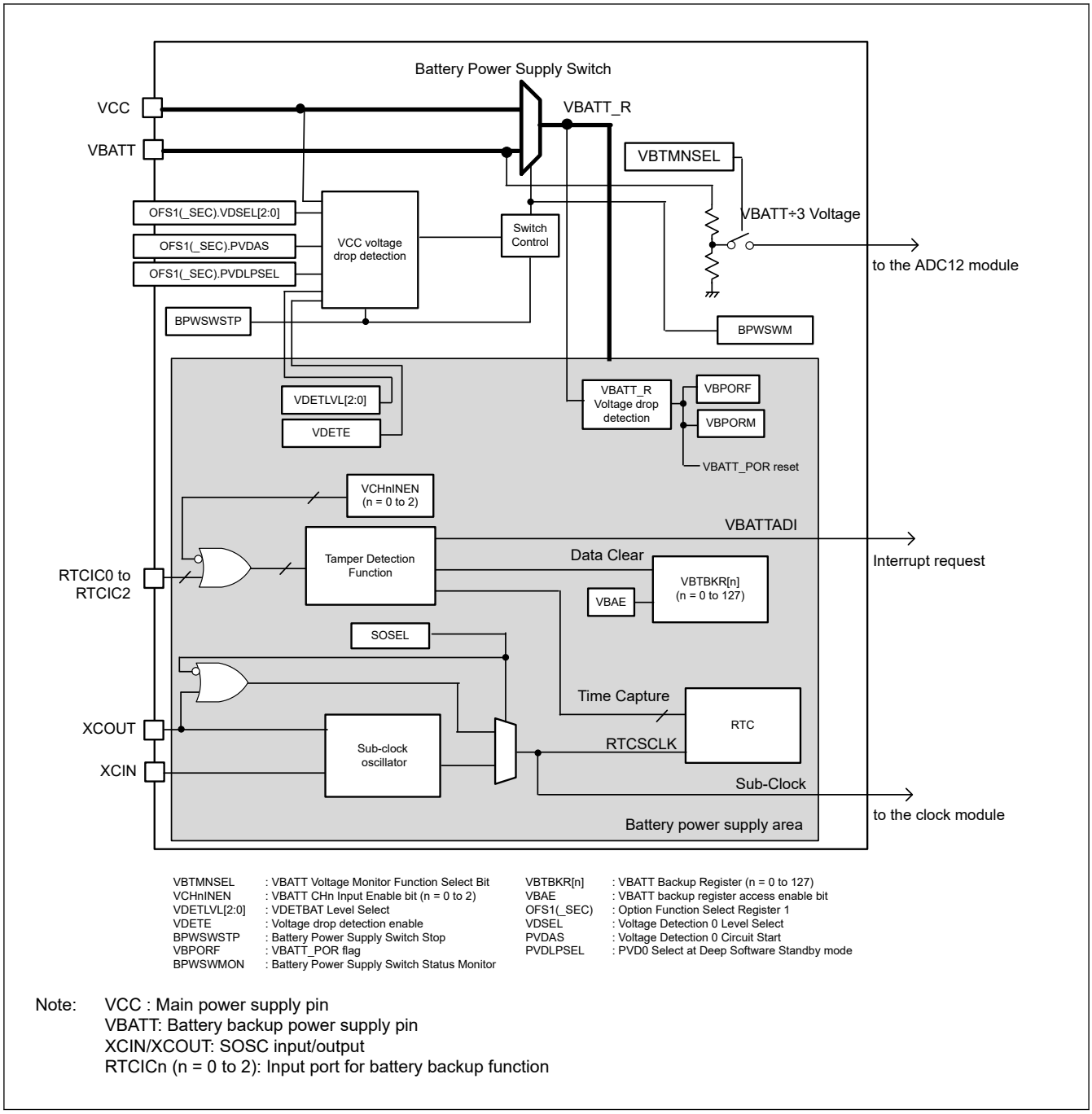
11.1 Overview

The MCU provides a battery backup function that maintains partial battery powering in the event of a power loss. Switching between VCC and VBATT, the battery-powered area includes RTC, SOSC, backup register, tamper detection and VBATT_R voltage drop detection. VBATT_R is the output voltage of the battery power supply switch. This is the power supply of the battery powered area.

During normal operation, the battery-powered area is powered by the main power supply, the VCC pin. When a VCC voltage drop is detected, the power source switches to the dedicated battery backup power pin, the VBATT pin. When the voltage rises again, the power source switches back from VBATT to VCC.

11.1.1 Battery backup function block diagram

Figure 11.1 shows the configuration of the battery backup function.



Note: VCC : Main power supply pin
VBATT: Battery backup power supply pin
XCIN/XCOUT: SOSC input/output
RTCICn (n = 0 to 2): Input port for battery backup function

Figure 11.1 Battery Backup Function block diagram

11.1.2 Features of Battery Backup Function

The features include:

- Battery power supply switch
- VBATT_R voltage drop detection function
- Backup registers
- Time capture pin detection
- Tamper Detection function
- VBATT voltage monitor function

11.1.3 Battery Power Supply Switch

When the voltage applied to the VCC pin drops, this feature switches the power supply from the VCC pin to the VBATT pin. When the voltage rises, it switches the power supply from the VBATT pin back to the VCC pin.

When VCC is lower than VDETBATT, and VCC is higher than VBAT, the injected current connects from the VCC to the VBATT pin through an internal diode. If the power supply battery connected to the VBATT pin cannot support this current injection, for example if the battery is not rechargeable, Renesas strongly recommends that you connect through a low-voltage threshold diode between the power supply battery and the VBATT pin.

It is necessary to enable voltage monitor 0 resets to use the battery backup function. The voltage monitor 0 level must be higher than the VBATT switch level.

11.1.4 VBATT_R voltage drop detection function

VBATT_R voltage drop detection function supports the battery-powered area. This function monitors the VBATT_R voltage level. VBATT_R is the output voltage of the battery power supply switch. This voltage drop detection causes a VBATT_POR reset and initializes the battery-powered area. See details in each register description. The VBATT status register includes a flag to check for this voltage drop detections.

11.1.5 Backup Registers

The battery-powered area provides 128-byte backup registers. These registers retain data when VBATT is supplied and VCC is powered off. When tampering is detected, data of backup register can be cleared to 0x00.

11.1.6 Time Capture Function

The RTC detects RTCICn (n = 0 to 2) pin input level change. For more information, see section 24, Realtime Clock (RTC).

11.1.7 Tamper Detection Function

The tamper detection function detects the RTCICn (n = 0 to 2) pin input event. The input event is defined as a change of RTCICn (n = 0 to 2) pin input level. The tamper detection flag is set to 1 by the input event . When interrupt is enabled and flag is set to 1, tamper detection interrupt is generated. When backup registers clear is enabled and flag is set to 1, the data of backup registers is cleared. Time Capture Function can select this flag as the source of the time capture trigger.

11.1.8 VBATT Voltage Monitor Function

VBATT voltage monitor function is to monitor the input voltage level to the VBATT pin. The voltage level can be monitored as analog signal. For more information, see section 45, 12-Bit A/D Converter (ADC12).

11.2 Register Descriptions

11.2.1 BBFSAR : Battery Backup Function Security Attribute Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x3D0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	NONSEC4	NONSEC3	NONSEC2	NONSEC1	NONSEC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NONSEC0	Non-secure Attribute bit 0 Target register: VBATTMNSELR 0: Secure 1: Non-secure	R/W
1	NONSEC1	Non-secure Attribute bit 1 Target register: VBTBER 0: Secure 1: Non-secure	R/W
2	NONSEC2	Non-secure Attribute bit 2 Target register: VBTICTLR, VBTICTLR2, VBTIMONR 0: Secure 1: Non-secure	R/W
3	NONSEC3	Non-secure Attribute bit 3 Target register: VBTBPCR1, VBTBPCR2, VBTBPSR 0: Secure 1: Non-secure	R/W
4	NONSEC4	Non-secure Attribute bit 4 Target register: VBTADSR,VBTADCR1, VBTADCR2 0: Secure 1: Non-secure	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1
Note: This register is write-protected by PRCR register.

The BBFSAR register controls the secure attribute of the battery backup function registers.

NONSEC0 bit (Non-secure Attribute bit 0)

This bit controls the security attribute of VBATTMNSELR.

NONSEC1 bit (Non-secure Attribute bit 1)

This bit controls the security attribute of VBTBER.

NONSEC2 bit (Non-secure Attribute bit 2)

This bit controls the security attribute of VBTICTLR, VBTICTLR2, and VBTIMONR.

NONSEC3 bit (Non-secure Attribute bit 3)

This bit controls the security attribute of VBTBPCR1 and VBTADCR2.

NONSEC4 bit (Non-secure Attribute bit 4)

This bit controls the security attribute of VBTADSR, VBTADCR1 and VBTADCR2.

11.2.2 VBRSABAR : VBATT Backup Register Security Attribute Boundary Address Register

Base address: SYSC = 0x4001_E000

Offset address: 0x3B0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SABA[15:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SABA[15:0]	Boundary address between secure and Non-secure	R/W

Note: S-TYPE-6, P-TYPE-2

VBRSABAR specify the boundary address between Secure and Non-secure regions of VBATT Backup Register. This register specifies lower 16 bits of VBATT Backup Register address. Secure region is less than SABA. Non-secure region is SABA or higher. The boundary address can set in units of 32 byte, so the value written from b4 to b0 should be 0. SABA has no effect other than VBATT backup register.

VBATT Backup register is separated as follows.

Secure region : 0x4001_ED00 ≤ Address < 0x4001_0000 + SABA

Non-secure region : 0x4001_0000 + SABA ≤ Address

The initial value specifies an address greater than the end address of VBATT Backup Register, so all area is Secure.

If you specify all area of the VBATT Backup Register as non-secure, specify the top address in the VBATT Backup Register area.

11.2.3 VBRPABARS : VBATT Backup Register Privilege Attribute Boundary Address Register for Secure Region

Base address: SYSC = 0x4001_E000

Offset address: 0x3B4

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PABAS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PABAS[15:0]	Boundary address between privileged and unprivileged.	R/W

Note: S-TYPE-6, P-TYPE-2

VBRPABARS specify the boundary address between Privileged and Unprivileged areas in Secure region of VBATT Backup Register. This register further separates the Secure region of VBATT Backup Register set by VBRSABAR register into Privileged and Unprivileged areas. This register specifies lower 16 bits of VBATT Backup Register address. Privileged area is less than PABAS. Unprivileged area is PABAS or higher. The boundary address can set in units of 32byte, so the value written from b4 to b0 should be 0. PABAS has no effect other than VBATT backup register.

VBATT Backup register is separated as follows.

Privileged area in Secure region : 0x4001_ED00 ≤ Address < 0x4001_0000 + PABAS
0x4001_ED00 ≤ Address < 0x4001_0000 + SABA

This area that satisfies both address conditions.

Unprivileged area in Secure region : 0x4001_0000 + PABAS ≤ Address < 0x4001_0000 + SABA

The initial value specifies before the top address of the VBATT Backup Register, so all the Secure region is Unprivileged. In this case, the setting of this register is actually ignored.

If you specify Privileged for the entire VBATT Backup Register that you have set as the Secure region, specify VBRPABARS register to the 0xFFE0 .

This is to narrow the verification space of the target specification base verification.

11.2.4 VBRPABARNS : VBATT Backup Register Privilege Attribute Boundary Address Register for Non-secure Region

Base address: SYSC_NS = 0x5001_E000

Offset address: 0x3B8

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PABANS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PABANS[15:0]	Boundary address between privileged and unprivileged.	R/W

Note: S-TYPE-7, P-TYPE-2

VBRPABARNS specify the boundary address between Privileged area and Unprivileged area in Non-secure region of VBATT Backup Register. This register further separates the Non-secure region of VBATT Backup Register set by VBRSABAR register into Privileged and Unprivileged areas. This register specifies lower 16 bits of VBATT Backup Register address. Privileged area is less than PABANS. Unprivileged area is PABANS or higher. The boundary address can set in units of 32byte, so the value written from b4 to b0 should be 0. PABANS has no effect other than VBATT backup register.

VBATT Backup register is separated as follows.

Privileged area in Secure region :	0x5001_0000 + SABA ≤ Address < 0x5001_0000 + PABANS
Unprivileged area in Secure region :	0x5001_0000 + PABANS ≤ Address 0x5001_0000 + SABA ≤ Address This area that satisfies both address conditions

The initial value specifies before the top address of VBATT Backup Register, so all the Non-secure region is Unprivileged. In this case, the setting of this register is actually ignored.

If you specify Privileged for the entire VBATT Backup Register that you have set as the Non-secure region, specify VBRPABARNS register to the 0xFFE0 .

This is to narrow the verification space of the target specification base verification

11.2.5 VBATTMNSELR : Battery Backup Voltage Monitor Function Select Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA84

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	VBTMNSEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VBTMNSEL	VBATT Voltage Monitor Function Select Bit 0: Disables VBATT voltage monitor function 1: Enables VBATT voltage monitor function	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

VBATTMNSELR is the register which controls VBATT voltage monitor function.

This register is initialized by all reset sources except VBATT_POR reset.

VBTMNSEL bit (VBATT Voltage Monitor Function Select Bit)

Select VBATT low voltage monitor function. After setting this bit to 1, it is necessary to wait t_{MONWT} for the monitor level to stabilize.

Consumption current increases while VBTMNSEL = 1. So, after monitoring the VBATT voltage level, clear VBTMNSEL to 0 in order to reduce power consumption of VBATT power supply.

For more information on t_{MONWT}, see section 60, Electrical Characteristics

11.2.6 VBTBER : VBATT Backup Enable Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC40

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	VBAE	—	—	—
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	VBAE	VBATT backup register access enable bit 0: Disable to access VBTBKR[n] 1: Enable to access VBTBKR[n]	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

VBAE bit is initialized by the power on reset of VCC or after transition to VBATT mode.

VBAE bit (VBATT backup register access enable bit)

You must write 1 to VBAE before accessing VBTBKR[n] (n = 0 to127) and you must write 0 to VBAE after finishing all access (write or read) to VBTBKR[n]. If you do not write 0 to VBAE, the data of VBTBKR[n] is not kept in VBATT mode.

Also, while VBAE is set to 0, the value of VBTBKR[n] can be retained even if the VDD power supply and VCC power supply are powered off.

To access VBTBKR[n], wait for at least 500 ns after writing 1 to VBAE, and then access VBTBKR[n].

Before entering the Deep Software Standby mode, you must write 0 to VBAE.

Wait at least 250 ns after writing 0 to VBAE, before entering the Deep Software Standby mode.

If it is not used VBTBKR[n], It should be changed VBAE to 0 to reduce power consumption of VBTBKR[n].

11.2.7 VBTBKR[n] : VBATT Backup Register (n = 0 to 127)

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xD00 + 0x001 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:								
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	n/a	VBATT Backup Register The value of this register is retained even in VBATT mode. This register is not initialized by any reset sources.	R/W

Note: S-TYPE-3, P-TYPE-3

VBTBKR[n] bits (VBATT Backup Register)

The value of this register is retained even in VBATT mode.

You can use 32 ,16 or 8-bit access instruction when accessing the VBTBKR[n] register. When accessing, please note that the byte order of the data stored in the VBTBKR[n] register is little endian.

The data of VBTBKR[n] register is cleared to 0x00 by VBATT_POR reset or tamper detection function.

Table 11.1 Address of VBATT Backup Register for Secure Region

Address	Symbol
0x4001_ED00 to 0x4001_ED0F	VBTBKR[0] to VBTBKR[15]
0x4001_ED10 to 0x4001_ED1F	VBTBKR[16] to VBTBKR[31]
0x4001_ED20 to 0x4001_ED2F	VBTBKR[32] to VBTBKR[47]
0x4001_ED30 to 0x4001_ED3F	VBTBKR[48] to VBTBKR[63]
0x4001_ED40 to 0x4001_ED4F	VBTBKR[64] to VBTBKR[79]
0x4001_ED50 to 0x4001_ED5F	VBTBKR[80] to VBTBKR[95]
0x4001_ED60 to 0x4001_ED6F	VBTBKR[96] to VBTBKR[111]
0x4001_ED70 to 0x4001_ED7F	VBTBKR[112] to VBTBKR[127]

11.2.8 VBTICTLR : VBATT Input Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC4C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	VCH2INEN	VCH1INEN	VCH0INEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VCH0INEN	VBATT CH0 Input Enable 0: RTCIC0 input disable 1: RTCIC0 input enable	R/W
1	VCH1INEN	VBATT CH1 Input Enable 0: RTCIC1 input disable 1: RTCIC1 input enable	R/W
2	VCH2INEN	VBATT CH2 Input Enable 0: RTCIC2 input disable 1: RTCIC2 input enable	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

VBTICTLR is the register that can select RTCICn (n = 0 to 2) pin as input. This register is only initialized by VBATT_POR reset.

VCHnINEN bits (VBATT CHn Input Enable Bits) (n = 0 to 2)

The VCHnINEN bits enable the input direction on the RTCICn. A wait time of 50 µs is required for the operation to stabilize after the RTCICn pin input is enabled. If VCC voltage of the MCU is lower than 1.8V and these bits are set to 1, RTC registers and VBTBKR[n] (n = 0 to127) can be accessed 20 µs after the setting of this bit.

11.2.9 VBTICTLR2 : VBATT Input Control Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC4D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	VCH2EG	VCH1EG	VCH0EG	—	VCH2NCE	VCH1NCE	VCH0NCE
Value after reset:	0	1	1	1	0	0	0	0

Bit	Symbol	Function	R/W
0	VCH0NCE	VBATT CH0 Input Noise Canceler Enable 0: RTCIC0 pin input noise canceler disable 1: RTCIC0 pin input noise canceler enable	R/W
1	VCH1NCE	VBATT CH1 Input Noise Canceler Enable 0: RTCIC1 pin input noise canceler disable 1: RTCIC1 pin input noise canceler enable	R/W
2	VCH2NCE	VBATT CH2 Input Noise Canceler Enable 0: RTCIC2 pin input noise canceler disable 1: RTCIC2 pin input noise canceler enable	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	VCH0EG	VBATT CH0 Input Edge Select 0: RTCIC0 pin input event is detected on falling edge 1: RTCIC0 pin input event is detected on rising edge	R/W
5	VCH1EG	VBATT CH1 Input Edge Select 0: RTCIC1 pin input event is detected on falling edge 1: RTCIC1 pin input event is detected on rising edge	R/W
6	VCH2EG	VBATT CH2 Input Edge Select 0: RTCIC2 pin input event is detected on falling edge 1: RTCIC2 pin input event is detected on rising edge	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

VBTICTLR2 is the register that can select RTCICn (n = 0 to 2) pin input mode. This register is only initialized by VBATT_POR reset.

VCHnNCE bits (VBATT CHn Input Noise Canceler Enable) (n = 0 to 2)

The VCHnNCEN bit enables the input noise canceler on the RTCICn (n = 0 to 2) pin input.

VCHnEG bits (VBATT CHn Input Edge Select) (n = 0 to 2)

The VCHnEG bit selects input event detection edge to set VBTADFn flag.

11.2.10 VBTIMONR : VBATT Input Monitor Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC4E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	VCH2MON	VCH1MON	VCH0MON
Value after reset:	0	0	0	0	0	x	x	x

Bit	Symbol	Function	R/W
0	VCH0MON	VBATT CH0 Input monitor 0: RTCIC0 pin input is low level 1: RTCIC0 pin input is high level.	R
1	VCH1MON	VBATT CH1 Input monitor 0: RTCIC1 pin input is low level 1: RTCIC1 pin input is high level	R
2	VCH2MON	VBATT CH2 Input monitor 0: RTCIC2 pin input is low level 1: RTCIC2 pin input is high level.	R
7:3	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-2

VBTIMONR is the register that can monitor RTCICn (n = 0 to 2) pin input level.

VCHnMON bits (VBATT CHn Monitor) (n = 0 to 2)

The VCHnMON bit indicates input level on the RTCICn (n = 0 to 2) pin.

11.2.11 VBTBPCR1 : VBATT Battery Power Supply Control Register 1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA88

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	BPWSWSTP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BPWSWSTP	Battery Power Supply Switch Stop 0: Battery power supply switch enable 1: Battery power supply switch stop	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VBTBPCR1 register controls battery power supply switch. This register is initialized by all reset sources except Deep Software Standby reset and VBATT_POR reset.

BPWSWSTP bit (Battery Power Supply Switch Stop)

The BPWSWSTP bit can enable switching the backup module supply power source from VCC to VBATT when the voltage applied to the VCC pin drops.

When stop is selected, the battery backup module power supply is always from VCC. Set the VDETE bit to 0 after setting theBPWSWSTP bit to 1. The BPWSWSTP bit must not be set from 1 to 0 while the VDETE bit is 1. If this setting is made, the state of the backup power area cannot be guaranteed.

11.2.12 VBTBPCR2 : VBATT Battery Power Supply Control Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC45

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	VDETE	—	VDETLVL[2:0]		
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Function	R/W
2:0	VDETLVL[2:0]	V_{DETBAT} Level Select 0 0 0: 2.80 V 0 0 1: 2.53 V 0 1 0: 2.10 V 0 1 1: 1.95 V 1 0 0: 1.85 V 1 0 1: 1.75 V 1 1 0: setting prohibited 1 1 1: setting prohibited	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	VDETE	Voltage drop detection enable 0: VCC Voltage drop detection disable 1: VCC Voltage drop detection enable	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VBTBPCR2 register controls battery power supply switch. This register is only initialized by VBATT_POR reset.

VDETLVL[2:0] bits (V_{DETBAT} Level Select)

The VDETLVL[2:0] bit selects V_{DETBAT} level. When the voltage of the VCC pin drops below V_{DETBAT}, the power supply source switches from the VCC pin to the VBATT pin. The V_{DETBAT} level should be below Voltage monitoring 0 level.

This MCU can reduce power consumption of the Deep Software Standby mode 1 or 2. When the MCU enters the Deep Software Standby mode 1 or 2 and the low power consumption function of voltage monitor 0 is enabled by OFS1(_SEC).PVDLPSEL bit and OFS1(_SEC).PVDAS bit, the battery power supply switch is controlled by the voltage monitor 0.

The VDETLVL[2:0] bit should change while the VDETE bit is 0. The VCC voltage detection function needs t_{DETWT} wait time for stabilization at the change of V_{DETBAT} level. The VDETE bit should be set to 1 after this wait time.

For details on VDETLVL[2:0] bit, see section 11.3.2. VBATT Battery Power Supply Switch Usage.

For more information on t_{DETWT}, see section 60, Electrical Characteristics.

VDETE bit (Voltage drop detection enable)

The VDETE bit enables the VCC voltage drop detection function for the battery power supply switch. The initial value of the VDETE bit is 0, at this time, the VCC voltage drop detection function for the battery power supply switch is disabled. If you use the battery power supply switch function, you must select V_{DETBAT} level and enables the VCC voltage drop detection function.

For details on VDETE bit, see section 11.3.2. VBATT Battery Power Supply Switch Usage.

11.2.13 VBTBPSR : VBATT Battery Power Supply Status Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC46

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	BPWSWM	VBPORM	—	—	—	VBPORF
Value after reset:	0	0	x	x	0	0	0	x

Bit	Symbol	Function	R/W
0	VBPORF	VBATT_POR Flag 0: VBATT_R voltage drop is not detected 1: VBATT_R voltage drop is detected	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
4	VBPORM	VBATT_POR Monitor 0: VBATT_R voltage < $V_{PDR\ (BATR)}$ 1: VBATT_R voltage > $V_{PDR\ (BATR)}$	R
5	BPWSWM	Battery Power Supply Switch Status Monitor 0: VCC voltage < $V_{DETBATT\_m}$ 1: VCC voltage > $V_{DETBATT\_m}$	R
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VBTBPSR register indicate battery power supply status.

VBPORF bit (VBATT_POR Flag)

The VBPORF flag indicates that VBATT_R voltage drop was detected and VBATT_POR reset was asserted.

[Setting condition]

- When VBATT_R voltage drops below $V_{PDR\ (BATR)}$.

[Clearing conditions]

- When 0 is written.

VBPORM bit (VBATT_POR Monitor)

The VBPORM bit indicates comparison result between VBATT_R and $V_{PDR\ (BATR)}$.

BPWSWM bit (Battery Power Supply Switch Status Monitor)

The BPWSWM bit indicates comparison result between VCC and $V_{DETBATT\_m}$ (m = 0 to 6).

11.2.14 VBTADSR : VBATT Tamper detection Status Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC48

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	VBTADF2	VBTADF1	VBTADF0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VBTADF0	VBATT Tamper Detection flag 0 0: RTCIC0 input edge is not detected 1: RTCIC0 input edge is detected	R/W
1	VBTADF1	VBATT Tamper Detection flag 1 0: RTCIC1 input edge is not detected 1: RTCIC1 input edge is detected	R/W
2	VBTADF2	VBATT Tamper Detection flag 2 0: RTCIC2 input edge is not detected 1: RTCIC2 input edge is detected	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VBTADSR register indicate tamper detection function status. This register is only initialized by VBATT_POR reset.

VBTADFn flags (VBATT Tamper Detection flag n) (n = 0, 1, 2)

The VBTADFn (n = 0, 1, 2) flag indicates that RTCICn (n = 0, 1, 2) input edge is detected. The edge type of RTCICn (n = 0, 1, 2) input can be selected by VCHnEG (n = 0, 1, 2) bit.

[Setting condition]

- When RTCICn (n = 0, 1, 2) input edge is detected.

[Clearing conditions]

- When 0 is written after 1 is read from VBTADFn (n = 0, 1, 2) flag.

11.2.15 VBTADCR1 : VBATT Tamper detection Control Register 1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC49

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	VBTADCE2	VBTADCE1	VBTADCE0	—	VBTADIE2	VBTADIE1	VBTADIE0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VBTADIE0	VBATT Tamper Detection Interrupt Enable 0 0: Interrupt by VBTADF0 flag is disable 1: Interrupt by VBTADF0 flag is enable	R/W
1	VBTADIE1	VBATT Tamper Detection Interrupt Enable 1 0: Interrupt by VBTADF1 flag is disable 1: Interrupt by VBTADF1 flag is enable	R/W
2	VBTADIE2	VBATT Tamper Detection Interrupt Enable 2 0: Interrupt by VBTADF2 flag is disable 1: Interrupt by VBTADF2 flag is enable	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	VBTADCE0	VBATT Tamper Detection Backup Register Clear Enable 0 0: Clear Backup Register by VBTADF0 flag is disable 1: Clear Backup Register by VBTADF0 flag is enable	R/W
5	VBTADCE1	VBATT Tamper Detection Backup Register Clear Enable 1 0: Clear Backup Register by VBTADF1 flag is disable 1: Clear Backup Register by VBTADF1 flag is enable	R/W
6	VBTADCE2	VBATT Tamper Detection Backup Register Clear Enable 2 0: Clear Backup Register by VBTADF2 flag is disable 1: Clear Backup Register by VBTADF2 flag is enable	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VBTADCR1 register control tamper detection functions. This register is only initialized by VBATT_POR reset.

VBTADIEn bits (VBATT Tamper Detection Interrupt Enable) (n = 0, 1, 2)

The VBTADIEn (n = 0, 1, 2) bit enables Tamper detection interrupt.

VBTADCEn bits (VBATT Tamper Detection Backup Register Clear Enable) (n = 0, 1, 2)

The VBTADCEn (n = 0, 1, 2) bit enables backup register clearing by tamper detection flag.

11.2.16 VBTADCR2 : VBATT Tamper detection Control Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC4A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	VBRTCES2	VBRTCES1	VBRTCES0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VBRTCES0	VBATT RTC Time Capture Event Source Select 0 0: RTCIC0 1: VBTADF0	R/W
1	VBRTCES1	VBATT RTC Time Capture Event Source Select 1 0: RTCIC1 1: VBTADF1	R/W
2	VBRTCES2	VBATT RTC Time Capture Event Source Select 2 0: RTCIC2 1: VBTADF2	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VBTADCR2 register control tamper detection functions. This register is only initialized by VBATT_POR reset.

VBRTCESn bits (VBATT RTC Time Capture Event Source Select) (n = 0, 1, 2)

The VBRTCESn (n = 0, 1, 2) bit selects RTC time capture event source.

11.3 Operation

11.3.1 Battery Backup Function

When the voltage on the VCC pin drops, power can be supplied to the VBATT_R backup power area from the VBATT pin. The power supply from the VCC pin is resumed when the voltage on the VCC pin exceeds $V_{DETBATT\_m}$ (m = 0 to 6). When a drop of power supply from VCC pin is detected, backup power area enter the VBATT Mode. In the VBATT Mode, power supply is switched to VBATT pin. The power supply from the VCC pin is resumed when the voltage on the VCC pin exceeds $V_{DETBATT\_m}$ (m = 0 to 6). This power supply change does not affect the VBATT_R backup power area function.

It is necessary to enable voltage monitor 0 reset to use the battery backup

The VBATT_R backup power area include following functions:

- RTC (including time capture detection, triggered by a change of the time capture pin input level)
- Sub-clock oscillator (including XCIN and XCOUT pins)
- VBATT Backup Register
- Tamper Detection Function
- VBATT Voltage Monitor Function

Table 11.2 shows the operating states in VBATT mode.

Table 11.2 Operating States in VBATT Mode (1 of 2)

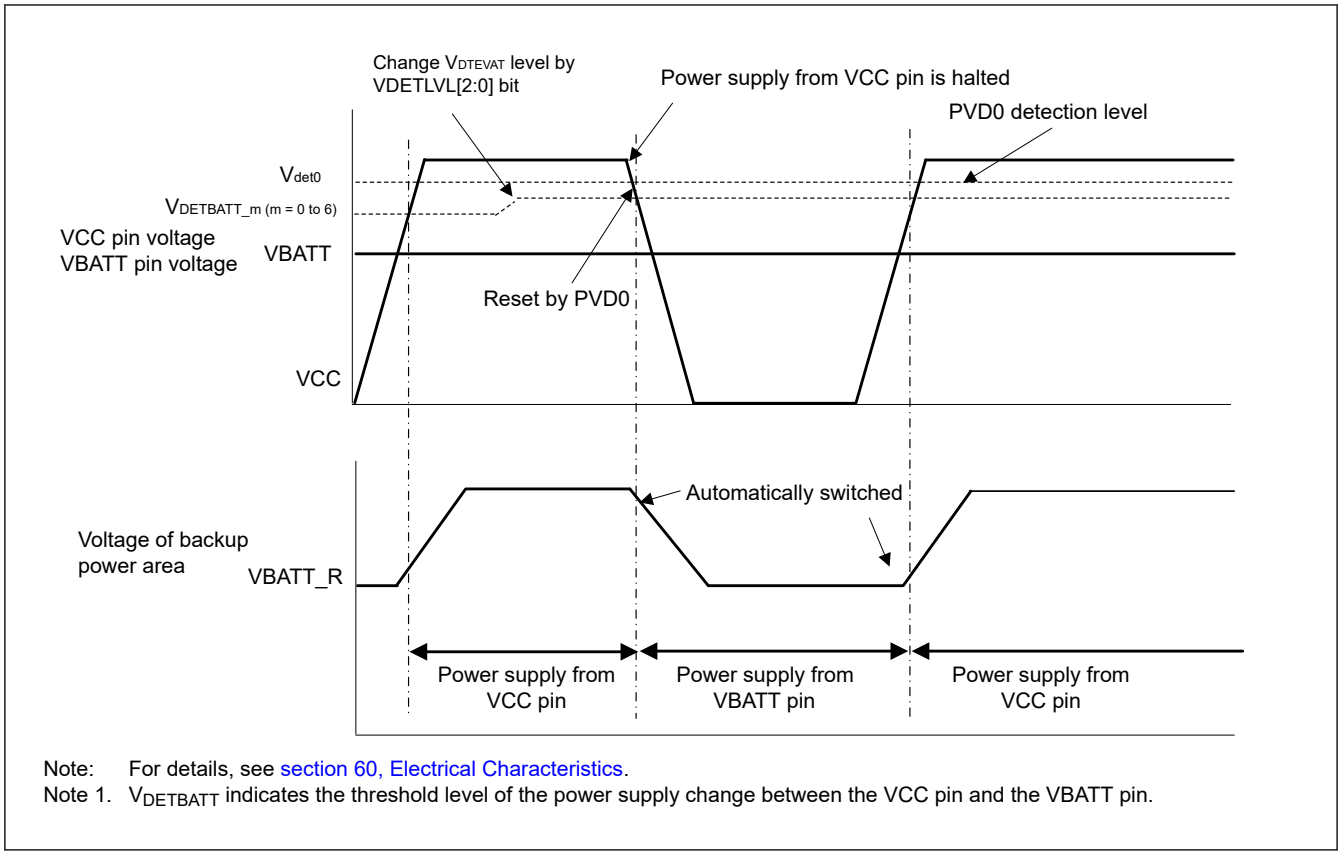
Operating state	VBATT Mode
Transition condition	Detection of VCC voltage drop
Canceling method other than reset	Detection of VCC voltage rise
State after cancellation by an interrupt	—

Table 11.2 Operating States in VBATT Mode (2 of 2)

Operating state	VBATT Mode
State after cancellation by a reset	—
Main clock oscillator	Stop
Sub-clock oscillator	Operating or not operating can be selected by SOSCCR.SOSTP bit. The status of the oscillator is same as before entering VBATT mode.
High-speed on-chip oscillator	Stop
Middle-speed on-chip oscillator	Stop
Low-speed on-chip oscillator	Stop
PLL1	Stop
PLL2	Stop
CPU	Stop (Undefined)
SRAM (ECC RAM included)	Stop (Undefined)
Standby SRAM	Stop (Undefined)
VBATT Backup Register	Stop (Retained or zeroized is selectable when Tamper input was detected.)
Flash memory	Stop (Retained)
Realtime clock (RTC)	Selectable when selecting clock which is operating as the count source.
Programmable voltage detection circuit (PVD)	Stop
Power-on reset circuit	Stop
Other Peripheral modules	Stop (Undefined)
I/O ports	RTCICn ports (n = 0 to 2): Selectable EXCIN: Selectable All ports not specified here: Undefined

Note: Stop (Retained) means that the contents of the internal registers are retained but the operations are suspended.
Note: Stop (Undefined) means that the contents of the internal registers are undefined and power to the internal circuit is cut off.

Figure 11.2 shows switching sequence of Battery backup function.



Note: For details, see section 60, Electrical Characteristics.
Note 1. $V_{DETBATT}$ indicates the threshold level of the power supply change between the VCC pin and the VBATT pin.

Figure 11.2 Switching sequence of battery backup function

11.3.2 VBATT Battery Power Supply Switch Usage

The battery power supply switch can switch the power supply from the VCC pin to the VBATT pin when the voltage being applied to the VCC pin drops. When the voltage rises, this switch changes the power supply from the VBATT pin to the VCC pin.

The VCC voltage drop detection function for the battery power supply switch in "Figure 11.1" consists of two voltage drop detectors. One is a VCC drop detector for normal operation of the battery power supply switch that operates according to the settings of the VDTETLVL[2:0] bit and the VDETE bit. The other is voltage monitoring 0 to reduce power consumption during the Deep Software Standby mode 1 or 2.

In cold start, VDETE bit is initialized by VBATT_POR reset and the VCC voltage drop detection function for the battery power supply switch is disabled.

To use the battery power supply switch, the proper $V_{DETBATT\_m}$ (m = 0 to 6) level must be selected by VDETLVL[2:0] bits and wait t_{DETWT} for stabilization. After wait, the VCC voltage drop detection function is enabled by VDETE bit. The battery power supply switch has a constraint that the $V_{DETBATT\_m}$ (m = 0 to 6) level selected by VDETLVL[2:0] bits should be below Voltage monitoring 0 level.

When the MCU transitions to the Deep Software Standby mode 1 or 2 and the low power consumption function of voltage monitor 0 is enabled, the VCC drop detector for normal operation of the battery power supply switch is stopped to reduce power consumption. At this time, the power supply switch is controlled by the voltage monitor 0. This control by voltage monitor 0 continues until it returns from Deep Software Standby mode.

The battery backup function should be used after the voltage monitoring 0 reset is enabled (OFS1(_SEC).PVDAS bit is 0).

When the low power consumption function of voltage monitor 0 is enabled by OFS1(_SEC).PVDLPSEL bit, the voltage monitor 1 and 2 must be disabled before enter the Deep Software Standby mode 1 or 2.

If you don't use battery power supply switch. You must set BPWSWSTP bit to 1 and short the VCC and VBATT pins. When BPWSWSTP bit is 1, switch is stop, and power is always supplied from VCC pin.

11.3.3 VBATT_R voltage drop detection function Usage

This function enables to detect VBATT_R voltage drop. When VBATT_R voltage is drop and lower than $V_{PDR\ (BATR)}$, VBATT_POR reset is asserted. The VBPORF flag is set to 1 by VBATT_POR reset. If VBPORF flag is 1, RTC counter and register is invalid and Battery Backup Function registers are reset. You must initialize the functions included backup power area.

Figure 11.3 shows VBATT_R voltage drop detection function.

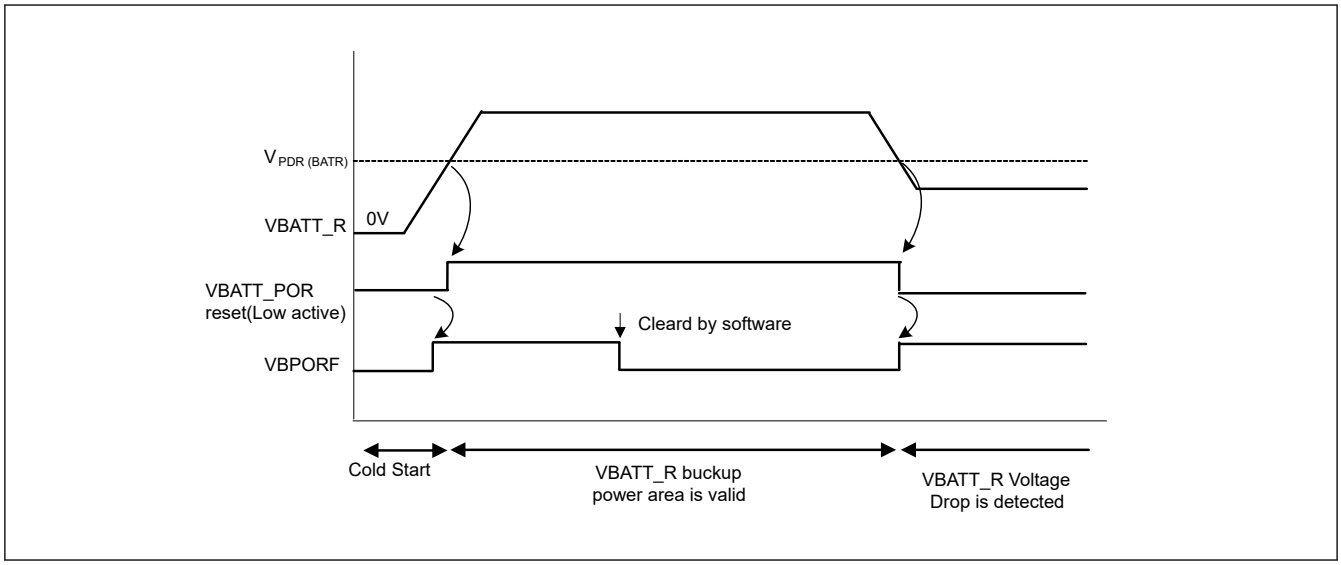


Figure 11.3 VBATT_R voltage drop detection function

11.3.4 VBATT Backup Register Usage

You can use 32, 16 or 8-bit access instruction when accessing the VBATT backup registers.

However, for example, when a 32 bit access instruction is executed, 8-bit read or write operation is executed with 4 consecutive times. When accessing, please note that the byte order of the data stored in the VBTBKR[n] (n = 0 to 127) register is little endian.

The data of VBATT backup register is cleared to 0x00 by VBATT_POR reset or tamper detection function.

The tamper detection function starts clearing operation of the VBATT backup register when the clear function is enabled and the tamper detection flag is set to 1. Do not cancel the clearing operation for 100 ns after starting. To cancel the clearing operation, disable the clear function or clear the tamper detection flag to 0. After canceling the clearing operation, do not access the VBATT backup register for 500 ns.

11.3.5 Tamper Detection Usage

The tamper detection function detects the RTCICn (n = 0 to 2) pin input event. The input event is defined as a change of RTCICn pin input level. The VBTADFn (n = 0 to 2) flag is set to 1 by the input event . When generate interrupt is enabled and the flag is set to 1, tamper detection interrupt is generated. When backup register clear function is enabled and flag is set to 1, the data of VBATT backup register is cleared. RTC time capture event source can select RTCICn pin input or the VBTADFn flag. If you need to synchronize time capture with a set of VBTADFn flags to 1, select the VBTADFn flag as the RTC time capture event source is recommended.

Figure 11.4 shows Tamper detection function.

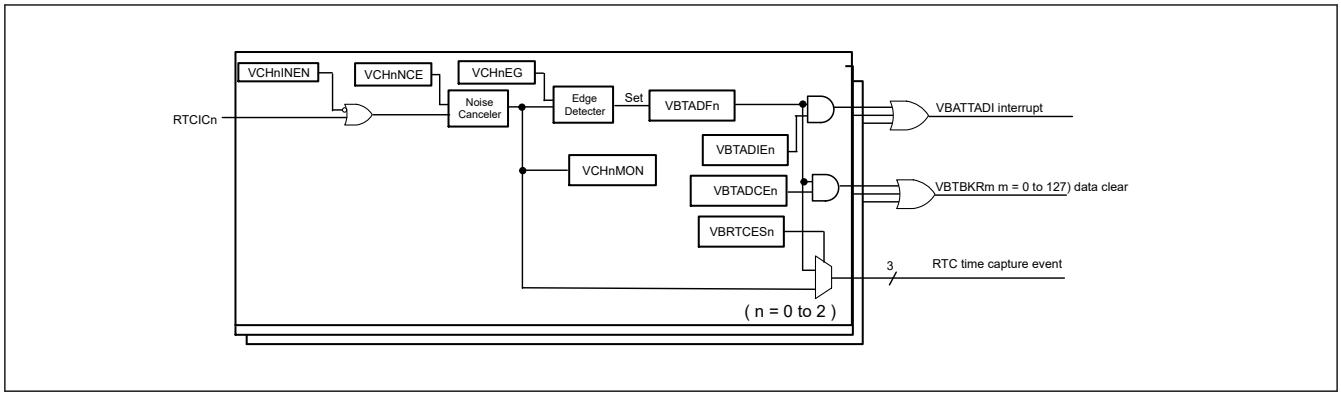


Figure 11.4 Tamper detection function

RTCICn pin inputs is enabled by the VCHnINEN (n = 0 to 2) bit. A wait time of 50us is required for the operation to stabilize after the RTCICn input is enabled.

The VCHnNCE (n = 0 to 2) bit enables noise canceler. The noise canceler samples input signals at the Sub-Clock (RTCSCLK) and removes the pulses whose length is less than three sampling cycles. Figure 11.5 shows noise canceler circuit block diagram.

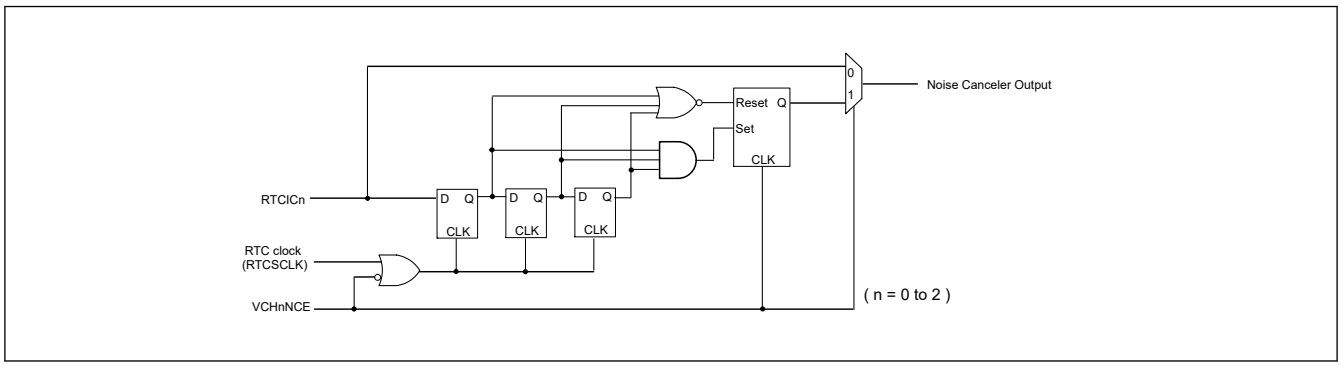


Figure 11.5 Tamper detection function

After enabling this noise canceler, a stabilization time of 5 clocks is required.

The VCHnEG (n = 0 to 2) bit enables to select Tamper detection edge. The edge detector operate asynchronously. When these control registers are changed, The VBTADFn flag may be set to 1 in a pseudo. The VBTADFn flag need check and clear to 0 after initialization of control registers . The VCHNnMON (n = 0 to 2) bit enables to monitor current input status. This bit also need check for inactive level after initialization of control registers .

The VBATTADI interrupt is asserted when 1 or more of 3 channels are flagged and interrupt is enabled.

The VBTBKRm(m = 0 to 127) register data is cleared to 0x00 when 1 or more of 3 channels are flagged and the VBATT backup register clear function is enabled.

11.3.6 VBATT voltage monitor function Usage

You can monitor the input voltage level of VBATT pin. 1/3 of VBATT pin voltage level can be monitored as analog signal. After setting the VBTMNSEL bit to 1, it is necessary to wait t_{MONWT} for the monitor level to stabilize. For details on t_{MONWT}, see section 60, Electrical Characteristics.

Figure 11.6 shows VBATT voltage monitor function.

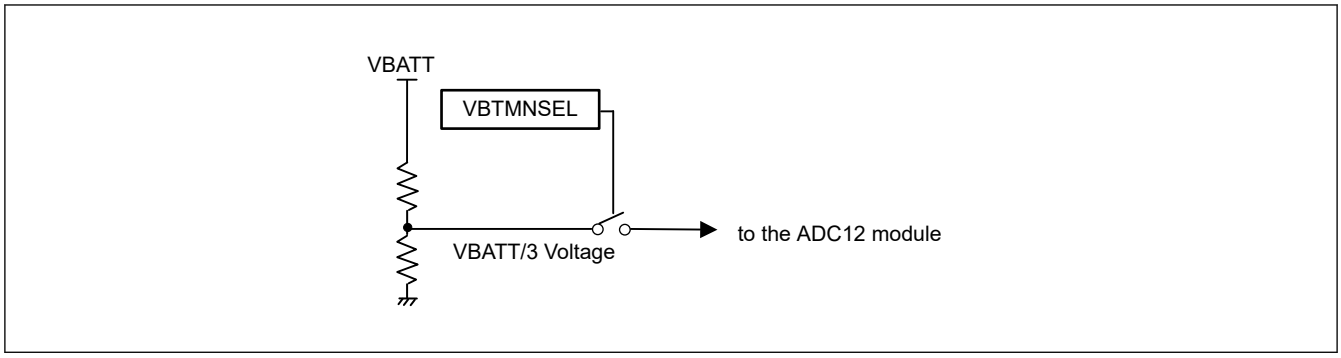


Figure 11.6 VBATT voltage monitor function

Note: For more information, see section 45, 12-Bit A/D Converter (ADC12).

Note: When VBTMNSEL bit is 1, VBAT power consumption is increased. It is recommended that VBTMNSEL bit sets to 1 only monitor timing.

11.3.7 Initial Settings Examples

11.3.7.1 Cold start and using the power supply switch flow example

When turning on VCC pin and VBATT pin power for the first time, use the following flow to initialize.

1. Check VBPORM bit. If VBPORM flag is 0, wait until it changes to 1.
2. Clear VBPORF flag to 0.
3. Set the VDETLVL[2:0] bit to appropriate value. The $V_{DETBATT\_m}$ level is Selected.
4. Wait t_{DETWT} for the VCC voltage detection function stabilization.
5. Set the VDETE bit to 1. The VCC voltage detection function is enabled.
6. Enable Sub-Clock Oscillator if needed.
7. Set Other Battery Backup Function and RTC registers.

For details on t_{DETWT}, see section 60, Electrical Characteristics.

For details on Sub-Clock Oscillator, see section 8, Clock Generation Circuit .

11.3.7.2 Warm start and using the power supply switch flow example

The MCU transitions to VBATT mode when the power of the VCC pin is turned off while the power of the VBATT pin is on. In this section, warm start means when the power supply of the VCC pin is turned on in VBATT mode. When the MCU starts processing with a warm start, execute the following flow in order to check the VBATT_R voltage drop before executing other process.

1. Check VBPORM flag. If VBPORM flag is 0, wait until it changes to 1.
2. Check VBPORF flag. If VBPORF flag is 1, the VBATT_R voltage drop is detected. Branch "section 11.3.7.1. Cold start and using the power supply switch flow example 2 to re-initialize the backup power area. If not, it means that the VBATT_R voltage does not drop and the state of the backup power area is retained. So, you do not need to re-initialize the backup power area.

11.3.7.3 Not using the power supply switch flow example

When not using the power supply switch, VCC pin and VBATT pin should be shorted. In this case, the VBATT_POR reset cannot follow the power on reset of VCC and VBATT_R voltage drop may not be detected. So, initialize the backup power supply area with the following flow.

1. Set the BPWSWSTP bit to 1. The power supply switch is stopped.
2. Check VBPORM flag. If VBPORM flag is 0, wait until it changes to 1.
3. Clear the VDETE bit to 0. The VCC voltage drop detection function is stopped.

4. Clear the VDETLVL[2:0] bit to 110b. The initial value is selected.
5. Check VBPORF flag. When VBPORF flag is 1, clear it to 0.
6. Set the SOSTP bit to 1 regardless of its value. Stop Sub-Clock Oscillator.
7. Initialize the VBTICTLR register and SOMCR.SOSEL bit. Initialization is recommended because these registers are related to the control of the IO port.
8. Initialize Other Battery Backup Function registers if need. Other Battery Backup Function registers are VBTICTLR2, VBTADSR, VBTADCR1, VBTADCR2, and VBTBKR[n].
9. Enable Sub-Clock Oscillator if needed.
10. Set RTC registers.

For details on Sub-Clock Oscillator, see section 8, Clock Generation Circuit.

11.3.7.4 Tamper detection function initialization setting flow example

If using the tamper detection function, it is recommended to initialize according to the following flow example.

1. Set the VCHnINEN to appropriate value.
2. Wait 50us to stabilize after the RTCICn input.
3. Set the VCHnNCE bit and the VCHnEG bit.
4. Wait 5 RTC-clocks to stabilize , if noise canceler is enabled.
5. Checking the VCHnMON bit indicates inactive is recommended. If it is active, it may not be possible to detect the tamper.
6. Initialize the invalid status of VBTADFn flag by clear to 0 after reading dummy read.
7. Set VBTADCR1, VBTADCR2 to enable interrupt, backup register clear, and RTC time capture event.
8. Enables RTC time capture function if needed.

11.4 Interrupt Sources

The Battery Backup Function has two interrupt sources and are listed in Table 11.3.

Table 11.3 Battery Backup Function interrupt Sources

Symbol	Interrupt source	Interrupt flag	Interrupt conditions
VBATTADI	VBATT Tamper detection	VBTADF0	VBTADF0 = 1, VBTADIE0 = 1
		VBTADF1	VBTADF1 = 1, VBTADIE1 = 1
		VBTADF2	VBTADF2 = 1, VBTADIE2 = 1

11.5 Usage Notes

1. Operation of the sub-clock oscillator and RTC are not guaranteed when the voltage level on VBATT is lower than the guaranteed operation range. When the power supply is restarted after VBATT falls below the guaranteed operating range, the RTC must be initialized in software.
2. A reset generated while writing to registers described in this section might destroy the register value.
3. When VCC is higher than $V_{DETBATT\_m}$, the VCC pin and VBATT pin are separated. When VCC is lower than $V_{DETBATT}$ and the switch is connected to the VBATT pin, and if the voltage on VBATT drops lower than VCC, current might flow into the VBATT pin through the parasitic diode between the VCC and VBATT pins.
4. During RTC operation using the voltage from the VBATT pin and the I/O ports within the backup, the power supply area can only be used as time capture event input pins for the RTC.

12. Register Write Protection

12.1 Overview

The register write protection function protects important registers from being overwritten due to software errors. The registers to be protected are set with the Protect Register (PRCR_S and PRCR_NS).

The two protected registers work on one Secure set registers/bits and the other for Non-secure set registers/bits. They are collectively mentioned as PRCR.

Table 12.1 lists the association between the bits in the PRCR register and the registers to be protected.

The register information to which the PRCR bit is applied is provided in the description of each register.

Table 12.1 Association between the bits in the PRCR register and registers to be protected

PRCR bit	Register to be protected
PRC0	• Registers related to the clock generation circuit: SCKDIVCR, SCKDIVCR2, SCKSCR, PLLCCR, PLLCR, BCKCR, MOSCCR, HOCOCR, HOCOCR2, MOCOCR, FLLCR1, FLLCR2, CKOCR, OSTDCR, OSTDSR, PLL2CCR, PLL2CR, PLLCCR2, PLL2CCR2, EBCKOCR, SDCKOCR, SCICKDIVCR, SCICKCR, SPICKDIVCR, SPICKCR, LCDCKDIVCR, LCDCKCR, MOCOUTCR, HOCOUTCR, USBCKDIVCR, OCTACKDIVCR, CANFDCKDIVCR, USB60CKDIVCR, I3CCKDIVCR, USBCKCR, OCTACKCR, CANFDCKCR, USB60CKCR, I3CCKCR, MOSCSCR, HOCOSCR, MOSCWTCR, MOMCR, SOSCCR, SOMCR, LOCOCR, LOCOUTCR, SYRACCR
PRC1	• Registers related to the low power modes: SBYCR, OPCCR, PDCTRGD, PDRAMSCR0, PDRAMSCR1, SSCR1, LPSCR, DPSBYCR, DPSWCR, DPSIER0-3, DPSIFR0-3, DPSIEGR0-2, PLL1LDOCR, PLL2LDOCR, HOCOLDOCR, LVOCR • Register related to the battery backup function: VBTBER, VBTICTLR, VBTBKR[n] (n = 0 to 127), VBTBPCR1, VBTBPCR2, VBTBPSR, VBTADSR, VBTADCR1, VBTADCR2, VBTICTLR2
PRC3	• Registers related the PVD: PVD1CR1, PVD1SR, PVD2CR1, PVD2SR, PVD1CMPCR, PVD2CMPCR, PVD1CR0, PVD2CR0, PVD1FCR, PVD2FCR, VBATTMNSELR
PRC4	• Registers related to the Security and Privilege setting registers: ELCSARx (x = A, B)*1, ELCPARx (x = A, B), PSARx (x = B to E), MSSAR, PPARx (x = B to E), MSPAR, PmSAR (m = 0 to 9, A, B), CPUSAR, DEBUGSAR, ICUSARx (x = A, B, E to I), SRAMSAR, BUSSARx (x = A to C), BUSPARC, MMPUSARx (x = A, B), DTCSAR, DMACSAR, DMACCHSAR, DMACCHPAR, TEVTRCR, SRAMSABAR0-1, STBRAMSABAR, STBRAMPABAR_NS, STBRAMPABAR_S, FSAR, CGFSAR, RSTSAR, LPMSAR, PVDSAR, BBFSAR, DPFSAR, RSCSAR, PGCSAR, VBRSABAR, VBRPABARS, VBRPABARNS
PRC5*1	• Registers related to the reset control SYRSTMSK0, SYRSTMSK2

Note 1. Only PRCR_S is supported.

12.2 Register Descriptions

12.2.1 PRCR_S : Protect Register for Secure (PRCR_S)

Base address: SYSC = 0x4001_E000

Offset address: 0x3FA

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PRKEY[7:0]								—	—	PRC5	PRC4	PRC3	—	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRC0	Enable writing to the registers related to the clock generation circuit 0: Disable writes 1: Enable writes	R/W

Bit	Symbol	Function	R/W
1	PRC1	Enable writing to the registers related to the low power modes, and the battery backup function 0: Disable writes 1: Enable writes	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	PRC3	Enable writing to the registers related to the PVD 0: Disable writes 1: Enable writes	R/W
4	PRC4	Enables writing to the registers related to the security and privilege setting registers. 0: Disable writes 1: Enable writes	R/W
5	PRC5	Enables writing to the registers related to the reset control. 0: Disable writes 1: Enable writes	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	0xA5: Enables writing to the PRCR_S register. Other than the above: Disables writing to the PRCR_S register. 0x00 are readable if these bits are read.	W

Note: S-TYPE6, P-TYPE2

PRCR_S is used to protect registers that are always Secure or configured as Secure.

PRCn bits (Protect bit n) (n = 0, 1, 3, 4, 5)

The PRCn bits enable or disable writing to the protected registers listed in Table 12.1. Setting the PRCn bits to 1 enables writing, and to 0 disables writing.

The register controlled by PRC4 may not reflect the PRC4 change when PRCR_S and its controlled registers are continuously written access. Avoid continuous write access or read the PRCR_S after PRC4 change, and then write to the register controlled by PRC4.

PRKEY[7:0] (Register write protection key bit)

These bits control permission and prohibition of writing to the PRCR_S register. To write PRCn bits of PRCR_S register, write 0xA5 to the PRKEY[7:0]. In case of writing other than 0xA5 to PRKEY[7:0], PRCn bits do not be changed even if writing to the PRCR_S register.

12.2.2 PRCR_NS : Protect Register for Non-secure (PRCR_NS)

Base address: SYSC_NS = 0x5001_E000

Offset address: 0x3FE

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PRKEY[7:0]								—	—	—	PRC4	PRC3	—	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRC0	Enable writing to the registers related to the clock generation circuit 0: Disable writes 1: Enable writes	R/W
1	PRC1	Enable writing to the registers related to the low power modes, and the battery backup function 0: Disable writes 1: Enable writes	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
3	PRC3	Enable writing to the registers related to the PVD 0: Disable writes 1: Enable writes	R/W
4	PRC4	Enables writing to the registers related to the security and privilege setting registers. 0: Disable writes 1: Enable writes	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	0xA5: Enables writing to the PRCR_NS register. Other than the above: Disables writing to the PRCR_NS register. 0x00 are readable if these bits are read.	W

Note: S-TYPE7, P-TYPE2

PRCR_NS is used to protect registers that are configured as Non-secure.

PRCn bits (Protect bit n) (n = 0, 1, 3, 4)

The PRCn bits enable or disable writing to the protected registers listed in Table 12.1. Setting the PRCn bits to 1 enables writing, and to 0 disables writing.

The register controlled by PRC4 may not reflect the PRC4 change when PRCR_NS and its controlled registers are continuously written access. Avoid continuous write to that registers or read the PRCR_NS after PRC4 change, and then write to the register controlled by PRC4.

PRKEY (Register write protection key bit)

These bits control permission and prohibition of writing to the PRCR_NS register. To write PRCn bits of PRCR_NS register, write 0xA5 to the PRKEY[7:0]. In case of writing other than 0xA5 to PRKEY[7:0], PRCn bits do not be changed even if writing to the PRCR_NS register.

13. Interrupt Controller Unit (ICU)

13.1 Overview

The Interrupt Controller Unit (ICU) controls which event signals are linked to the Nested Vector Interrupt Controller (NVIC), the DMA Controller (DMAC), and the Data Transfer Controller (DTC) modules. The ICU also controls non-maskable interrupts.

Table 13.1 lists the ICU specifications, Figure 13.1 shows a block diagram, and Table 13.2 lists the I/O pins.

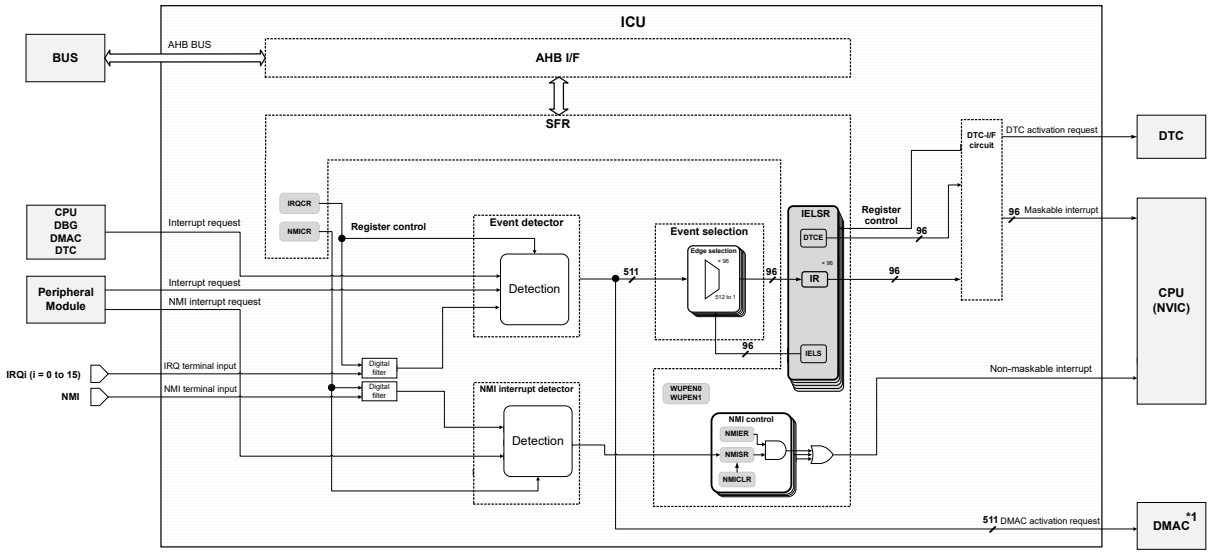
Table 13.1 ICU specifications

Parameter		Description
Maskable interrupts	Peripheral function interrupts	• Interrupts from peripheral modules Number of sources: 306 (select factor within event list numbers 17 to 511)
	External pin interrupts	• Interrupt detection on low level*4, falling edge, rising edge, rising and falling edges. One of these detection methods can be set for each source • Digital filter function supported • 16 sources, with interrupts from IRQi (i = 0 to 15) pins.
	Interrupt requests to CPU (NVIC)	• 96 interrupt requests are output to NVIC.*5
	DMAC control	• The DMAC can be activated using interrupt sources*1 • The target interrupt source can be selected individually for every DMAC channels.
	DTC control	• The DTC can be activated using interrupt sources*1 • The method for selecting an interrupt source is the same as that of the interrupt request to NVIC.
Non-maskable interrupts*2	NMI pin interrupt	• Interrupt from the NMI pin • Interrupt detection on falling edge or rising edge • Digital filter function supported
	Oscillation stop detection interrupt*3	Interrupt on detecting that the main oscillation has stopped
	WDT underflow/refresh error*3	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	IWDT underflow/refresh error*3	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	Voltage-monitoring 1 interrupt*3	Voltage monitor 1 interrupt of the voltage monitor 1 circuit (PVD_PVD1)
	Voltage-monitoring 2 interrupt*3	Voltage monitor 2 interrupt of the voltage monitor 2 circuit (PVD_PVD2)
	Common memory error interrupt	Common memory errors include SRAM ECC error, SRAM parity error, or Standby SRAM parity error
	Bus error Interrupt	Bus error includes MPU and TZF error
	CPU Lockup error interrupt	CPU Lockup error
Security	Secure	Some registers have Security Attribution
	Privilege	Each register of the ICU can only be accessed with Privilege access
Low power modes		• CPU Sleep mode: return is initiated by non-maskable interrupts or any other interrupt source • CPU Deep Sleep and Software Standby mode: Return is initiated by non-maskable interrupts. Interrupt can be selected as WUPENn register. See section 13.2.15. WUPEN0 : Wake Up Interrupt Enable Register 0, section 13.2.16. WUPEN1 : Wake Up interrupt enable register 1.
TrustZone Filter		Available

Note 1. For the DMAC and DTC activation sources, see section 13.3.2. Event Number.
Note 2. Non-maskable interrupts can be enabled only once after a reset release.
Note 3. These non-maskable interrupts can also be used as maskable interrupts. When used as maskable interrupts, do not change the value of the NMIER register from the reset state. To enable voltage monitor 1 and voltage monitor 2 interrupts, set the PVD1CR1.IRQSEL and PVD2CR1.IRQSEL bits to 1.

Note 4. Low level: interrupt detection is not canceled if you do not clear it after a detection.
Note 5. Appropriate alignment of the vector table according to the number of interrupts to be used is required. For example, in case of using maximum number of interrupts available in this products, at least 128-words alignment must be set.

Figure 13.1 shows the ICU block diagram.



Note 1. For more information on DELSR settings, see section 16, DMA Controller (DMAC)

Figure 13.1 ICU block diagram

Table 13.2 lists the ICU input/output pins.

Table 13.2 ICU I/O pins

Pin name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQi (i = 0 to 15)	Input	External interrupt request pins

13.2 Register Descriptions

This chapter does not describe the Arm® NVIC internal registers. For information about these registers, see Arm Limited., Arm® Cortex®-M85 Processor Technical Reference Manual (101924_0002_05_en).

13.2.1 ICUSARA : Interrupt Controller Unit Security Attribution Register A

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAIRQ CR15	SAIRQ CR14	SAIRQ CR13	SAIRQ CR12	SAIRQ CR11	SAIRQ CR10	SAIRQ CR9	SAIRQ CR8	SAIRQ CR7	SAIRQ CR6	SAIRQ CR5	SAIRQ CR4	SAIRQ CR3	SAIRQ CR2	SAIRQ CR1	SAIRQ CR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SAIRQCR15 to SAIRQCR0	Security attributes of registers for the IRQCRn register 0: Secure 1: Non-secure	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE1, P-TYPE1
Note: This register is write-protected by PRCR_S.PRC4 register

SAIRQCRn bits (Security attributes of registers for the IRQCRn register)

The target registers are as follows:

- IRQCR0 to IRQCR15 registers
- WUPEN0.IRQWUPEN[15:0] bits

13.2.2 ICUSARB : Interrupt Controller Unit Security Attribution Register B

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x44

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SANMI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SANMI	Security attributes of registers for nonmaskable interrupt 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE1, P-TYPE1
Note: This register is write-protected by PRCR_S.PRC4 register

SANMI bit (Security attributes of registers for nonmaskable interrupt)

Security attributes of registers for non-maskable interrupt. The target registers are as follows:

- NMISR
- NMIER
- NMICLR
- NMICR

The value of AIRCR.BFHFNMINS bit [13] in Application Interrupt and Reset Control Register of Arm CPU should be the same as the value of security attribution. The initial values of AIRCR.BFHFNMINS and the SANMI bits are different. AIRCR.BFHFNMINS is secure and SANMI is non-secure. Polarity has the same meaning so program these to match.

Note: Only one of Secure and Non-secure can set security attribution for non-maskable interrupt-related registers. If you program the Secure attribute as secure, it always goes to the Secure interrupt handler. To release any of the Non-maskable interrupt sources to the non-secure user, write a function to execute a nonsecure program from the interrupt handler for Secure.

13.2.3 ICUSARE : Interrupt Controller Unit Security Attribution Register E

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIIC0WUP	SAAGT1CBWUP	SAAGT1CAWUP	SAAGT1UDWUP	SAUSBFS0WUP	SAUSBHSWUP	SARTCPRDWUP	SARTCALMWUP	—	—	—	SAVBATTWUP	SAPVD2WUP	SAPVD1WUP	—	SAIWDTWUP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	SAIWDTWUP	Security attributes of registers for WUPEN0.b16 0: Secure 1: Non-secure	R/W
17	—	This bit is read as 0. The write value should be 0.	R/W
18	SAPVD1WUP	Security attributes of registers for WUPEN0.b18 0: Secure 1: Non-secure	R/W
19	SAPVD2WUP	Security attributes of registers for WUPEN0.b19 0: Secure 1: Non-secure	R/W
20	SAVBATTWUP	Security attributes of registers for WUPEN0.b20 0: Secure 1: Non-secure	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	SARTCALMWUP	Security attributes of registers for WUPEN0.b24 0: Secure 1: Non-secure	R/W
25	SARTCPRDWUP	Security attributes of registers for WUPEN0.b25 0: Secure 1: Non-secure	R/W
26	SAUSBHSWUP	Security attributes of registers for WUPEN0.b26 0: Secure 1: Non-secure	R/W
27	SAUSBFS0WUP	Security attributes of registers for WUPEN0.b27 0: Secure 1: Non-secure	R/W
28	SAAGT1UDWUP	Security attributes of registers for WUPEN0.b28 0: Secure 1: Non-secure	R/W
29	SAAGT1CAWUP	Security attributes of registers for WUPEN0.b29 0: Secure 1: Non-secure	R/W
30	SAAGT1CBWUP	Security attributes of registers for WUPEN0.b30 0: Secure 1: Non-secure	R/W
31	SAIIC0WUP	Security attributes of registers for WUPEN0.b31 0: Secure 1: Non-secure	R/W

Note: S-TYPE1, P-TYPE1
Note: This register is write-protected by PRCR_S.PRC4 register

13.2.4 ICUSARF : Interrupt Controller Unit Security Attribution Register F

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x54

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	SAULP1BWUP	SAULP1AWUP	SAULP1UWUP	SAI3CWUP	SAULP0BWUP	SAULP0AWUP	SAULP0UWUP	—	—	—	—	SACOMPHS0WUP	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	SACOMPHS0WUP	Security attributes of registers for WUPEN1.b3 0: Secure 1: Non-secure	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	SAULP0UWUP	Security attributes of registers for WUPEN1.b8 0: Secure 1: Non-secure	R/W
9	SAULP0AWUP	Security attributes of registers for WUPEN1.b9 0: Secure 1: Non-secure	R/W
10	SAULP0BWUP	Security attributes of registers for WUPEN1.b10 0: Secure 1: Non-secure	R/W
11	SAI3CWUP	Security attributes of registers for WUPEN1.b11 0: Secure 1: Non-secure	R/W
12	SAULP1UWUP	Security attributes of registers for WUPEN1.b12 0: Secure 1: Non-secure	R/W
13	SAULP1AWUP	Security attributes of registers for WUPEN1.b13 0: Secure 1: Non-secure	R/W
14	SAULP1BWUP	Security attributes of registers for WUPEN1.b14 0: Secure 1: Non-secure	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE1, P-TYPE1
Note: This register is write-protected by PRCR_S.PRC4 register

13.2.5 ICUSARG : Interrupt Controller Unit Security Attribution Register G

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x70

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIELSR31	SAIELSR30	SAIELSR29	SAIELSR28	SAIELSR27	SAIELSR26	SAIELSR25	SAIELSR24	SAIELSR23	SAIELSR22	SAIELSR21	SAIELSR20	SAIELSR19	SAIELSR18	SAIELSR17	SAIELSR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAIELSR15	SAIELSR14	SAIELSR13	SAIELSR12	SAIELSR11	SAIELSR10	SAIELSR9	SAIELSR8	SAIELSR7	SAIELSR6	SAIELSR5	SAIELSR4	SAIELSR3	SAIELSR2	SAIELSR1	SAIELSR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SAIELSR31 to SAIELSR0	Security attributes of registers for IELSR31 to IELSR0 0: Secure 1: Non-secure	R/W

Note: S-TYPE1, P-TYPE1
Note: This register is write-protected by PRCR_S.PRC4 register

SAIELSRn bits (Security attributes of registers for IELSR31 to IELSR0)

The Secure Attribute managed within the Arm CPU NVIC must match the security attribution of IELSEn (n = 0 to 31). NVIC internal registers are in NVIC_ITNS0[31:0]. Polarity has the same meaning so program these to match.

13.2.6 ICUSARH : Interrupt Controller Unit Security Attribution Register H

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x74

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIELSR63	SAIELSR62	SAIELSR61	SAIELSR60	SAIELSR59	SAIELSR58	SAIELSR57	SAIELSR56	SAIELSR55	SAIELSR54	SAIELSR53	SAIELSR52	SAIELSR51	SAIELSR50	SAIELSR49	SAIELSR48
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAIELSR47	SAIELSR46	SAIELSR45	SAIELSR44	SAIELSR43	SAIELSR42	SAIELSR41	SAIELSR40	SAIELSR39	SAIELSR38	SAIELSR37	SAIELSR36	SAIELSR35	SAIELSR34	SAIELSR33	SAIELSR32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SAIELSR63 to SAIELSR32	Security attributes of registers for IELSR63 to IELSR32 0: Secure 1: Non-secure	R/W

Note: S-TYPE1, P-TYPE1
Note: This register is write-protected by PRCR_S.PRC4 register

SAIELSRn bits (Security attributes of registers for IELSR63 to IELSR32)

The Secure Attribute managed within the Arm CPU NVIC must match the security attribution of IELSEn (n = 32 to 63). NVIC internal registers are in NVIC_ITNS1[31:0]. Polarity has the same meaning so program these to match.

13.2.7 ICUSARI : Interrupt Controller Unit Security Attribution Register I

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x78

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIELSR95	SAIELSR94	SAIELSR93	SAIELSR92	SAIELSR91	SAIELSR90	SAIELSR89	SAIELSR88	SAIELSR87	SAIELSR86	SAIELSR85	SAIELSR84	SAIELSR83	SAIELSR82	SAIELSR81	SAIELSR80
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAIELSR79	SAIELSR78	SAIELSR77	SAIELSR76	SAIELSR75	SAIELSR74	SAIELSR73	SAIELSR72	SAIELSR71	SAIELSR70	SAIELSR69	SAIELSR68	SAIELSR67	SAIELSR66	SAIELSR65	SAIELSR64
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SAIELSR95 to SAIELSR64	Security attributes of registers for IELSR95 to IELSR64 0: Secure 1: Non-secure	R/W

Note: S-TYPE1, P-TYPE1
Note: This register is write-protected by PRCR_S.PRC4 register

SAIELSRn bits (Security attributes of registers for IELSR95 to IELSR64)

The Secure Attribute managed within the Arm CPU NVIC must match the security attribution of IELSEn (n = 64 to 95). NVIC internal registers are in NVIC_ITNS2[31:0]. Polarity has the same meaning so program these to match.

13.2.8 TEVTRCR : Trusted Event Route Control Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x600

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TEVTE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TEVTE	Trusted Event Route Control Register for IELSRn, DELSRn and ELCSRn 0: Disable 1: Enable	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE1, P-TYPE1
Note: This register is write-protected by PRCR_S.PRC4

TEVTE bit (Trusted Event Route Control Register for IELSRn, DELSRn and ELCSRn)

When TEVTE = 1, the IELS [8:0] bits of IELSRn (n = 0 to 95), all DELS [8:0] bits of DMAC.DELSRn (n = 0 to 7), and all ELS [8:0] bits of ELC.ELSRn (n = 0 to18) are allowed secure access write. Non-secure access write is protected. Additionally, when TEVTE = 1, if the Security Attribution of the target register (IELS[31:16] bits of IELSRn (n = 0 to 95)) is non-secure, then secure access is not allowed. At this time, the upper level [31:16] cannot be read and write, but the response is OK, and no error occurs.

- IELSRn.IELS[8:0] (n = 0 to 95)
- DMAC.DELSRn.DELS[7:0] (n = 0 to 7)
- ELC.ELSRn.IELS[8:0] (n = 0 to 18)

13.2.9 IRQCRi : IRQ Control Register i (i = 0 to 15)

Base address: ICU_COMMON = 0x4000_6000
ICU_COMMON_NS = 0x5000_6000

Offset address: 0x000 + 0x1 × i

Bit position:	7	6	5	4	3	2	1	0
Bit field:	FLTEN	—	FCLKSEL[1:0]		—	—	IRQMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	IRQMD[1:0]	IRQi Detection Sense Select 0 0: Falling edge 0 1: Rising edge 1 0: Rising and falling edges 1 1: Low level	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
5:4	FCLKSEL[1:0]	IRQi Digital Filter Sampling Clock Select 0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	FLTEN	IRQi Digital Filter Enable 0: Digital filter is disabled 1: Digital filter is enabled.	R/W

Note: S-TYPE3, P-TYPE2

IRQCRi register changes must satisfy the following conditions:

- For a CPU interrupt or DTC trigger:
  Change the IRQCRi register value before setting the target IELSRn register (n = 0 to 95).
  The register value should be changed only when the value of the target IELSRn register is 0x0000.
- For a DMAC trigger:
  Change the IRQCRi register value before setting the target DMAC.DELSRn register (n = 0 to 7).
  The register value should be changed only when the value of the target DELSRn register is 0x0000.
- For a wakeup enable signal:
  Change the IRQCRi register setting before setting the target WUPEN0.IRQWUPEN[n] (n = 0 to 15). The register value should be changed when the target WUPEN0.IRQWUPEN[n] is 0.

IRQMD[1:0] bits (IRQi Detection Sense Select)

The IRQMD[1:0] bits set the detection sensing method for the IRQi external pin interrupt sources. For setting method when using external pin interrupt, see section 13.5.7. External Pin Interrupts.

FCLKSEL[1:0] bits (IRQi Digital Filter Sampling Clock Select)

The FCLKSEL[1:0] bits select the digital filter sampling clock for the IRQi external pin interrupt request pins, selectable to:

- PCLKB (every cycle)
- PCLKB/8 (once every 8 cycles)
- PCLKB/32 (once every 32 cycles)
- PCLKB/64 (once every 64 cycles)

For details of the digital filter, see section 13.5.6. Digital Filter.

FLTEN bit (IRQi Digital Filter Enable)

The FLTEN bit enables the digital filter used for the IRQi external pin interrupt sources. The digital filter is enabled when the IRQCRi.FLTEN bit is 1 and disabled when the IRQCRi.FLTEN bit is 0. The IRQi pin level is sampled at the clock cycle specified in the IRQCRi.FCLKSEL[1:0] bits. When the sampled level matches three times, the output level from the digital filter changes. For details of the digital filter, see section 13.5.6. Digital Filter.

13.2.10 NMISR : Non-Maskable Interrupt Status Register

Base address: ICU = 0x4000_C000
ICU_NS = 0x5000_C000

Offset address: 0x120

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	LUST	—	CMST	BUSST	—	—	—	—	NMIST	OSTST	—	—	PVD2ST	PVD1ST	WDTST	IWDTST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTST	IWDT Underflow/Refresh Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
1	WDTST	WDT Underflow/Refresh Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
2	PVD1ST	Voltage Monitor 1 Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
3	PVD2ST	Voltage Monitor 2 Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
5:4	—	These bits are read as 0.	R
6	OSTST	Main Clock Oscillation Stop Detection Interrupt Status Flag 0: Interrupt not requested for main clock oscillation stop 1: Interrupt requested for main clock oscillation stop	R
7	NMIST	NMI Pin Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
11:8	—	These bits are read as 0.	R
12	BUSST	Bus Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
13	CMST	Common Memory Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
14	—	This bit is read as 0.	R
15	LUST	CPU Lockup Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R

Note: S-TYPE3, P-TYPE2

The NMISR register monitors the status of non-maskable interrupt sources. Writes to the NMISR register are ignored. The setting in the Non-Maskable Interrupt Enable Register (NMIER) does not affect the status flags in this register. Before the end of the non-maskable interrupt handler, check that all of the bits in this register are set to 0 to confirm that no other NMI requests are generated during handler processing.

IWDTST flag (IWDT Underflow/Refresh Error Interrupt Status Flag)

The IWDTST flag indicates an IWDT underflow/refresh error interrupt request. It is read-only and cleared by the NMICLR.IWDTCLR bit.

[Setting condition]

When the IWDT underflow/refresh error interrupt is generated and this interrupt source is enabled.

[Clearing condition]

When 1 is written to the NMICLR.IWDTCLR bit.

WDTST flag (WDT Underflow/Refresh Error Interrupt Status Flag)

The WDTST flag indicates a WDT underflow/refresh error interrupt request. It is read-only and cleared by the NMICLR.WDTCLR bit.

[Setting condition]

When the WDT underflow/refresh error interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.WDTCLR bit.

PVD1ST flag (Voltage Monitor 1 Interrupt Status Flag)

The PVD1ST flag indicates a request for voltage monitor 1 interrupt. It is read-only and cleared by the NMICLR.PVD1CLR bit.

[Setting condition]

When the voltage monitor 1 interrupt is generated and this interrupt source is enabled.

[Clearing condition]

When 1 is written to the NMICLR.PVD1CLR bit.

PVD2ST flag (Voltage Monitor 2 Interrupt Status Flag)

The PVD2ST flag indicates a request for voltage monitor 2 interrupt. It is read-only and cleared by the NMICLR.PVD2CLR bit.

[Setting condition]

When the voltage monitor 2 interrupt is generated and this interrupt source is enabled.

[Clearing condition]

When 1 is written to the NMICLR.PVD2CLR bit.

OSTST flag (Main Clock Oscillation Stop Detection Interrupt Status Flag)

The OSTST flag indicates a main clock oscillation stop detection interrupt request. It is read-only and cleared by the NMICLR.OSTCLR bit.

[Setting condition]

When the main clock oscillation stop detection interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.OSTCLR bit.

NMIST flag (NMI Pin Interrupt Status Flag)

The NMIST flag indicates an NMI pin interrupt request. It is read-only and cleared by the NMICLR.NMICLR bit.

[Setting condition]

When an edge specified by the NMICR.NMIMD bit is input to the NMI pin.

[Clearing condition]

When 1 is written to the NMICLR.NMICLR bit.

BUSST flag (Bus Error Interrupt Status Flag)

The BUSST flag indicates a bus error interrupt request. It is read-only and cleared by the NMICLR.BUSCLR bit.

Bus error includes MPU and TZF errors.

[Setting condition]

When the bus error detection interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.SPECLR bit.

Be sure to clear the error status of the request source before clearing. If CPU does not clear the error status of the request source, the NMI status is set again even if this status is cleared, and an NMI request is issued to the CPU. If the CPU returns from the NMI handler, the CPU jumps back to the NMI handler.

In the case of level detection, use the following steps to clear the Status flag.

(1)Negate the level of an input factor.

(2)Perform a peripheral read access once and make sure the level interrupt is cleared.

(3)Clear the status flag by NMICLR.BUSCLR.

CMST flag (Common Memory Error Interrupt Status Flag)

The CMST flag indicates common memory error interrupt request. It is read-only and cleared by the NMICLR.CMCLR bit.

Common memory errors include SRAM ECC, SRAM Parity or StandbyRAM Parity.

[Setting condition]

When the common memory error detection interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.CMCLR bit.

Be sure to clear the error status of the request source before clearing. If CPU does not clear the error status of the request source, the NMI status is set again even if this status is cleared, and an NMI request is issued to the CPU. If the CPU returns from the NMI handler, the CPU jumps back to the NMI handler.

In the case of level detection, use the following steps to clear the Status flag.

(1)Negate the level of an input factor.

(2)Perform a peripheral read access once and make sure the level interrupt is cleared.

(3)Clear the status flag by NMICLR.CMCLR.

LUST flag (CPU Lockup Error Interrupt Status Flag)

The LUST flag indicates a CPU Lockup error interrupt request. It is read-only and cleared by the NMICLR. LUCLR bit.

[Setting condition]

When the CPU Lockup error interrupt is generated.

[Clearing condition]

Be sure to clear the error status of the request source before clearing. If CPU does not clear the error status of the request source, the NMI status is set again even if this status is cleared, and an NMI request is issued to the CPU. If the CPU returns from the NMI handler, the CPU jumps back to the NMI handler.

In the case of level detection, use the following steps to clear the Status flag.

(1)Negate the level of an input factor.

(2)Perform a peripheral read access once and make sure the level interrupt is cleared.

(3)Clear the status flag by NMICLR.LUSTCLR.

13.2.11 NMIER : Non-Maskable Interrupt Enable Register

Base address: ICU = 0x4000_C000
ICU_NS = 0x5000_C000

Offset address: 0x100

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	LUEN	—	CMEN	BUSEN	—	—	—	—	NMIEN	OSTEN	—	—	PVD2EN	PVD1EN	WDTEN	IWDTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTEN	IWDT Underflow/Refresh Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1 *2
1	WDTEN	WDT Underflow/Refresh Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1 *2
2	PVD1EN	Voltage monitor 1 Interrupt Enable 0: Disabled 1: Enabled	R/W*1 *2
3	PVD2EN	Voltage monitor 2 Interrupt Enable 0: Disabled 1: Enabled	R/W*1 *2
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	OSTEN	Main Clock Oscillation Stop Detection Interrupt Enable 0: Disabled 1: Enabled	R/W*1 *2
7	NMIEN	NMI Pin Interrupt Enable 0: Disabled 1: Enabled	R/W*1
11:8	—	These bits are read as 0. The write value should be 0.	R/W
12	BUSEN	Bus Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1
13	CMEN	Common Memory Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1
14	—	This bit is read as 0. The write value should be 0.	R/W
15	LUEN	CPU Lockup Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1

Note: S-TYPE3, P-TYPE2
Note 1. You can write 1 to this bit only once after reset. Subsequent write accesses are invalid. Writing 0 to this bit is invalid.
Note 2. Do not write 1 to this bit when the source is used as an interrupt signal.

IWDTEN bit (IWDT Underflow/Refresh Error Interrupt Enable)

The IWDTEN bit enables IWDT underflow/refresh error interrupt as an NMI trigger.

WDTEN bit (WDT Underflow/Refresh Error Interrupt Enable)

The WDTEN bit enables WDT underflow/refresh error interrupt as an NMI trigger.

PVD1EN bit (Voltage monitor 1 Interrupt Enable)

The PVD1EN bit enables voltage monitor 1 interrupt as an NMI trigger.

PVD2EN bit (Voltage monitor 2 Interrupt Enable)

The PVD2EN bit enables voltage monitor 2 interrupt as an NMI trigger.

OSTEN bit (Main Clock Oscillation Stop Detection Interrupt Enable)

The OSTEN bit enables main clock oscillation stop detection interrupt as an NMI trigger.

NMIEN bit (NMI Pin Interrupt Enable)

The NMIEN bit enables NMI pin interrupt as an NMI trigger.

BUSEN bit (Bus Error Interrupt Enable)

The BUSEN bit enables bus error interrupt as an NMI trigger. Bus error includes MPU and TZF errors.

CMEN bit (Common Memory Error Interrupt Enable)

The CMEN bit enables common memory error interrupt as an NMI trigger. Common memory errors include SRAM ECC, SRAM Parity or Standby RAM Parity.

LUEN bit (CPU Lockup Error Interrupt Enable)

LUEN bit enables CPU Lockup error interrupt as an NMI trigger.

13.2.12 NMICLR : Non-Maskable Interrupt Status Clear Register

Base address: ICU = 0x4000_C000
ICU_NS = 0x5000_C000

Offset address: 0x110

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	LUCLR	—	CMCLR	BUSCLR	—	—	—	—	NMICLR	OSTCLR	—	—	PVD2CLR	PVD1CLR	WDTCLR	IWDTCLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTCLR	IWDT Underflow/Refresh Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.IWDTST flag	R/W*1
1	WDTCLR	WDT Underflow/Refresh Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.WDTST flag	R/W*1
2	PVD1CLR	Voltage Monitor 1 Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.PVD1ST flag	R/W*1
3	PVD2CLR	Voltage Monitor 2 Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.PVD2ST flag.	R/W*1
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	OSTCLR	Oscillation Stop Detection Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.OSTST flag	R/W*1
7	NMICLR	NMI Pin Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.NMIST flag	R/W*1
11:8	—	These bits are read as 0. The write value should be 0.	R/W
12	BUSCLR	Bus Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.BUSST flag	R/W*1

Bit	Symbol	Function	R/W
13	CMCLR	Common Memory Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.CMST flag	R/W*1
14	—	This bit is read as 0. The write value should be 0.	R/W
15	LUCLR	CPU Lockup Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.LUST flag	R/W*1

Note: S-TYPE3, P-TYPE2
Note: here may be a difference in processing speed between the CPU and ICU, and the CPU may exit the interrupt handler before clearing the NMISR. Then the CPU will accidentally jump to the NMI handler again. To avoid this, be sure to read NMISR before exiting the NMI handler and make sure that NMISR is cleared before exiting the NMI handler.
Note 1. Only write 1 to this bit.

IWDTCLR bit (IWDT Underflow/Refresh Error Interrupt Status Flag Clear)

Writing 1 to the IWDTCLR bit clears the NMISR.IWDTST flag. This bit is read as 0.

WDTCLR bit (WDT Underflow/Refresh Error Interrupt Status Flag Clear)

Writing 1 to the WDTCLR bit clears the NMISR.WDTST flag. This bit is read as 0.

PVD1CLR bit (Voltage Monitor 1 Interrupt Status Flag Clear)

Writing 1 to the PVD1CLR bit clears the NMISR.PVD1ST flag. This bit is read as 0.

PVD2CLR bit (Voltage Monitor 2 Interrupt Status Flag Clear)

Writing 1 to the PVD2CLR bit clears the NMISR.PVD2ST flag. This bit is read as 0.

OSTCLR bit (Oscillation Stop Detection Interrupt Status Flag Clear)

Writing 1 to the OSTCLR bit clears the NMISR.OSTST flag. This bit is read as 0.

NMICLR bit (NMI Pin Interrupt Status Flag Clear)

Writing 1 to the NMICLR bit clears the NMISR.NMIST flag. This bit is read as 0.

BUSCLR bit (Bus Error Interrupt Status Flag Clear)

Writing 1 to the BUSCLR bit clears the NMISR.BUSST flag. This bit is read as 0.

CMCLR bit (Common Memory Error Interrupt Status Flag Clear)

Writing 1 to the CMCLR bit clears the NMISR.CMST flag. This bit is read as 0.

LUCLR bit (CPU Lockup Error Interrupt Status Flag Clear)

Writing 1 to the LUCLR bit clears the NMISR.LUST flag. This bit is read as 0.

13.2.13 NMICR : NMI Pin Interrupt Control Register

Base address: ICU_COMMON = 0x4000_6000
ICU_COMMON_NS = 0x5000_6000

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	NFLTEN	—	NFCLKSEL[1:0]		—	—	—	NMIMD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NMIMD	NMI Detection Set 0: Falling edge 1: Rising edge	R/W

Bit	Symbol	Function	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
5:4	NFCLKSEL[1:0]	NMI Digital Filter Sampling Clock Select 0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	NFLTEN	NMI Digital Filter Enable 0: Disabled 1: Enabled	R/W

Note: S-TYPE3, P-TYPE2

Change the NMICR register settings before enabling NMI pin interrupts, that is, before setting NMIER.NMIEN to 1.

NMIMD bit (NMI Detection Set)

The NMIMD bit selects the detection sensing method for the NMI pin interrupts.

NFCLKSEL[1:0] bits (NMI Digital Filter Sampling Clock Select)

The NFCLKSEL[1:0] bits select the digital filter sampling clock for the NMI pin interrupts, selectable to:

- PCLKB (every cycle)
- PCLKB/8 (once every 8 cycles)
- PCLKB/32 (once every 32 cycles)
- PCLKB/64 (once every 64 cycles)

For details of the digital filter, see section 13.5.6. Digital Filter.

NFLTEN bit (NMI Digital Filter Enable)

The NFLTEN bit enables the digital filter used for NMI pin interrupts. The filter is enabled when NFLTEN is 1, and disabled when NFLTEN is 0. The NMI pin level is sampled at the clock cycle specified in NFCLKSEL[1:0]. When the sampled level matches three times, the output level from the digital filter changes. For details of the digital filter, see section 13.5.6. Digital Filter.

13.2.14 IELSRn : ICU Event Link Setting Register n (n = 0 to 95)

Base address: ICU = 0x4000_C000
ICU_NS = 0x5000_C000

Offset address: 0x300 + 0x4 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	DTCE	—	—	—	—	—	—	—	IR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	IELS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	IELS[8:0]	ICU Event Link Select 0x00: Disable interrupts to the associated NVIC or DTC module Others: Event signal number to be linked. For details, see section 13.3.2. Event Number.	R/W*1
15:9	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	IR	Interrupt Status Flag 0: No interrupt request generated. 1: An interrupt request is generated.	R/W*2
23:17	—	These bits are read as 0. The write value should be 0.	R/W
24	DTCE	DTC Activation Enable 0: DTC activation is disabled. 1: DTC activation is enabled.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note 1. The [15:0] bits in this register can only be accessed in halfword or word access. Byte access is ignored.
Note 2. Writing 1 to the IR flag is prohibited.
This register has different secure access permission depending on the setting of Trusted Event Route Control Register (TEVTRCR).
If the security attribution is configured as secure,

- Secure access is allowed.
- Non-secure write access is ignored and non-secure read access is read as 0. TrustZone access error is generated.

If the security attribution is configured as non-secure and the trusted event route is disabled:

- Secure write access is ignored and secure read access is read as 0. TrustZone access error is generated.
- Non-secure access is allowed.

If the security attribution is configured as non-secure and the trusted event route is enabled:

- Secure access to IELS bit is allowed.
- Non-secure write access is ignored and non-secure read access is allowed.
- Secure write access to other bits is ignored and secure read access to other bits is read as 0.
- Non-secure access to other bits is allowed.
- TrustZone access error is not generated.

P-TYPE-2

The IELSRn register selects the interrupt source used by the NVIC. For details, see Table 13.3. IELSRn corresponds to the NVIC interrupt input source number, where n = 0 to 95.

IELS[8:0] bits (ICU Event Link Select)

The IELS[8:0] bits link an event signal to the associated NVIC or DTC module. Event options are classified into 8 groups (groups 0 to 7). For details, see section 13.3. Vector Table. All IELS[8:0] bits must be written at the same time.

IR flag (Interrupt Status Flag)

The IR status flag indicates an individual interrupt request from the event specified in IELS[8:0].

[Setting condition]

When an interrupt request is received from the associated peripheral module or IRQi pin.

[Clearing condition]

- The IR flag is cleared to 0 by writing 0.
- In the case of DTC.DISEL = 0. At the time other than the final transfer end in DTC transfer during DTCE = 1, IR flag repeat set and cleared by Hardware.
- In the case of DTC.DISEL = 1. For DTC transfers during DTCE = 1, the hardware does not clear the IR flag. Should be cleared by the CPU writing 0.

When DTC transfer except last transfer is completed (DTCE bit is changed from 1 to 0).

During DTCE = 1, write 0 to IR register is prohibited.

In the case of level detection, clear of the IR flag should follow the steps below.

1. Negate the input interrupt signal.
2. Run the peripheral read access once and wait for 2 clock cycles of the target module clock.
3. Clear the IR flag by writing 0.

Note: There may be a difference in processing speed between the CPU and ICU, and the CPU may exit the interrupt handler before clearing the IR. Then the CPU will accidentally jump to the interrupt handler again. To avoid this, be sure to read this register before exiting the interrupt handler and make sure that IR is cleared before exiting the interrupt handler.

DTCE bit (DTC Activation Enable)

When the DTCE bit is set to 1, the associated event is selected as the source for DTC activation.

[Setting condition]

- When 1 is written to the DTCE bit.

[Clearing condition]

- When the specified number of transfers is complete. For chain transfers, when the specified number of transfers for the last chain transfer is complete.
- When 0 is written to the DTCE bit.

Note: The secure attribution managed within the Arm CPU NVIC must match the SA (Security Attribution) of IELSEn (n = 0 to 95).

Note: Error during DTC transfer

If an error response occurs during DTC transfer, the DTC notifies the ICU that an error has occurred. ICU clears all bits of the target IELSRn (n = 0 to 95). IELSRn that is not the target is not cleared.

13.2.15 WUPEN0 : Wake Up Interrupt Enable Register 0

Base address: ICU = 0x4000_C000
ICU_NS = 0x5000_C000

Offset address: 0x1A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	IIC0WUPEN	AGT1CBWUPEN	AGT1CAWUPEN	AGT1UDWUPEN	USBFS0WUPEN	USBHSWUPEN	RTCPRDWUPEN	RTCALMWUPEN	—	—	—	VBATTWUPEN	PVD2WUPEN	PVD1WUPEN	—	IWDTWUPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	IRQWUPEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	IRQWUPEN[15:0]	IRQn Interrupt Deep Sleep/Software Standby Mode Returns Enable(n = 0 to 15) 0: Deep Sleep/Software Standby Mode returns by IRQn interrupt is disabled . 1: Deep Sleep/Software Standby Mode returns by IRQn interrupt is enabled*1	R/W
16	IWDTWUPEN	IWDT Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by IWDT interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by IWDT interrupt is enabled	R/W
17	—	This bit is read as 0. The write value should be 0.	R/W
18	PVD1WUPEN	PVD1 Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by PVD1 interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by PVD1 interrupt is enabled	R/W
19	PVD2WUPEN	PVD2 Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by PVD2 interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by PVD2 interrupt is enabled	R/W

Bit	Symbol	Function	R/W
20	VBATTWUPEN	VBATT Monitor Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by VBATT monitor interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by VBATT monitor interrupt is enabled	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	RTCALMWUPEN	RTC Alarm Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by RTC alarm interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by RTC alarm interrupt is enabled	R/W
25	RTCPRDWUPEN	RTC Period Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by RTC period interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by RTC period interrupt is enabled	R/W
26	USBHSWUPEN	USBHS Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by USBHS interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by USBHS interrupt is enabled	R/W
27	USBFS0WUPEN	USBFS Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by USBFS interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by USBFS interrupt is enabled	R/W
28	AGT1UDWUPEN	AGT1 Underflow Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by AGT1 underflow interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by AGT1 underflow interrupt is enabled	R/W
29	AGT1CAWUPEN	AGT1 Compare Match A Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by AGT1 compare match A interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by AGT1 compare match A interrupt is enabled	R/W
30	AGT1CBWUPEN	AGT1 Compare Match B Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by AGT1 compare match B interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by AGT1 compare match B interrupt is enabled	R/W
31	IIC0WUPEN	IIC0 Address Match Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by IIC0 address match A interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by IIC0 address match A interrupt is enabled	R/W

Note: S-TYPE4, P-TYPE2
Note 1. Description is a description of each bit.

IRQWUPEN[15:0] bits (IRQn Interrupt Deep Sleep/Software Standby Mode Returns Enable(n = 0 to 15))

The IRQWUPEN[15:0] are the enable bits to control the use of the IRQn pin as a Deep Sleep/Software Standby mode return factor. n= 0 to 15

IWDTWUPEN bit (IWDT Interrupt Deep Sleep/Software Standby Mode Returns Enable)

The IWDTWUPEN is the enable bit to control the use of the IWDT interrupt as a Deep Sleep/Software Standby mode return factor.

PVD1WUPEN bit (PVD1 Interrupt Deep Sleep/Software Standby Mode Returns Enable)

The PVD1WUPEN is the enable bit to control the use of the PVD1 interrupt as a Deep Sleep/Software Standby mode return factor.

PVD2WUPEN bit (PVD2 Interrupt Deep Sleep/Software Standby Mode Returns Enable)

The PVD2WUPEN is the enable bit to control the use of the PVD2 interrupt as a Deep Sleep/Software Standby mode return factor.

VBATTWUPEN bit (VBATT Monitor Interrupt Deep Sleep/Software Standby Mode Returns Enable)

The VBATTWUPEN is the enable bit to control the use of the VBATT monitor interrupt as a Deep Sleep/Software Standby mode return factor.

RTCALMWUPEN bit (RTC Alarm Interrupt Deep Sleep/Software Standby Mode Returns Enable)

The RTCALMWUPEN is the enable bit to control the use of the RTC alarm interrupt as a Deep Sleep/Software Standby mode return factor.

RTCPRDWUPEN bit (RTC Period Interrupt Deep Sleep/Software Standby Mode Returns Enable)

The RTCPRDWUPEN is the enable bit to control the use of the RTC period interrupt as a Deep Sleep/Software Standby mode return factor.

USBHSWUPEN bit (USBHS Interrupt Deep Sleep/Software Standby Mode Returns Enable)

The USBHSWUPEN is the enable bit to control the use of the USBHS interrupt as a Deep Sleep/Software Standby mode return factor.

USBFS0WUPEN bit (USBFS Interrupt Deep Sleep/Software Standby Mode Returns Enable)

The USBFS0WUPEN is the enable bit to control the use of the USBFS interrupt as a Deep Sleep/Software Standby mode return factor.

AGT1UDWUPEN bit (AGT1 Underflow Interrupt Deep Sleep/Software Standby Mode Returns Enable)

The AGT1UDWUPEN is the enable bit to control the use of the AGT1 underflow interrupt as a Deep Sleep/Software Standby mode return factor.

AGT1CAWUPEN bit (AGT1 Compare Match A Interrupt Deep Sleep/Software Standby Mode Returns Enable)

The AGT1CAWUPEN is the enable bit to control the use of the AGT1 compare match A interrupt as a Deep Sleep/Software Standby mode return factor.

AGT1CBWUPEN bit (AGT1 Compare Match B Interrupt Deep Sleep/Software Standby Mode Returns Enable)

The AGT1CBWUPEN is the enable bit to control the use of the AGT1 compare match B interrupt as a Deep Sleep/Software Standby mode return factor.

IIC0WUPEN bit (IIC0 Address Match Interrupt Deep Sleep/Software Standby Mode Returns Enable)

The IIC0WUPEN is the enable bit to control the use of the IIC0 interrupt as a Deep Sleep/Software Standby mode return factor.

Note: The security attribution of this register is set for each wakeup event.
To avoid the occurrence of a security hole, the target event of a wakeup and the security attribution added to this bit must match.

13.2.16 WUPEN1 : Wake Up interrupt enable register 1

Base address: ICU = 0x4000_C000
ICU_NS = 0x5000_C000

Offset address: 0x1A4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ULP1BWUPEN	ULP1AWUPEN	ULP1UWUPEN	I3CWUPEN	ULP0BWUPEN	ULP0AWUPEN	ULP0UWUPEN	—	—	—	—	COMPHS0WUPEN	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	COMPHS0WUPEN	Comparator-HS0 Interrupt Deep Sleep/Software Standby Mode returns Enable bit 0: Deep Sleep/Software Standby returns by Comparator-HS0 interrupt is disabled. 1: Deep Sleep/Software Standby returns by ULPT0 Underflow interrupt is enabled.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	ULP0UWUPEN	ULPT0 Underflow Interrupt Deep Sleep/Software Standby Mode returns Enable bit 0: Deep Sleep/Software Standby returns by ULPT0 interrupt is disabled. 1: Deep Sleep/Software Standby returns by ULPT0 Underflow interrupt is enabled.	R/W
9	ULP0AWUPEN	ULPT0 Compare Match A Interrupt Deep Sleep/Software Standby Mode returns Enable bit 0: Deep Sleep/Software Standby returns by ULPT0 Compare match A interrupt is disabled. 1: Deep Sleep/Software Standby returns by ULPT0 Compare match A interrupt is enabled.	R/W
10	ULP0BWUPEN	ULPT0 Compare Match B Interrupt Deep Sleep/Software Standby Mode returns Enable bit 0: Deep Sleep/Software Standby returns by ULPT0 Compare match B interrupt is disabled. 1: Deep Sleep/Software Standby returns by ULPT0 Compare match B interrupt is enabled.	R/W
11	I3CWUPEN	I3C Wakeup Condition Detection Interrupt Deep Sleep/Software Standby Mode returns Enable bit 0: Deep Sleep/Software Standby returns by I3C wake up interrupt is disabled. 1: Deep Sleep/Software Standby returns by I3C wake up interrupt is enabled.	R/W
12	ULP1UWUPEN	ULPT1 Underflow Interrupt Deep Sleep/Software Standby Mode returns Enable bit 0: Deep Sleep/Software Standby returns by ULPT1 Underflow interrupt is disabled. 1: Deep Sleep/Software Standby returns by ULPT1 Underflow interrupt is enabled	R/W
13	ULP1AWUPEN	ULPT1 Compare Match A Interrupt Deep Sleep/Software Standby Mode returns Enable bit 0: Deep Sleep/Software Standby returns by ULPT1 Compare match A interrupt is disabled. 1: Deep Sleep/Software Standby returns by ULPT1 Compare match A interrupt is enabled.	R/W
14	ULP1BWUPEN	ULPT1 Compare Match B Interrupt Deep Sleep/Software Standby Mode returns Enable bit 0: Deep Sleep/Software Standby returns by ULPT1 Compare match B interrupt is disabled. 1: Deep Sleep/Software Standby returns by ULPT1 Compare match B interrupt is enabled.	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE4, P-TYPE2

COMPHS0WUPEN bit (Comparator-HS0 Interrupt Deep Sleep/Software Standby Mode returns Enable bit)

This bit is the enable bit to control the use of the Comparator-HS0 interrupt as a Deep Sleep/Software Standby return factor.

ULP0UWUPEN bit (ULPT0 Underflow Interrupt Deep Sleep/Software Standby Mode returns Enable bit)

This bit is the enable bit to control the use of the ULPT0 Underflow interrupt as a Deep Sleep/Software Standby return factor.

ULP0AWUPEN bit (ULPT0 Compare Match A Interrupt Deep Sleep/Software Standby Mode returns Enable bit)

This bit is the enable bit to control the use of the ULPT0 Compare match A interrupt as a Deep Sleep/Software Standby return factor.

ULP0BWUPEN bit (ULPT0 Compare Match B Interrupt Deep Sleep/Software Standby Mode returns Enable bit)

This bit is the enable bit to control the use of the ULPT0 Compare match B interrupt as a Deep Sleep/Software Standby return factor

I3CWUPEN bit (I3C Wakeup Condition Detection Interrupt Deep Sleep/Software Standby Mode returns Enable bit)

This bit is the enable bit to control the use of the I3C wake up condition detection interrupt as a Deep Sleep/Software Standby return factor.

ULP1UWUPEN bit (ULPT1 Underflow Interrupt Deep Sleep/Software Standby Mode returns Enable bit)

This bit is the enable bit to control the use of the ULPT1 Underflow interrupt as a Deep Sleep/Software Standby return factor.

ULP1AWUPEN bit (ULPT1 Compare Match A Interrupt Deep Sleep/Software Standby Mode returns Enable bit)

This bit is the enable bit to control the use of the ULPT1 Compare match A interrupt as a Deep Sleep/Software Standby return factor.

ULP1BWUPEN bit (ULPT1 Compare Match B Interrupt Deep Sleep/Software Standby Mode returns Enable bit)

This bit is the enable bit to control the use of the ULPT1 Compare match B interrupt as a Deep Sleep/Software Standby return factor.

Note: The security attribution of this register is set for each wakeup event.

To avoid the occurrence of a security hole, the target event of a wakeup and the security attribution added to this bit must match.

13.3 Vector Table

The ICU detects maskable and non-maskable interrupts. Interrupt priorities are set up in the Arm NVIC. For information about these registers, see section 13.10. Reference.

13.3.1 Interrupt Vector Table

Table 13.3 describes the interrupt vector table. The interrupt vector addresses conform to the NVIC specifications.

Table 13.3 Interrupt vector table (1 of 4)

Exception number	IRQ number	Vector offset	Source	Description
0	—	0x000	Arm	Initial stack pointer
1	—	0x004	Arm	Initial program counter (reset vector)
2	—	0x008	Arm	Non-Maskable Interrupt (NMI)
3	—	0x00C	Arm	HardFault
4	—	0x010	Arm	MemManage Fault
5	—	0x014	Arm	BusFault

Table 13.3 Interrupt vector table (2 of 4)

Exception number	IRQ number	Vector offset	Source	Description
6	—	0x018	Arm	UsageFault
7	—	0x01C	Arm	SecureFault
8	—	0x020	Arm	Reserved
9	—	0x024	Arm	Reserved
10	—	0x028	Arm	Reserved
11	—	0x02C	Arm	Supervisor Call (SVCall)
12	—	0x030	Arm	DebugMonitor
13	—	0x034	Arm	Reserved
14	—	0x038	Arm	Pendable request for system service (PendableSrvReq)
15	—	0x03C	Arm	System Tick Timer (SysTick)
16	0	0x040	ICU.IELSR0	Event selected in the ICU.IELSR0 register
17	1	0x044	ICU.IELSR1	Event selected in the ICU.IELSR1 register
18	2	0x048	ICU.IELSR2	Event selected in the ICU.IELSR2 register
19	3	0x04C	ICU.IELSR3	Event selected in the ICU.IELSR3 register
20	4	0x050	ICU.IELSR4	Event selected in the ICU.IELSR4 register
21	5	0x054	ICU.IELSR5	Event selected in the ICU.IELSR5 register
22	6	0x058	ICU.IELSR6	Event selected in the ICU.IELSR6 register
23	7	0x05C	ICU.IELSR7	Event selected in the ICU.IELSR7 register
24	8	0x060	ICU.IELSR8	Event selected in the ICU.IELSR8 register
25	9	0x064	ICU.IELSR9	Event selected in the ICU.IELSR9 register
26	10	0x068	ICU.IELSR10	Event selected in the ICU.IELSR10 register
27	11	0x06C	ICU.IELSR11	Event selected in the ICU.IELSR11 register
28	12	0x070	ICU.IELSR12	Event selected in the ICU.IELSR12 register
29	13	0x074	ICU.IELSR13	Event selected in the ICU.IELSR13 register
30	14	0x078	ICU.IELSR14	Event selected in the ICU.IELSR14 register
31	15	0x07C	ICU.IELSR15	Event selected in the ICU.IELSR15 register
32	16	0x080	ICU.IELSR16	Event selected in the ICU.IELSR16 register
33	17	0x084	ICU.IELSR17	Event selected in the ICU.IELSR17 register
34	18	0x088	ICU.IELSR18	Event selected in the ICU.IELSR18 register
35	19	0x08C	ICU.IELSR19	Event selected in the ICU.IELSR19 register
36	20	0x090	ICU.IELSR20	Event selected in the ICU.IELSR20 register
37	21	0x094	ICU.IELSR21	Event selected in the ICU.IELSR21 register
38	22	0x098	ICU.IELSR22	Event selected in the ICU.IELSR22 register
39	23	0x09C	ICU.IELSR23	Event selected in the ICU.IELSR23 register
40	24	0x0A0	ICU.IELSR24	Event selected in the ICU.IELSR24 register
41	25	0x0A4	ICU.IELSR25	Event selected in the ICU.IELSR25 register
42	26	0x0A8	ICU.IELSR26	Event selected in the ICU.IELSR26 register
43	27	0x0AC	ICU.IELSR27	Event selected in the ICU.IELSR27 register
44	28	0x0B0	ICU.IELSR28	Event selected in the ICU.IELSR28 register
45	29	0x0B4	ICU.IELSR29	Event selected in the ICU.IELSR29 register
46	30	0x0B8	ICU.IELSR30	Event selected in the ICU.IELSR30 register

Table 13.3 Interrupt vector table (3 of 4)

Exception number	IRQ number	Vector offset	Source	Description
47	31	0x0BC	ICU.IELSR31	Event selected in the ICU.IELSR31 register
48	32	0x0C0	ICU.IELSR32	Event selected in the ICU.IELSR32 register
49	33	0x0C4	ICU.IELSR33	Event selected in the ICU.IELSR33 register
50	34	0x0C8	ICU.IELSR34	Event selected in the ICU.IELSR34 register
51	35	0x0CC	ICU.IELSR35	Event selected in the ICU.IELSR35 register
52	36	0x0D0	ICU.IELSR36	Event selected in the ICU.IELSR36 register
53	37	0x0D4	ICU.IELSR37	Event selected in the ICU.IELSR37 register
54	38	0x0D8	ICU.IELSR38	Event selected in the ICU.IELSR38 register
55	39	0x0DC	ICU.IELSR39	Event selected in the ICU.IELSR39 register
56	40	0x0E0	ICU.IELSR40	Event selected in the ICU.IELSR40 register
57	41	0x0E4	ICU.IELSR41	Event selected in the ICU.IELSR41 register
58	42	0x0E8	ICU.IELSR42	Event selected in the ICU.IELSR42 register
59	43	0x0EC	ICU.IELSR43	Event selected in the ICU.IELSR43 register
60	44	0x0F0	ICU.IELSR44	Event selected in the ICU.IELSR44 register
61	45	0x0F4	ICU.IELSR45	Event selected in the ICU.IELSR45 register
62	46	0x0F8	ICU.IELSR46	Event selected in the ICU.IELSR46 register
63	47	0x0FC	ICU.IELSR47	Event selected in the ICU.IELSR47 register
64	48	0x100	ICU.IELSR48	Event selected in the ICU.IELSR48 register
65	49	0x104	ICU.IELSR49	Event selected in the ICU.IELSR49 register
66	50	0x108	ICU.IELSR50	Event selected in the ICU.IELSR50 register
67	51	0x10C	ICU.IELSR51	Event selected in the ICU.IELSR51 register
68	52	0x110	ICU.IELSR52	Event selected in the ICU.IELSR52 register
69	53	0x114	ICU.IELSR53	Event selected in the ICU.IELSR53 register
70	54	0x118	ICU.IELSR54	Event selected in the ICU.IELSR54 register
71	55	0x11C	ICU.IELSR55	Event selected in the ICU.IELSR55 register
72	56	0x120	ICU.IELSR56	Event selected in the ICU.IELSR56 register
73	57	0x124	ICU.IELSR57	Event selected in the ICU.IELSR57 register
74	58	0x128	ICU.IELSR58	Event selected in the ICU.IELSR58 register
75	59	0x12C	ICU.IELSR59	Event selected in the ICU.IELSR59 register
76	60	0x130	ICU.IELSR60	Event selected in the ICU.IELSR60 register
77	61	0x134	ICU.IELSR61	Event selected in the ICU.IELSR61 register
78	62	0x138	ICU.IELSR62	Event selected in the ICU.IELSR62 register
79	63	0x13C	ICU.IELSR63	Event selected in the ICU.IELSR63 register
80	64	0x140	ICU.IELSR64	Event selected in the ICU.IELSR64 register
81	65	0x144	ICU.IELSR65	Event selected in the ICU.IELSR65 register
82	66	0x148	ICU.IELSR66	Event selected in the ICU.IELSR66 register
83	67	0x14C	ICU.IELSR67	Event selected in the ICU.IELSR67 register
84	68	0x150	ICU.IELSR68	Event selected in the ICU.IELSR68 register
85	69	0x154	ICU.IELSR69	Event selected in the ICU.IELSR69 register
86	70	0x158	ICU.IELSR70	Event selected in the ICU.IELSR70 register
87	71	0x15C	ICU.IELSR71	Event selected in the ICU.IELSR71 register

Table 13.3 Interrupt vector table (4 of 4)

Exception number	IRQ number	Vector offset	Source	Description
88	72	0x160	ICU.IELSR72	Event selected in the ICU.IELSR72 register
89	73	0x164	ICU.IELSR73	Event selected in the ICU.IELSR73 register
90	74	0x168	ICU.IELSR74	Event selected in the ICU.IELSR74 register
91	75	0x16C	ICU.IELSR75	Event selected in the ICU.IELSR75 register
92	76	0x170	ICU.IELSR76	Event selected in the ICU.IELSR76 register
93	77	0x174	ICU.IELSR77	Event selected in the ICU.IELSR77 register
94	78	0x178	ICU.IELSR78	Event selected in the ICU.IELSR78 register
95	79	0x17C	ICU.IELSR79	Event selected in the ICU.IELSR79 register
96	80	0x180	ICU.IELSR80	Event selected in the ICU.IELSR80 register
97	81	0x184	ICU.IELSR81	Event selected in the ICU.IELSR81 register
98	82	0x188	ICU.IELSR82	Event selected in the ICU.IELSR82 register
99	83	0x18C	ICU.IELSR83	Event selected in the ICU.IELSR83 register
100	84	0x190	ICU.IELSR84	Event selected in the ICU.IELSR84 register
101	85	0x194	ICU.IELSR85	Event selected in the ICU.IELSR85 register
102	86	0x198	ICU.IELSR86	Event selected in the ICU.IELSR86 register
103	87	0x19C	ICU.IELSR87	Event selected in the ICU.IELSR87 register
104	88	0x1A0	ICU.IELSR88	Event selected in the ICU.IELSR88 register
105	89	0x1A4	ICU.IELSR89	Event selected in the ICU.IELSR89 register
106	90	0x1A8	ICU.IELSR90	Event selected in the ICU.IELSR90 register
107	91	0x1AC	ICU.IELSR91	Event selected in the ICU.IELSR91 register
108	92	0x1B0	ICU.IELSR92	Event selected in the ICU.IELSR92 register
109	93	0x1B4	ICU.IELSR93	Event selected in the ICU.IELSR93 register
110	94	0x1B8	ICU.IELSR94	Event selected in the ICU.IELSR94 register
111	95	0x1BC	ICU.IELSR95	Event selected in the ICU.IELSR95 register

13.3.2 Event Number

The following table lists heading details for Table 13.4, which describes each event number.

Heading	Description
Interrupt request source	Name of the source generating the interrupt request
Name	Name of the interrupt
Connect to NVIC	“ ✓” indicates the interrupt can be used as a CPU interrupt
Invoke DTC	“ ✓” indicates the interrupt can be used to request DTC activation
Invoke DMAC	“✓” indicates the interrupt can be used to request DMAC activation
Canceling CPU Deep Sleep	“✓” indicates the interrupt can be used to request a return from CPU Deep Sleep mode
Canceling Software Standby	“ ✓” indicates the interrupt can be used to request a return from Software Standby mode
Canceling Deep Software Standby	“✓” indicates the interrupt can be used to request a return from Deep Software Standby mode

Table 13.4 Event table (1 of 10)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x001	Port	PORT_IRQ0	✓	✓	✓	✓	✓	✓
0x002		PORT_IRQ1	✓	✓	✓	✓	✓	✓
0x003		PORT_IRQ2	✓	✓	✓	✓	✓	✓
0x004		PORT_IRQ3	✓	✓	✓	✓	✓	✓
0x005		PORT_IRQ4	✓	✓	✓	✓	✓	✓
0x006		PORT_IRQ5	✓	✓	✓	✓	✓	✓
0x007		PORT_IRQ6	✓	✓	✓	✓	✓	✓
0x008		PORT_IRQ7	✓	✓	✓	✓	✓	✓
0x009		PORT_IRQ8	✓	✓	✓	✓	✓	✓
0x00A		PORT_IRQ9	✓	✓	✓	✓	✓	✓
0x00B		PORT_IRQ10	✓	✓	✓	✓	✓	✓
0x00C		PORT_IRQ11	✓	✓	✓	✓	✓	✓
0x00D		PORT_IRQ12	✓	✓	✓	✓	✓	✓
0x00E		PORT_IRQ13	✓	✓	✓	✓	✓	✓
0x00F		PORT_IRQ14	✓	✓	✓	✓	✓	✓
0x010		PORT_IRQ15	✓	✓	✓	✓	✓	✓
0x011	DMAC00	DMAC00_INT	✓	✓	—	—	—	—
0x012	DMAC01	DMAC01_INT	✓	✓	—	—	—	—
0x013	DMAC02	DMAC02_INT	✓	✓	—	—	—	—
0x014	DMAC03	DMAC03_INT	✓	✓	—	—	—	—
0x015	DMAC04	DMAC04_INT	✓	✓	—	—	—	—
0x016	DMAC05	DMAC05_INT	✓	✓	—	—	—	—
0x017	DMAC06	DMAC06_INT	✓	✓	—	—	—	—
0x018	DMAC07	DMAC07_INT	✓	✓	—	—	—	—
0x022	DTC0	DTC0_COMPLETE	✓	—	—	—	—	—
0x027	DMAC/DTC	DMA0_TRANSERR	✓	—	—	—	—	—
0x029	DBG	DBG_CTIIRQ0	✓	—	—	—	—	—
0x02A		DBG_CTIIRQ1	✓	—	—	—	—	—
0x02B		DBG_JBRXI	✓	—	—	—	—	—
0x030	FCU	FCU_FIFERR	✓	—	—	—	—	—
0x031		FCU_FRDYI	✓	—	—	—	—	—
0x038	PVD1	PVD_PVD1	✓	—	—	✓	✓	✓*4
0x039	PVD2	PVD_PVD2	✓	—	—	✓	✓	✓*4
0x03D	BBF	VBATT_TADI	✓	—	—	✓	✓	✓
0x03E	MOSC	MOSC_STOP	✓	—	—	—	—	—
0x040	ULPT0	ULPT0_ULPTI	✓	✓	✓	✓	✓	✓*5
0x041		ULPT0_ULPTCMAI	✓	✓	✓	✓	✓	—
0x042		ULPT0_ULPTCMBI	✓	✓	✓	✓	✓	—

Table 13.4 Event table (2 of 10)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x043	ULPT1	ULPT1_ULPTI	✓	✓	✓	✓	✓	✓*5
0x044		ULPT1_ULPTCMAI	✓	✓	✓	✓	✓	—
0x045		ULPT1_ULPTCMBI	✓	✓	✓	✓	✓	—
0x046	AGT0	AGT0_AGTI	✓	✓	✓	—	—	—
0x047		AGT0_AGTCMAI	✓	✓	✓	—	—	—
0x048		AGT0_AGTCMBI	✓	✓	✓	—	—	—
0x049	AGT1	AGT1_AGTI	✓	✓	✓	✓	✓	—
0x04A		AGT1_AGTCMAI	✓	✓	✓	✓	✓	—
0x04B		AGT1_AGTCMBI	✓	✓	✓	✓	✓	—
0x052	IWDT	IWDT_NMIUNDF	✓	—	—	✓	✓	✓*5
0x053	WDT	WDT0_NMIUNDF	✓	—	—	—	—	—
0x055	RTC	RTC_ALM	✓	—	—	✓	✓	✓
0x056		RTC_PRD	✓	—	—	✓	✓	✓
0x057		RTC_CUP	✓	—	—	—	—	—
0x058	USBFS	USBFS_D0FIFO	✓	✓	✓	—	—	—
0x059		USBFS_D1FIFO	✓	✓	✓	—	—	—
0x05A		USBFS_USBI	✓	—	—	—	—	—
0x05B		USBFS_USBR	✓	—	—	✓	✓	✓*5
0x05C	IIC0	IIC0_RXI	✓	✓	✓	—	—	—
0x05D		IIC0_TXI	✓	✓	✓	—	—	—
0x05E		IIC0_TEI	✓	—	—	—	—	—
0x05F		IIC0_EEI	✓	—	—	—	—	—
0x060		IIC0_WUI	✓	—	—	✓	✓	—
0x061	IIC1	IIC1_RXI	✓	✓	✓	—	—	—
0x062		IIC1_TXI	✓	✓	✓	—	—	—
0x063		IIC1_TEI	✓	—	—	—	—	—
0x064		IIC1_EEI	✓	—	—	—	—	—
0x06B	SDHI/ MMC0	SDHI_MMC0_ACCS	✓	—	—	—	—	—
0x06C		SDHI_MMC0_SDIO	✓	—	—	—	—	—
0x06D		SDHI_MMC0_CARD	✓	—	—	—	—	—
0x06E		SDHI_MMC0_ODMSDBREQ	—	✓	✓	—	—	—
0x06F	SDHI/ MMC1	SDHI_MMC1_ACCS	✓	—	—	—	—	—
0x070		SDHI_MMC1_SDIO	✓	—	—	—	—	—
0x071		SDHI_MMC1_CARD	✓	—	—	—	—	—
0x072		SDHI_MMC1_ODMSDBREQ	—	✓	✓	—	—	—
0x073	SSI0	SSI0_SSITXI	✓	✓	✓	—	—	—
0x074		SSI0_SSIRXI	✓	✓	✓	—	—	—
0x076		SSI0_SSIF	✓	—	—	—	—	—
0x079	SSI1	SSI1_SSIRT	✓	✓	✓	—	—	—
0x07A		SSI1_SSIF	✓	—	—	—	—	—

Table 13.4 Event table (3 of 10)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x07B	ACMPHS	ACMP_HS0	✓	—	—	✓*1	✓*1	—
0x07C		ACMP_HS1	✓	—	—	—	—	—
0x083	ELC	ELC_SWEVT0	✓*3	✓	—	—	—	—
0x084		ELC_SWEVT1	✓*3	✓	—	—	—	—
0x088	PORT	IOPORT_GROUP1	✓	✓*2	✓*2	—	—	—
0x089		IOPORT_GROUP2	✓	✓*2	✓*2	—	—	—
0x08A		IOPORT_GROUP3	✓	✓*2	✓*2	—	—	—
0x08B		IOPORT_GROUP4	✓	✓*2	✓*2	—	—	—
0x08C	CAC	CAC_FEERI	✓	—	—	—	—	—
0x08D		CAC_MENDI	✓	—	—	—	—	—
0x08E		CAC_OVFI	✓	—	—	—	—	—
0x08F	POEG	POEG_GROUPA	✓	—	—	—	—	—
0x090		POEG_GROUPB	✓	—	—	—	—	—
0x091		POEG_GROUPC	✓	—	—	—	—	—
0x092		POEG_GROUPD	✓	—	—	—	—	—
0x0A0	GPT	GPT_UVWEDGE	✓	—	—	—	—	—
0x0A1	GPT0	GPT0_CCMPA	✓	✓	✓	—	—	—
0x0A2		GPT0_CCMPB	✓	✓	✓	—	—	—
0x0A3		GPT0_CMPC	✓	✓	✓	—	—	—
0x0A4		GPT0_CMPD	✓	✓	✓	—	—	—
0x0A5		GPT0_CMPE	✓	✓	✓	—	—	—
0x0A6		GPT0_CMPF	✓	✓	✓	—	—	—
0x0A7		GPT0_OVF	✓	✓	✓	—	—	—
0x0A8		GPT0_UDF	✓	✓	✓	—	—	—
0x0A9		GPT0_PC	✓	✓	✓	—	—	—
0x0AA	GPT1	GPT1_CCMPA	✓	✓	✓	—	—	—
0x0AB		GPT1_CCMPB	✓	✓	✓	—	—	—
0x0AC		GPT1_CMPC	✓	✓	✓	—	—	—
0x0AD		GPT1_CMPD	✓	✓	✓	—	—	—
0x0AE		GPT1_CMPE	✓	✓	✓	—	—	—
0x0AF		GPT1_CMPF	✓	✓	✓	—	—	—
0x0B0		GPT1_OVF	✓	✓	✓	—	—	—
0x0B1		GPT1_UDF	✓	✓	✓	—	—	—
0x0B2		GPT1_PC	✓	✓	✓	—	—	—

Table 13.4 Event table (4 of 10)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x0B3	GPT2	GPT2_CCMPA	✓	✓	✓	—	—	—
0x0B4		GPT2_CCMPB	✓	✓	✓	—	—	—
0x0B5		GPT2_CMPC	✓	✓	✓	—	—	—
0x0B6		GPT2_CMPD	✓	✓	✓	—	—	—
0x0B7		GPT2_CMPE	✓	✓	✓	—	—	—
0x0B8		GPT2_CMPF	✓	✓	✓	—	—	—
0x0B9		GPT2_OVF	✓	✓	✓	—	—	—
0x0BA		GPT2_UDF	✓	✓	✓	—	—	—
0x0BB		GPT2_PC	✓	✓	✓	—	—	—
0x0BC	GPT3	GPT3_CCMPA	✓	✓	✓	—	—	—
0x0BD		GPT3_CCMPB	✓	✓	✓	—	—	—
0x0BE		GPT3_CMPC	✓	✓	✓	—	—	—
0x0BF		GPT3_CMPD	✓	✓	✓	—	—	—
0x0C0		GPT3_CMPE	✓	✓	✓	—	—	—
0x0C1		GPT3_CMPF	✓	✓	✓	—	—	—
0x0C2		GPT3_OVF	✓	✓	✓	—	—	—
0x0C3		GPT3_UDF	✓	✓	✓	—	—	—
0x0C4		GPT3_PC	✓	✓	✓	—	—	—
0x0C5	GPT4	GPT4_CCMPA	✓	✓	✓	—	—	—
0x0C6		GPT4_CCMPB	✓	✓	✓	—	—	—
0x0C7		GPT4_CMPC	✓	✓	✓	—	—	—
0x0C8		GPT4_CMPD	✓	✓	✓	—	—	—
0x0C9		GPT4_CMPE	✓	✓	✓	—	—	—
0x0CA		GPT4_CMPF	✓	✓	✓	—	—	—
0x0CB		GPT4_OVF	✓	✓	✓	—	—	—
0x0CC		GPT4_UDF	✓	✓	✓	—	—	—
0x0CE	GPT5	GPT5_CCMPA	✓	✓	✓	—	—	—
0x0CF		GPT5_CCMPB	✓	✓	✓	—	—	—
0x0D0		GPT5_CMPC	✓	✓	✓	—	—	—
0x0D1		GPT5_CMPD	✓	✓	✓	—	—	—
0x0D2		GPT5_CMPE	✓	✓	✓	—	—	—
0x0D3		GPT5_CMPF	✓	✓	✓	—	—	—
0x0D4		GPT5_OVF	✓	✓	✓	—	—	—
0x0D5		GPT5_UDF	✓	✓	✓	—	—	—

Table 13.4 Event table (5 of 10)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x0D7	GPT6	GPT6_CCMPA	✓	✓	✓	—	—	—
0x0D8		GPT6_CCMPB	✓	✓	✓	—	—	—
0x0D9		GPT6_CMPC	✓	✓	✓	—	—	—
0x0DA		GPT6_CMPD	✓	✓	✓	—	—	—
0x0DB		GPT6_CMPE	✓	✓	✓	—	—	—
0x0DC		GPT6_CMPF	✓	✓	✓	—	—	—
0x0DD		GPT6_OVF	✓	✓	✓	—	—	—
0x0DE		GPT6_UDF	✓	✓	✓	—	—	—
0x0E0	GPT7	GPT7_CCMPA	✓	✓	✓	—	—	—
0x0E1		GPT7_CCMPB	✓	✓	✓	—	—	—
0x0E2		GPT7_CMPC	✓	✓	✓	—	—	—
0x0E3		GPT7_CMPD	✓	✓	✓	—	—	—
0x0E4		GPT7_CMPE	✓	✓	✓	—	—	—
0x0E5		GPT7_CMPF	✓	✓	✓	—	—	—
0x0E6		GPT7_OVF	✓	✓	✓	—	—	—
0x0E7		GPT7_UDF	✓	✓	✓	—	—	—
0x0E9	GPT8	GPT8_CCMPA	✓	✓	✓	—	—	—
0x0EA		GPT8_CCMPB	✓	✓	✓	—	—	—
0x0EB		GPT8_CMPC	✓	✓	✓	—	—	—
0x0EC		GPT8_CMPD	✓	✓	✓	—	—	—
0x0ED		GPT8_CMPE	✓	✓	✓	—	—	—
0x0EE		GPT8_CMPF	✓	✓	✓	—	—	—
0x0EF		GPT8_OVF	✓	✓	✓	—	—	—
0x0F0		GPT8_UDF	✓	✓	✓	—	—	—
0x0F1		GPT8_PC	✓	✓	✓	—	—	—
0x0F2	GPT9	GPT9_CCMPA	✓	✓	✓	—	—	—
0x0F3		GPT9_CCMPB	✓	✓	✓	—	—	—
0x0F4		GPT9_CMPC	✓	✓	✓	—	—	—
0x0F5		GPT9_CMPD	✓	✓	✓	—	—	—
0x0F6		GPT9_CMPE	✓	✓	✓	—	—	—
0x0F7		GPT9_CMPF	✓	✓	✓	—	—	—
0x0F8		GPT9_OVF	✓	✓	✓	—	—	—
0x0F9		GPT9_UDF	✓	✓	✓	—	—	—
0x0FA		GPT9_PC	✓	✓	✓	—	—	—

Table 13.4 Event table (6 of 10)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x0FB	GPT10	GPT10_CCMPA	✓	✓	✓	—	—	—
0x0FC		GPT10_CCMPB	✓	✓	✓	—	—	—
0x0FD		GPT10_CMPC	✓	✓	✓	—	—	—
0x0FE		GPT10_CMPD	✓	✓	✓	—	—	—
0x0FF		GPT10_CMPE	✓	✓	✓	—	—	—
0x100		GPT10_CMPF	✓	✓	✓	—	—	—
0x101		GPT10_OVF	✓	✓	✓	—	—	—
0x102		GPT10_UDF	✓	✓	✓	—	—	—
0x103		GPT10_PC	✓	✓	✓	—	—	—
0x104	GPT11	GPT11_CCMPA	✓	✓	✓	—	—	—
0x105		GPT11_CCMPB	✓	✓	✓	—	—	—
0x106		GPT11_CMPC	✓	✓	✓	—	—	—
0x107		GPT11_CMPD	✓	✓	✓	—	—	—
0x108		GPT11_CMPE	✓	✓	✓	—	—	—
0x109		GPT11_CMPF	✓	✓	✓	—	—	—
0x10A		GPT11_OVF	✓	✓	✓	—	—	—
0x10B		GPT11_UDF	✓	✓	✓	—	—	—
0x10D	GPT12	GPT12_CCMPA	✓	✓	✓	—	—	—
0x10E		GPT12_CCMPB	✓	✓	✓	—	—	—
0x10F		GPT12_CMPC	✓	✓	✓	—	—	—
0x110		GPT12_CMPD	✓	✓	✓	—	—	—
0x111		GPT12_CMPE	✓	✓	✓	—	—	—
0x112		GPT12_CMPF	✓	✓	✓	—	—	—
0x113		GPT12_OVF	✓	✓	✓	—	—	—
0x114		GPT12_UDF	✓	✓	✓	—	—	—
0x116	GPT13	GPT13_CCMPA	✓	✓	✓	—	—	—
0x117		GPT13_CCMPB	✓	✓	✓	—	—	—
0x118		GPT13_CMPC	✓	✓	✓	—	—	—
0x119		GPT13_CMPD	✓	✓	✓	—	—	—
0x11A		GPT13_CMPE	✓	✓	✓	—	—	—
0x11B		GPT13_CMPF	✓	✓	✓	—	—	—
0x11C		GPT13_OVF	✓	✓	✓	—	—	—
0x11D		GPT13_UDF	✓	✓	✓	—	—	—
0x120	Ether	ETHER_EINT0	✓	—	—	—	—	—
0x121	USBHS	USBHS_D0FIFO	✓	✓	✓	—	—	—
0x122		USBHS_D1FIFO	✓	✓	✓	—	—	—
0x123		USBHS_USBIR	✓	—	—	✓	✓	✓*5

Table 13.4 Event table (7 of 10)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x124	SCI0	SCI0_RXI	✓	✓	✓	—	—	—
0x125		SCI0_TXI	✓	✓	✓	—	—	—
0x126		SCI0_TEI	✓	—	—	—	—	—
0x127		SCI0_ERI	✓	—	—	—	—	—
0x128		SCI0_AED	✓	—	—	—	—	—
0x129		SCI0_BFD	✓	—	—	—	—	—
0x12A		SCI0_AM	✓	—	—	—	—	—
0x12B	SCI1	SCI1_RXI	✓	✓	✓	—	—	—
0x12C		SCI1_TXI	✓	✓	✓	—	—	—
0x12D		SCI1_TEI	✓	—	—	—	—	—
0x12E		SCI1_ERI	✓	—	—	—	—	—
0x12F		SCI1_AED	✓	—	—	—	—	—
0x130		SCI1_BFD	✓	—	—	—	—	—
0x131		SCI1_AM	✓	—	—	—	—	—
0x132	SCI2	SCI2_RXI	✓	✓	✓	—	—	—
0x133		SCI2_TXI	✓	✓	✓	—	—	—
0x134		SCI2_TEI	✓	—	—	—	—	—
0x135		SCI2_ERI	✓	—	—	—	—	—
0x138		SCI2_AM	✓	—	—	—	—	—
0x139	SCI3	SCI3_RXI	✓	✓	✓	—	—	—
0x13A		SCI3_TXI	✓	✓	✓	—	—	—
0x13B		SCI3_TEI	✓	—	—	—	—	—
0x13C		SCI3_ERI	✓	—	—	—	—	—
0x13F		SCI3_AM	✓	—	—	—	—	—
0x140	SCI4	SCI4_RXI	✓	✓	✓	—	—	—
0x141		SCI4_TXI	✓	✓	✓	—	—	—
0x142		SCI4_TEI	✓	—	—	—	—	—
0x143		SCI4_ERI	✓	—	—	—	—	—
0x146		SCI4_AM	✓	—	—	—	—	—
0x163	SCI9	SCI9_RXI	✓	✓	✓	—	—	—
0x164		SCI9_TXI	✓	✓	✓	—	—	—
0x165		SCI9_TEI	✓	—	—	—	—	—
0x166		SCI9_ERI	✓	—	—	—	—	—
0x169		SCI9_AM	✓	—	—	—	—	—
0x178	SPI0	SPI0_SPRI	✓	✓	✓	—	—	—
0x179		SPI0_SPTI	✓	✓	✓	—	—	—
0x17A		SPI0_SPII	✓	—	—	—	—	—
0x17B		SPI0_SPEI	✓	—	—	—	—	—
0x17C		SPI0_SPCEND	✓	—	—	—	—	—

Table 13.4 Event table (8 of 10)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x17D	SPI1	SPI1_SPRI	✓	✓	✓	—	—	—
0x17E		SPI1_SPTI	✓	✓	✓	—	—	—
0x17F		SPI1_SPII	✓	—	—	—	—	—
0x180		SPI1_SPEI	✓	—	—	—	—	—
0x181		SPI1_SPCEND	✓	—	—	—	—	—
0x182	OSPI0	OSPI0_ERR	✓	—	—	—	—	—
0x183		OSPI0_CMP	✓	—	—	—	—	—
0x185	CANFD	CAN_RXF	✓	—	—	—	—	—
0x186		CAN_GLERR	✓	—	—	—	—	—
0x187		CAN0_RF_DMAREQ0	✓	✓	✓	—	—	—
0x188		CAN0_RF_DMAREQ1	✓	✓	✓	—	—	—
0x18B		CAN1_RF_DMAREQ0	✓	✓	✓	—	—	—
0x18C		CAN1_RF_DMAREQ1	✓	✓	✓	—	—	—
0x18F		CAN0_TX	✓	—	—	—	—	—
0x190		CAN0_CHERR	✓	—	—	—	—	—
0x191		CAN0_COMFRX	✓	—	—	—	—	—
0x192		CAN0_CF_DMAREQ	✓	✓	✓	—	—	—
0x193		CAN0_RXMB	✓	—	—	—	—	—
0x194		CAN1_TX	✓	—	—	—	—	—
0x195		CAN1_CHERR	✓	—	—	—	—	—
0x196		CAN1_COMFRX	✓	—	—	—	—	—
0x197		CAN1_CF_DMAREQ	✓	✓	✓	—	—	—
0x198		CAN1_RXMB	✓	—	—	—	—	—
0x19B		CAN0_MRAM_ERI	✓	—	—	—	—	—
0x19C		CAN1_MRAM_ERI	✓	—	—	—	—	—

Table 13.4 Event table (9 of 10)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x19D	I3C	I3C_RESP	✓	✓	✓	—	—	—
0x19E		I3C_CMD	✓	✓	✓	—	—	—
0x19F		I3C_IBI	✓	✓	✓	—	—	—
0x1A0		I3C_RX	✓	✓	✓	—	—	—
0x1A1		I3C_TX	✓	✓	✓	—	—	—
0x1A2		I3C_RCV	✓	✓	✓	—	—	—
0x1A3		I3C_HRESP	✓	✓	✓	—	—	—
0x1A4		I3C_HCMD	✓	✓	✓	—	—	—
0x1A5		I3C_HRX	✓	✓	✓	—	—	—
0x1A6		I3C_HTX	✓	✓	✓	—	—	—
0x1A7		I3C_TEND	✓	—	—	—	—	—
0x1A8		I3C_EEI	✓	—	—	—	—	—
0x1A9		I3C_STEV	✓	—	—	—	—	—
0x1AA		I3C_MREFOVF	✓	—	—	—	—	—
0x1AB		I3C_MREFCPT	✓	—	—	—	—	—
0x1AC		I3C_AMEV	✓	—	—	—	—	—
0x1AD		I3C_WU	✓	—	—	✓	✓	—
0x1AE	ADC120	ADC120_ADI	✓	✓	✓	—	—	—
0x1AF		ADC120_GBADI	✓	✓	✓	—	—	—
0x1B0		ADC120_CMPAI	✓	—	—	—	—	—
0x1B1		ADC120_CMPBI	✓	—	—	—	—	—
0x1B2		ADC120_WCMPM	—	✓	✓	—	—	—
0x1B3		ADC120_WCMPUM	—	✓	✓	—	—	—
0x1B4	ADC121	ADC121_ADI	✓	✓	✓	—	—	—
0x1B5		ADC121_GBADI	✓	✓	✓	—	—	—
0x1B6		ADC121_CMPAI	✓	—	—	—	—	—
0x1B7		ADC121_CMPBI	✓	—	—	—	—	—
0x1B8		ADC121_WCMPM	—	✓	✓	—	—	—
0x1B9		ADC121_WCMPUM	—	✓	✓	—	—	—
0x1BA	DOC	DOC_DOPCI	✓	—	—	—	—	—
0x1BC	RSIP-E51A	RSIP_TADI	✓	—	—	—	—	—
0x1CD	GLCDC	GLCDC_VPOS	✓	—	—	—	—	—
0x1CE		GLCDC_L1UNDF	✓	—	—	—	—	—
0x1CF		GLCDC_L2UNDF	✓	—	—	—	—	—
0x1D0	DRW	DRW_IRQ	✓	—	—	—	—	—

Table 13.4 Event table (10 of 10)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x1D3	MIPI DSI	DSI_SEQ0	✓	—	—	—	—	—
0x1D4		DSI_SEQ1	✓	—	—	—	—	—
0x1D5		DSI_VIN1	✓	—	—	—	—	—
0x1D6		DSI_RCV	✓	—	—	—	—	—
0x1D7		DSI_FERR	✓	—	—	—	—	—
0x1D8		DSI_PPI	✓	—	—	—	—	—
0x1DA	CEU	CEU_CEUI	✓	—	—	—	—	—

Note 1. Only supported when CMPCTL.CSTEN = 1.
Note 2. Only the first edge detection is valid.
Note 3. Support only interrupt after DTC transfer.
Note 4. Only supported in Deep Software Standby mode 1 and Deep Software Standby mode 2.
Note 5. Only supported in Deep Software Standby mode 1.

13.4 Maskable Interrupt Operation

The ICU performs the following functions:

- Detecting interrupts
- Enabling and disabling interrupts
- Selecting interrupt request destinations such as CPU interrupt, DTC activation.

13.4.1 Detecting Interrupts

The ICU selects an event source input from a peripheral function interrupt or an external pin interrupt with IELSRn.IELS [8:0].

The accepted interrupt source sets the IELSRn.IR to 1 and sends an interrupt request to the NVIC.

External pin interrupt requests are detected by either:

- Edges (falling edge, rising edge, or rising and falling edges)
- Level (low level) of the interrupt signal.

Set the IRQCRi.IRQMD[1:0] bits to select the detection mode for the IRQi pins. For interrupt sources associated with peripheral modules, see Table 13.3 and Table 13.4. Events must be accepted by the NVIC before an interrupt occurs and is accepted by the CPU.

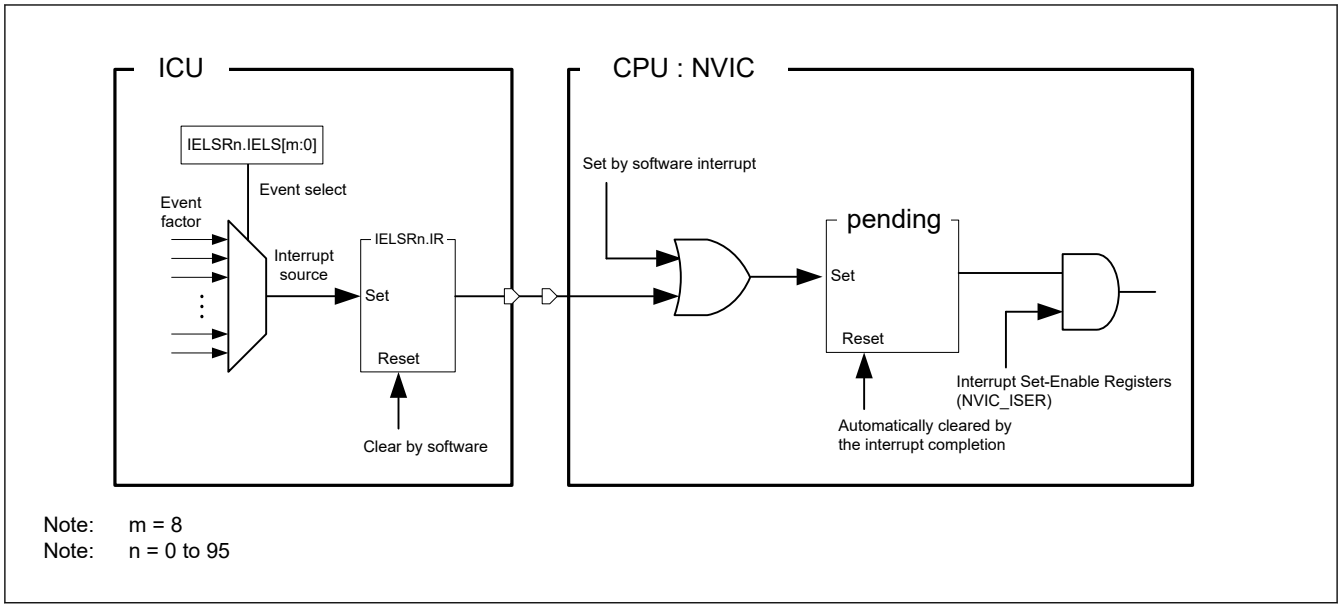


Figure 13.2 Interrupt path of the ICU and CPU (NVIC)

13.5 Maskable Interrupt Setting Procedure

13.5.1 Operations During an Interrupt

-When an interrupt is generated: except for software interrupts:

The IELSRn.IR (n = 0 to 95) flag and Interrupt Set Pending Register (NVIC_ISPRn) are set.

- When an interrupt is generated: software interrupts:

Interrupt Set Pending Register (NVIC_ISPRn) are set.

- When an interrupt is complete:

Clear the IELSRn.IR (n = 0 to 95) flag by software. The Interrupt Set Pending Register (NVIC_ISPRn) is automatically cleared.

Note: There may be a difference in processing speed between the CPU and ICU, and the CPU may exit the interrupt handler before clearing the IR. Then the CPU will accidentally jump to the interrupt handler again. To avoid this, be sure to read IELSRn (n = 0 to 95) register before exiting the interrupt handler and make sure that IR is cleared before exiting the interrupt handler.

13.5.2 Enabling Interrupt Requests

The procedure for enabling an interrupt request is as follows:

1. Set the Interrupt Set-Enable register (NVIC_ISER).
2. Set the IELSRn.IELS[8:0] bits as the interrupt source.
3. Specify the operation settings for the event source, such as Deep Sleep/Software Standby mode cancellation (WUPENn register setting).

13.5.3 Disabling Interrupt Requests

The procedure to disable the interrupt request is as follows:

1. Disable the operation settings for the event source, such as Deep Sleep/Software Standby mode cancellation (WUPENn register setting).
2. Clear the interrupt source setting (IELSRn.IELS[8:0] = 0x00).
3. Clear the interrupt status flag (IELSRn.IR = 0).
4. Clear the interrupt Clear-Enable register (NVIC_ICER) and interrupt Clear-Pending register (NVIC_ICPR).

13.5.4 Polling for Interrupts

The procedure for polling for interrupt requests is as follows:

1. Set the Interrupt Clear-Enable register (NVIC_ICER).
2. Set the IELSRn.IELS[8:0] bits as the interrupt source.
3. Specify the operation settings for the event source, such as DMAC activation (DELSRn.DELS[8:0]), Deep Sleep/ Software Standby mode cancellation (WUPENn register setting).
4. Poll the interrupt Set-Pending register (NVIC_ISPR).

13.5.5 Selecting Interrupt Request Destinations

The available destinations are fixed for each interrupt, as described in Table 13.3, Table 13.4.

The interrupt output destination, CPU and DTC can be independently selected for each interrupt source.

Use an interrupt request destination setting that is indicated by a "✓" in the event list (see section 13.3.2. Event Number).

Note: Setting the same interrupt source for IELSRn and DELSRn is prohibited.

If the DMAC or DTC is selected as the destination for requests from an IRQi pin, you must set the IRQCRi.IRQMD[1:0] bits for that interrupt to select edge detection.

13.5.5.1 CPU Interrupt Request

When IELSRn.DTCE = 0, the event specified in the IELSRn register is output to the NVIC.

13.5.5.2 DTC Activation

When IELSRn.DTCE = 1, the event specified in the IELSRn register is output to the DTC. Use the following procedure:

1. Set the IELSRn.IELS[8:0] bits to the target event and set the IELSRn.DTCE bit to 1.
2. Set the DTC Module Start bit (DTCST.DTCST) to 1.

Table 13.5 shows operation when the DTC is the interrupt request destination.

Table 13.5 Operation when DTC becomes interrupt request destination

Interrupt request destination	DISEL*1	Remaining transfer operations	Operation per request	IR*2	Interrupt request destination after transfer
DTC*3	1	≠ 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	DTC
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	CPU (IELSRn.DTCE bit is automatically cleared)
	0	≠ 0	DTC transfer	Cleared at the start of DTC data transfer after reading DTC transfer data	DTC
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	CPU (IELSRn.DTCE bit is automatically cleared)

Note 1. DTC.MRB.DISEL bit controls the interrupt generates timing from DTC to CPU.
Note 2. When the IELSRn.IR flag is 1, an interrupt request (DTC activation request) that occurs again is ignored.
Note 3. For chain transfers, DTC transfer continues until the last chain transfer ends. The DISEL bit state and the remaining transfer count determine whether a CPU interrupt occurs, the IELSRn.IR flag clear timing, and the interrupt request destination after transfer. See Table 17.2 in section 17, Data Transfer Controller (DTC).

Note: Error during DTC transfer

If an error response occurs during DTC transfer, the DTC notifies the ICU that an error has occurred. ICU clears all bits of the target IELSRn (n = 0 to 95). IELSRn that is not the target is not cleared.

13.5.6 Digital Filter

A digital filter function is provided for the external interrupt request pins IRQi, (i = 0 to 15) and the NMI pin interrupt. It samples input signals on the filter PCLKB sampling clock and removes any signal with a pulse width less than 3 sampling cycles.

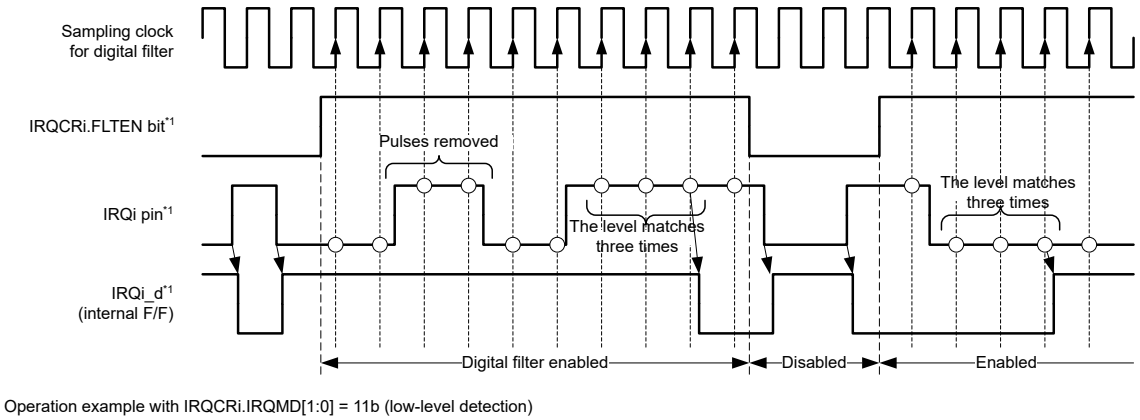
To use the digital filter for an IRQi pin:

1. Set the sampling clock cycle to PCLKB, PCLKB/8, PCLKB/32, or PCLKB/64 in the IRQCRi.FCLKSEL[1:0] bits (i = 0 to 15).
2. Set the IRQCRi.FLTEN bit (i = 0 to 15) to 1 (digital filter enabled).

To use the digital filter for an NMI pin:

1. Set the sampling clock cycle to PCLKB, PCLKB/8, PCLKB/32, or PCLKB/64 in the NMICR.NFCLKSEL[1:0] bits.
2. Set the NMICR.NFLTEN bit to 1 (digital filter enabled).

Figure 13.3 shows an example of digital filter operation.



Note 1. i = 0 to 15

Figure 13.3 Digital filter operation example

During Software Standby mode or Deep Software Standby mode, the digital filter is forcibly turned off by Hardware. After returning to normal mode from Software Standby mode or Deep Software Standby mode, follow the values of IRQCRi.FLTEN (i = 0 to 15) and NMICR.NFLTEN. Once the filter is disabled, the event information sampled up to that point will be lost.

13.5.7 External Pin Interrupts

To use external pin interrupts:Configure I/O ports settings.

1. Clear the IRQCRi.FLTEN bit (i = 0 to 15) to 0 (digital filter disabled).
2. Specify or confirm the I/O port settings.
3. Set the IRQMD[1:0] the FCLKSEL[1:0] bits, and the FLTEN bit of the IRQCRi register (i = 0 to 15) register.
4. Select the IRQ pin as follows:
   - If the IRQ pin is to be used for CPU interrupt requests, set the IELSRn.IELS[8:0] (n = 0 to 95) bits and the IELSRn.DTCE (n = 0 to 95) bit to 0.
   - If the IRQ pin is to be used for DTC activation, set the IELSRn.IELS[8:0] (n = 0 to 95) bits and the IELSRn.DTCE (n = 0 to 95) bit to 1

- If the IRQ pin is to be used for DMAC activation, set the DMAC.DELSRn.DELS[8:0] (n = 0 to 7) bits.

13.6 Non-Maskable Interrupt Operation

The following sources can trigger a non-maskable interrupt:

- NMI pin interrupt
- Oscillation stop detection interrupt
- WDT underflow/refresh error interrupt
- IWDT underflow/refresh error interrupt
- Voltage monitor 1 interrupt
- Voltage monitor 2 interrupt
- Common memory error interrupt
- Bus error interrupt
- CPU Lockup error interrupt

Non-maskable interrupts can only be used with the CPU, not to activate the DTC. Non-maskable interrupts take precedence over all other interrupts. The non-maskable interrupt states can be verified in the Non-Maskable Interrupt Status Register (NMISR). Confirm that all bits in the NMISR are 0 before returning from the NMI handler.

There may be a difference in processing speed between the CPU and ICU, and the CPU may exit the interrupt handler before clearing the NMISR. Then the CPU will accidentally jump to the NMI handler again. To avoid this, be sure to read NMISR before exiting the NMI handler and make sure that NMISR is cleared before exiting the NMI handler.

Non-maskable interrupts are disabled by default. To use non-maskable interrupts:

1. Clear the NMICR.NFLTEN bit to 0 (digital filter disabled).
2. Set the NMIMD bit, NFCLKSEL[1:0] bits, and NFLTEN bit of NMICR register.
3. Write 1 to the NMICLR.NMICLR bit to clear the NMISR.NMIST flag to 0.
4. Enable the non-maskable interrupt by writing 1 to the associated bit in the Non-Maskable Interrupt Enable Register (NMIER).

After 1 is written to the NMIER register, subsequent write access to the NMIEN bit in NMIER is ignored. An NMI cannot be disabled when enabled, except by a reset.

13.6.1 Correspondence to TrustZone-M by NMI

The NMI security is set by AIRCR.BFHFNMINS.

Although there is only one NMI per CPU, multiple factors can be set. This section describes the procedure for mixing Secure and Non-secure programs of NMI. When doing so, the NMI-related registers of the ICU are set to Secure.

Register related to Non-Maskable Interrupt :

- NMISR
- NMIER
- NMICLR
- NMICR

Figure 13.4 shows the flow.

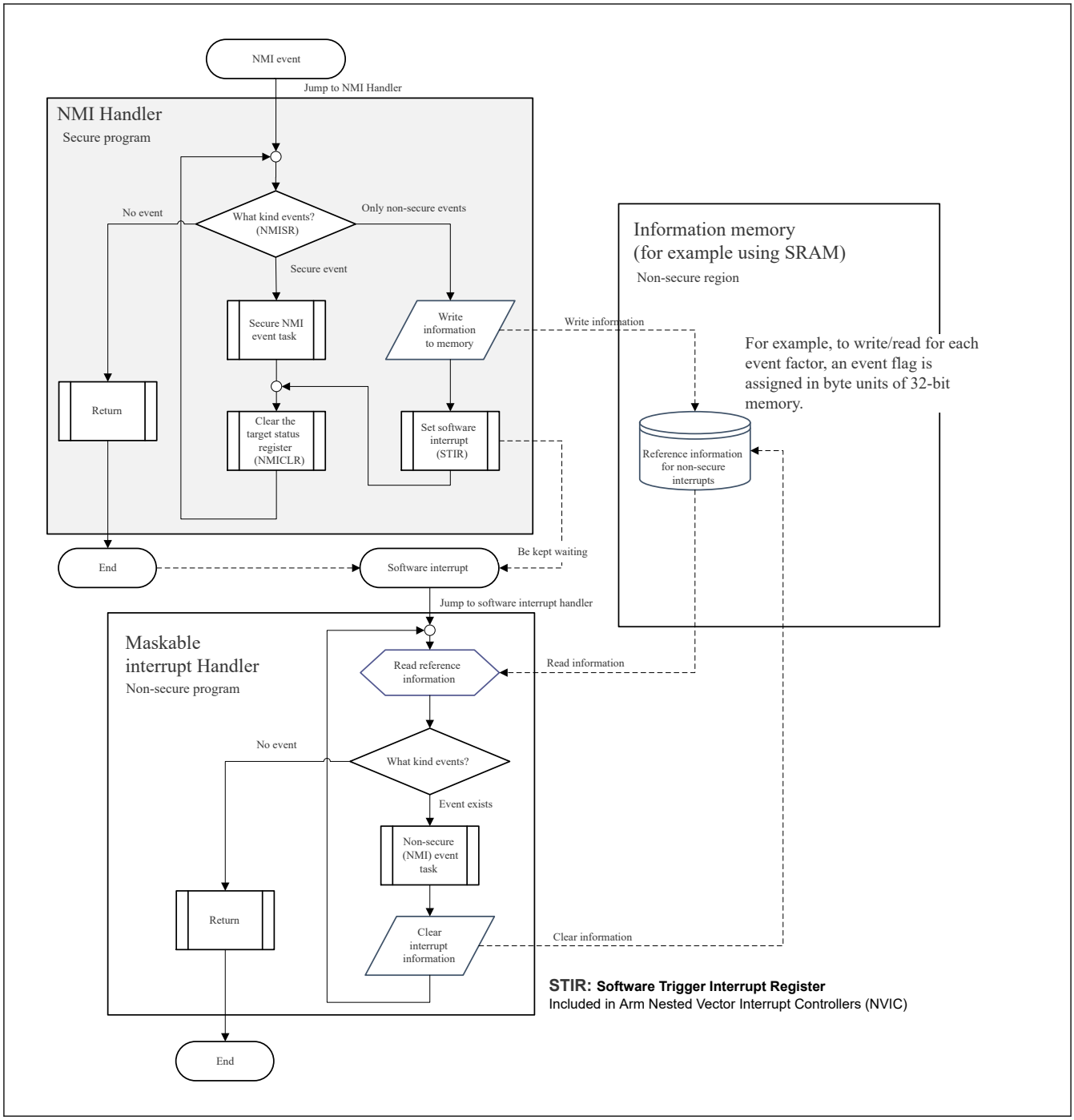


Figure 13.4 Correspondence to TrustZone-M by NMI

13.7 Security

13.7.1 Security Related to CPU Interrupt Inputs

Non-maskable interrupt

Arm CPU NMI Secure is modified by AIRCR.BFHFNMINS and is managed by the software developer who manages Secure.

The following registers must follow the secure attribution set in AIRCR.BFHFNMINS.

- ICU.NMISR
- ICU.NMIER

- ICU.NMICLR

Therefore, the secure attribute set by ICUSARB must match the secure attribute set by AIRCR.BFHFNMINS.

Maskable interrupts

The secure of maskable interrupts are set in the Arm CPU NVIC internal registers (NVIC_ITNS0 to NVIC_ITNS15).

Which maskable interrupts are secured is controlled by the software developer who manages secure.

The following registers must follow the secure attribution set in NVIC_ITNS0 to NVIC_ITNS15.

- ICU.IELSRn (n = 0 to 95)

Therefore, the secure attribute set by ICUSARG, ICUSARH, and ICUSARI must match the secure attribute set by NVIC_ITNS0 to NVIC_ITNS15.

13.7.2 Trusted Interrupt Management

The secure-interrupt should not be visible to non-secure-operation. Because there is a risk that vital information will be leaked to the attacker, the on-chip interrupt network can route any interrupt to secure or non-secure world. However, secure interrupt routing must be suppressed so that it can only be configured from a secure world.

When set TEVTRCR.TEVTE = 1, secure program manages the selection of interrupt factors. It protects non-secure programs from using and monitoring secure interrupt factors without permission. Specifically, when TEVTRCR.TEVTE = 1, the write permission of IELSR.IELS [8:0] is limited to the secure attribute, and the write by the non-secure attribute is ignored. TEVTRCR.TEVTE protects only IELS [8:0], and the security attributes of IR and DTCE follow the settings of ICUSARG, ICUSARH, and ICUSARI. If TEVTRCR.TEVTE = 1, it is necessary to set all IELS including non-secure interrupts in the secure program or prepare a secure API to set IELS in response to the request from the non-secure program.

13.7.3 Trusted IELSR Setting Procedure

13.7.3.1 Case Where Secure Program Sets all IELSRs

Case where all settings are performed in the secure initial sequence.

Initial IELSR setting procedure after system reset release

1. Secure program sets TEVTE=1
2. Secure program sets security attributes of all interrupts (NVIC_ITNS, Security Attribution of ICU.IELSRn.IELS)
3. Secure program selects all interrupt source (ICU.IELSRn.IELS)
4. Secure program allows secure interrupts (NVIC_ISER for secure interrupt)
5. Secure program jumps to non-secure program
6. Non-secure program allows non-secure interrupts (NVIC_ISER for non-secure interrupt)

13.7.3.2 Case Where Non-secure Program Sets IELSRn by Secure API

Case of setting upon receiving a request from a non-secure program.

Initial IELSR setting procedure after system reset release

1. Secure program sets TEVTE=1
2. Secure program sets security attributes of all interrupts (NVIC_ITNS, Security Attribution of ICU.IELSRn.IELS)
3. Secure program selects secure interrupt source (ICU.IELSRn.IELS)
4. Secure program enables secure interrupt (NVIC_ISER for secure interrupt)
5. Secure program jumps to non-secure program
6. Non-secure program calls secure program (API call)
7. Secure program selects the cause of non-secure interrupt (ICU.IELSRn.IELS)
8. Secure program returns to non-secure program
9. Non-secure program allows non-secure interrupt (NVIC_ISER for non-secure interrupt)

13.7.3.3 IELSR Release Procedure When TEVTE = 1

< When clearing interrupt settings of secure attribute by the secure program >

1. Clear the interrupt source setting (IELSRn.IELS = 0x00)
2. Clear the interrupt status flag (ICU.IELSRn.IR = 0)
3. Clear the Interrupt Clear Enable register (NVIC_ICPR)

<When clearing interrupt settings of non-secure attribute by the secure program >

1. Clear the interrupt source setting (IELSRn.IELS =0x00)
2. Clear the interrupt status flag by using the non-secure alias address (IELSRn.IR = 0)
3. Clear the Interrupt Clear Enable register (NVIC_ICPR)

Secure can also rewrite NVIV_ICPRn_NS .

< When clearing interrupt settings of secure attribute by the non-secure program >

It cannot be canceled.

< When clearing interrupt settings of non-secure attribute by non-secure program>

1. Non-secure program calls secure program (API call)
2. Clear the interrupt source setting (IELSRn.IELS = 0x00)
3. Return from secure program to non-secure program
4. Clear the interrupt status flag (IELSRn.IR = 0)
5. Clear the Interrupt Clear Enable register (NVIC_ICPR)

13.8 Return from Low Power Modes

Table 13.4 lists the interrupt sources that can be used to exit Sleep, Deep Sleep, Software Standby mode or Deep Software Standby mode. For more information, see section 10, Low Power Modes.

Note: The return factor from low power mode must have a priority that the CPU can accept.

13.8.1 Return from CPU Sleep Mode

To return from CPU Sleep mode in response to an interrupt:

non-maskable interrupt

- Use the NMIER register to enable the target interrupt request.

maskable interrupt

- Select the CPU as the interrupt request destination.
- Enable the interrupt in the NVIC.

13.8.2 Return from CPU Deep Sleep Mode

The ICU returns from CPU Deep Sleep mode using a non-maskable interrupt or a maskable interrupt. For maskable interrupt of canceling source, see Table 13.4 .

To return from CPU Deep Sleep mode:

Non-maskable interrupt

Use the NMIER register to enable the target interrupt request.

Maskable interrupt

1. Select the interrupt source that enables return from CPU Deep Sleep mode.

2. Use the WUPENn register to enable the target interrupt request.
3. Select the CPU as the interrupt request destination.
4. Enable the interrupt in the NVIC.

Interrupt requests through the IRQi (i = 0 to 15) pins that do not satisfy these conditions are not detected while the clock is stopped in CPU Deep Sleep mode. Similarly, it cannot detect a request for a non-maskable interrupt from a request source whose clock is stopped in CPU Deep Sleep mode. Therefore, those events cannot be used as return factors. For example, the CPU cannot return from software standby mode because the following NMI events do not occur.

- CPU Lockup error interrupt

13.8.3 Return from Software Standby Mode

MCU goes to the Software Standby mode via CPU Deep sleep state.

The ICU returns from Software Standby mode using a non-maskable interrupt or a maskable interrupt. For maskable interrupt of canceling source, see Table 13.4.

Waking up from Software Standby mode is triggered by the same event as CPU Deep Sleep mode. No event occurs from the function that stops in Software Standby mode. Therefore, those events cannot be used as return factors. For example, the CPU cannot return from Software Standby mode because the following NMI events do not occur.

- WDT underflow/refresh error
- Common memory error interrupt
- Bus error interrupt
- CPU Lockup error interrupt

13.9 Using the WFI Instruction with Non-Maskable Interrupts

Whenever a WFI instruction is executed, confirm that all status flags in the NMISR register are 0.

13.10 Reference

- Arm Limited., Arm® Cortex®-M85 Processor Technical Reference Manual (101924_0002_05_en)

14. Buses

14.1 Overview

Table 14.1 lists the bus masters and bus slaves, Figure 14.1 shows the system bus configuration, and Table 14.3 lists the addresses assigned for each bus.

Table 14.1 Bus specifications

Classification	Bus master/ slave name	Sync clock	Description
Bus masters	CPUMAXIBI (Arm® Cortex®-M85)	CPUCLK	Connected to the CPU Master-AXI (M-AXI) Interface
	CPUPAHBI (Arm® Cortex®-M85)	CPUCLK	Connected to the CPU Peripheral AHB (P-AHB) Interface
	DMAC/DTCBI	ICLK	Connected to the DMAC/DTC Interface
	EDMACBI	PCLKA	Connected to the Ether DMAC Interface
	CEUBI	PCLKA	Connected to the CEU Interface
	GLCDC0BI	PCLKA	Connected to the GLCDC0 Interface
	GLCDC1BI	PCLKA	Connected to the GLCDC1 Interface
	DRW0BI	PCLKA	Connected to the DRW0 Interface
	DRW1BI	PCLKA	Connected to the DRW1 Interface
	MIPIBI	PCLKA	Connected to the MIPI-DSI Interface
Bus slaves	FHBI	ICLK	Connected to Code Flash memory and Configuration area
	FLBI	FCLK	Connected to Data Flash memory and FACI
	CPUSAHBI	CPUCLK	Connected to CPU S-AHB
	S0BI	ICLK	Connected to SRAM0
	S1BI	ICLK	Connected to SRAM1
	STBYSBI	ICLK	Connected to Standby SRAM
	ECBI	BCLK	Connected to the external devices (External Memory Interface)
	EOBI	PCLKA	Connected to the OSPI (External Memory Interface)
	PBBI	PCLKB	Connected to peripheral modules synchronizes with PCLKB
	PABI	PCLKA	Connected to peripheral modules synchronizes with PCLKA
	PIBI	ICLK	Connected to peripheral modules synchronizes with ICLK
	ICUBI	ICLK	Connected to ICU controller
	PSBI	ICLK	Connected to peripheral system modules (MPU, CSC/SDRAM, SRAM, Debug component, Flash controller, Bus controller, common ICU controller, DMAC/DTC, CPU controller and security attribution controller)
		PCLKB	Connected to peripheral system module. (system controller)
		DCLK	Connected to peripheral system module. (debug controller)

Note: BCLK (external bus clock) : 120 MHz (max.) (The CSC (CS area controller) operates in synchronization with the BCLK).
BCLK pin output: The frequency is the same as the default of BCLK. Half of BCLK can be supplied by setting the EBCLK pin and the output select bit (BCKCR.BCLKDIV) in the External Bus Clock Control register. For details, see section 8, Clock Generation Circuit
CPUMAXIBI : CPU M-AXI Bus Interface
CPUPAHBI : CPU P-AHB Bus Interface
DMAC/DTCBI : DMAC/DTC Bus Interface
EDMACBI : EDMAC (Ether) Bus Interface
GLCDC0BI : GLCDC (graphic 1) Bus Interface
GLCDC1BI : GLCDC (graphic 2) Bus Interface
DRW0BI : DRW (texture) Bus Interface
DRW1BI : DRW (data) Bus Interface
CEUBI : CEU Bus Interface

MIPIBI : MIPI Bus Interface
FHBI : Flash High-speed Bus Interface
FLBI : Flash Low-speed Bus Interface
CPUSAHBI : CPU S-AHB Interface
S0BI : SRAM0 Bus Interface
S1BI : SRAM1 Bus Interface
STBYSBI : Standby SRAM Bus Interface
ECBI : External memory CSC and SDRAM Bus Interface
EOBI : External memory OSPI Bus Interface
PBBI : Peripheral synchronizes with PCLKB Bus Interface
PABI : Peripheral synchronizes with PCLKA Bus Interface
PIBI : Peripheral synchronizes with ICLK Bus Interface
ICUBI : ICU Controller Bus Interface
PSBI : Peripheral System Bus Interface

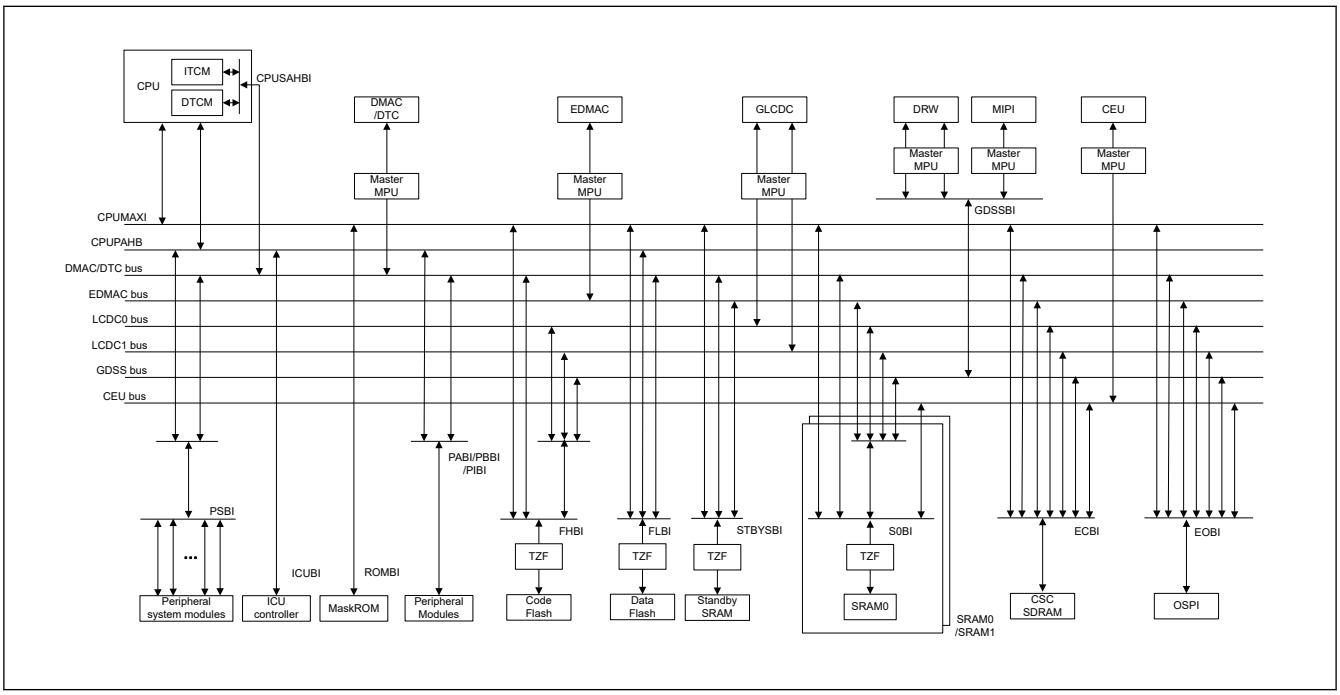


Figure 14.1 System bus connection

Table 14.2 System bus access path

		Master							
	Name	**CPUMAXIBI**	**CPUPAHBI**	**DMAC/DTCBI**	**EDMACBI**	**GLCDC0BI**	**GLCDC1BI**	**GDSSBI*1**	**CEUBI**
Slave	PSBI	F	T	T	F	F	F	F	F
	ICUBI	F	T	F	F	F	F	F	F
	CPUSAHBI	F	F	T	F	F	F	F	F
	PIBI	F	T	T	F	F	F	F	F
	PABI	F	T	T	F	F	F	F	F
	PBBI	F	T	T	F	F	F	F	F
	FHBI	T	F	T	F	T	T	T	F
	FLBI	T	T	T	F	F	F	F	F
	STBYSBI	T	F	T	T	F	F	F	F
	S0BI	T	F	T	T	T	T	T	T
	S1BI	T	F	T	T	T	T	T	T
	ECBI	T	F	T	T	T	T	T	T
	EOBI	T	F	T	T	T	T	T	T

Note: TZF is TrustZone Filter for memory resources. CPUMAXIBI accesses Data Flash Memory and Configuration area through FLBI. CPUPAHBI accesses FACI (Peripheral region) through FLBI.
Note 1. GDSSBI is a BUS interface that combines DRW0BI, DRW1BI and MIPIBI.

T : The master can access slaves

F : The master cannot access slaves

Table 14.3 Addresses assigned for each slave (1 of 2)

Region	Address	Bus Interface	Area
Code	0x0000_0000 to 0x0000_FFFF (Non-secure callable or Secure) 0x1000_0000 to 0x1000_FFFF (Non-secure)	CPUSAHBI	ITCM
	0x0200_0000 to 0x022F_7FFF (Non-secure callable or Secure) 0x1200_0000 to 0x122F_7FFF (Non-secure)	FHBI	Code Flash memory
	0x0300_8000 to 0x0300_A2FF (Non-secure callable or Secure) 0x1300_8000 to 0x1300_A2FF (Non-secure)	FHBI	Factory Flash and option-setting memory
Data	0x2000_0000 to 0x2000_FFFF (Non-secure callable or Secure) 0x3000_0000 to 0x3000_FFFF (Non-secure)	CPUSAHBI	DTCM
	0x2200_0000 to 0x2205_FFFF (Non-secure callable or Secure) 0x3200_0000 to 0x3205_FFFF (Non-secure)	S0BI	SRAM0
	0x2206_0000 to 0x220D_FFFF (Non-secure callable or Secure) 0x3206_0000 to 0x320D_FFFF (Non-secure)	S1BI	SRAM1
	0x2600_0000 to 0x2600_03FF (Non-secure callable or Secure) 0x3600_0000 to 0x3600_03FF (Non-secure)	STBYSBI	Standby SRAM
	0x2700_0000 to 0x2700_2FFF (Non-secure callable or Secure) 0x3700_0000 to 0x3700_2FFF (Non-secure)	FLBI	Data Flash memory
	0x2703_0050 to 0x2703_03FF (Non-secure callable or Secure) 0x3703_0050 to 0x3703_03FF (Non-secure)	FLBI	option-setting memory

Table 14.3 Addresses assigned for each slave (2 of 2)

Region	Address	Bus Interface	Area
Peripheral	0x4000_0000 to 0x4000_0FFF (Secure) 0x5000_0000 to 0x5000_0FFF (Non-secure)	PSBI	MPU controller
	0x4000_2000 to 0x4000_2FFF (Secure) 0x5000_2000 to 0x5000_2FFF (Non-secure)		SRAM controller
	0x4000_3000 to 0x4000_3FFF (Secure) 0x5000_3000 to 0x5000_3FFF (Non-secure)		CSC/SDRAM controller
	0x4000_4000 to 0x4000_4FFF (Secure) 0x5000_4000 to 0x5000_4FFF (Non-secure)		BUS controller
	0x4000_6000 to 0x4000_6FFF (Secure) 0x5000_6000 to 0x5000_6FFF (Non-secure)		Common ICU controller
	0x4000_8000 to 0x4000_8FFF (Secure) 0x5000_8000 to 0x5000_8FFF (Non-secure)		Security attribution controller
	0x4000_A000 to 0x4000_AFFF (Secure) 0x5000_A000 to 0x5000_AFFF (Non-secure)		DMAC/DTC
	0x4000_C000 to 0x4000_CFFF (Secure) 0x5000_C000 to 0x5000_CFFF (Non-secure)	ICUBI	ICU controller
	0x4000_F000 to 0x4000_FFFF (Secure) 0x5000_F000 to 0x5000_FFFF (Non-secure)	PSBI	CPU controller
	0x4001_0000 to 0x4001_AFFF (Secure) 0x5001_0000 to 0x5001_AFFF (Non-secure)		Debug component
	0x4001_B000 to 0x4001_BFFF (Secure) 0x5001_B000 to 0x5001_BFFF (Non-secure)		Debug controller
	0x4001_C000 to 0x4001_CFFF (Secure) 0x5001_C000 to 0x5001_CFFF (Non-secure)		Flash controller
	0x4001_E000 to 0x4001_EFFF (Secure) 0x5001_E000 to 0x5001_EFFF (Non-secure)		System controller
	0x4010_0000 to 0x401F_FFFF (Secure) 0x5010_0000 to 0x501F_FFFF (Non-secure)	FLBI	Flash register
	0x4020_0000 to 0x402F_FFFF (Secure) 0x5020_0000 to 0x502F_FFFF (Non-secure)	PBBI	Peripheral synchronizes with PCLKB
	0x4030_0000 to 0x403F_FFFF (Secure) 0x5030_0000 to 0x503F_FFFF (Non-secure)	PABI	Peripheral synchronizes with PCLKA
	0x4040_0000 to 0x404F_FFFF (Secure) 0x5040_0000 to 0x504F_FFFF (Non-secure)	PIBI	Peripheral synchronizes with ICLK
External RAM	0x6000_0000 to 0x6FFF_FFFF	ECBI	CSC/SDRAM (external bus)
	0x8000_0000 to 0x9FFF_FFFF	EOBI	OSPI

14.2 Description of Buses

14.2.1 Arbitration

For arbitration between masters in each slave, fixed-priority and round-robin methods can be selected. For details, see section 14.3.24. BUSMABT : Bus Master Arbitration Control Register, section 14.3.25. BUSSABT0<slave> : Bus Slave Arbitration Control Register 0(<slave> = FLBI, STBYSBI, ECBI, EOBI, PBBI, PABI, PIBI, PSBI) and section 14.3.26. BUSSABT1<slave> : Bus Slave Arbitration Control Register 1(<slave> = FHBI, S0BI, S1BI).

14.2.2 External Bus

The external bus controller arbitrates requests for bus access on the external address space from the CPU M-AXI bus, DMAC/DTC bus, EDMAC (Ether) bus, GLCDC0 bus, GLCDC1 bus, DRW0 bus, DRW1 bus, MIPI bus and CEU bus. See section 14.2.1. Arbitration for the priority and arbitration method of each master to the External bus.

The bus system provides an external space for the OSPI (EOBI). See section 37, Octal Serial Peripheral Interface (OSPI).

Table 14.4 lists the external bus specifications and Table 14.5 lists the I/O pins.

Table 14.4 External bus specifications

Parameter	Specifications
External address space	• The external address space is divided into 8 CS areas (CS0 to CS7) and the SDRAM area (SDCS) for management. • Chip select signals can be output for each area. • The bus width can be set for each area. – Separate bus: Selectable to 8-bit, 16-bit or 32-bit bus space – Address/data multiplexed bus: Selectable to 8-bit or 16-bit bus space • Endian mode can be specified for each area.
CS area controller	• Recovery cycles can be inserted: – Read recovery: Up to 15 cycles – Write recovery: Up to 15 cycles • Cycle wait function: Wait for up to 31 cycles (for page access, up to 7 cycles) • Use wait control to set up: – Assertion and negation timing of chip select signals (CS0 to CS7) – Assertion timing of the read signal (RD) and write signals (WR0/WR and WR1 to WR3) – Timing of data output starts and ends • Write access modes: – Single-write strobe mode and byte strobe mode Separate bus or address/data multiplexed bus can be set for each area.
SDRAM area controller	• Multiplexed output of row address and column address (8, 9, 10, or 11 bits) • Self-refresh and auto-refresh selectable • CAS latency can be specified from 1 to 3 cycles.
Write buffer function	When write data from the bus master is written to the write buffer, write access by the bus master is complete.
Frequency	• The CS area controller (CSC) operates in synchronization with the external bus clock (BCLK)*1 • The frequency of the EBCLK pin output is the same as BCLK by default. Half of the BCLK cycles can be supplied by setting the EBCLK Pin Output Select bit, BCKCR.BCLKDIV, in the External Bus Clock Control Register. For more information, see section 8, Clock Generation Circuit. • The SDRAM area controller (SDRAMC) operates in synchronization with the SDRAM clock (SDCLK).

Note 1. BCLK and SDCLK must operate at the same frequency when the SDRAM is in use.

Table 14.5 External bus I/O pins (1 of 2)

Pin name	I/O	Related functions	Description
EBCLK, SDCLK*1	Output	CSC, SDRAMC	Clock output pin
A23 to A00*2	Output	CSC, SDRAMC	Address output pins
D31 to D00 DQ31 to DQ00	I/O	CSC, SDRAMC	D31 to D00 are CSC data input/output pins. DQ31 to DQ00 are SDRAMC data input/output pins. • D31 to D00, DQ31 to DQ00 pins are enabled when the 32-bit bus space is specified. • D15 to D00, DQ15 to DQ00 pins are enabled when the 16-bit bus space is specified. • D07 to D00, DQ07 to DQ00 pins are enabled when the 8-bit bus space is specified.
BC0	Output	CSC	• Strobe signal that indicates (when low) that D07 to D00 are valid during access to an external address space in single-write strobe mode, active-low. • When an 8-bit bus space is specified, this output pin is always held low regardless of the write access mode.
BC1	Output	CSC	• Strobe signal that indicates (when low) that D15 to D08 are valid during access to an external address space in single-write strobe mode, active-low. • This pin is not used when the 8-bit bus space is specified.
BC2	Output	CSC	• Strobe signal that indicates (when low) that D23 to D16 are valid during access to an external address space in single-write strobe mode, active-low. • This pin is not used when the 8- or 16-bit bus space is specified.
BC3	Output	CSC	• Strobe signal that indicates (when low) that D31 to D24 are valid during access to an external address space in single-write strobe mode, active-low. • This pin is not used when the 8- or 16-bit bus space is specified.
CS0*3	Output	CSC	Chip select signal for area 0 (CS0), active-low

Table 14.5 External bus I/O pins (2 of 2)

Pin name	I/O	Related functions	Description
CS1*3	Output	CSC	Chip select signal for area 1 (CS1), active-low
CS2*3	Output	CSC	Chip select signal for area 2 (CS2), active-low
CS3*3	Output	CSC	Chip select signal for area 3 (CS3), active-low
CS4	Output	CSC	Chip select signal for area 4 (CS4), active-low
CS5	Output	CSC	Chip select signal for area 5 (CS5), active-low
CS6	Output	CSC	Chip select signal for area 6 (CS6), active-low
CS7	Output	CSC	Chip select signal for area 7 (CS7), active-low
RD	Output	CSC	Strobe signal that indicates that a read from an external address space (CS0 to CS7) is in progress, active-low.
WR0/WR*4	Output	CSC	• WR0 signal is a strobe signal that indicates that a write to an external address space is in progress in byte strobe mode, and D07 to D00 are valid, active-low. • WR signal is a strobe signal that indicates that a write to an external address space is in progress in single-write strobe mode, active-low. • When an 8-bit bus space is specified, this output pin is held low during a write access regardless of the write access mode.
WR1	Output	CSC	• Strobe signal that indicates that D15 to D08 are valid during a write to an external address space in byte strobe mode, active-low. • This signal is invalid in single-write strobe mode. • This pin is not used when the 8-bit bus space is specified.
WR2	Output	CSC	• Strobe signal that indicates that D23 to D16 are valid during a write to an external address space in byte strobe mode, active-low. • This signal is invalid in single-write strobe mode. • This pin is not used when the 8- or 16-bit bus space is specified.
WR3	Output	CSC	• Strobe signal that indicates that D31 to D24 are valid during a write to an external address space in byte strobe mode, active-low. • This signal is invalid in single-write strobe mode. • This pin is not used when the 8- or 16-bit bus space is specified.
ALE	Output	CSC	Address latch signal when address/data multiplexed bus is selected.
WAIT	Input	CSC	Wait request signal used when accessing the external address space (CS0 to CS7), active-low
CKE	Output	SDRAMC	Clock enable signal
SDCS	Output	SDRAMC	Chip select signal, active-low
RAS	Output	SDRAMC	Row address strobe signal, active-low
CAS	Output	SDRAMC	Column address strobe signal, active-low
WE	Output	SDRAMC	Write enable signal, active-low
DQM0	Output	SDRAMC	I/O data mask enable signal for DQ07 to DQ00
DQM1	Output	SDRAMC	I/O data mask enable signal for DQ15 to DQ08
DQM2	Output	SDRAMC	I/O data mask enable signal for DQ23 to DQ16
DQM3	Output	SDRAMC	I/O data mask enable signal for DQ31 to DQ24

Note 1. The EBCLK and the SDCLK pin functions are shared by the CS area controller (CSC) and the SDRAM area controller (SDRAMC). When using the CSC and the SDRAMC simultaneously, the SDCLK pin function is valid.

Note 2. The A23 to A00 pin functions are shared by the CSC and the SDRAMC.
When using the CSC only:
The A00 and BC0 pin functions share the same pin, and either becomes valid according to the area, with the function being A00 in byte strobe mode and BC0 in single-write strobe mode. Setting the 8-bit external bus width is prohibited in single-write strobe mode.
When using the SDRAMC only:
The A16 to A00 pin functions are valid.
The A00 and DQM1, A01 and DQM3 pin functions share the same pin, and either pin function becomes valid according to the external bus width.
When selecting 8-bit bus width, the pin function is A00 and A01. When selecting 16-bit bus width, the pin function is A0 and DQM1.
When selecting 32-bit bus width, the pin function is DQM1 and DQM3.
When using the CSC and the SDRAMC simultaneously:

The A23 to A17 pin functions are valid for CSC. The A16 to A00 pin functions are shared by the CSC and the SDRAMC. In the SDRAMC functions, the A00 and the DQM1 pin functions work as described.
In the CSC functions, the A00 and the BC0 pin function works as described.

Note 3. The CS0 to CS3 (CSC) and SDRAMC pin functions share the same pin. When using the CSC and the SDRAMC simultaneously, the CS0 to CS3 pin functions are invalid.

Note 4. The WR0 signal and WR signal are identical. The WR0 signal is referred to as WR in single-write strobe mode.

14.2.3 Parallel Operation

Parallel operation is possible when different bus-master modules are requesting access to different slave interface. For example, if the CPU is fetching an instruction from Code Flash and an operand from SRAM0, the DMAC can handle transfers between a peripheral module and the external memory interface at the same time.

Figure 14.2 shows an example of parallel operations. In this example, the CPU uses CPUMAXIBI to access FHBI. Furthermore, the DMAC/DTC or EDMAC simultaneously accesses the peripheral bus or the external bus during access to FHBI and S0BI by the CPU.

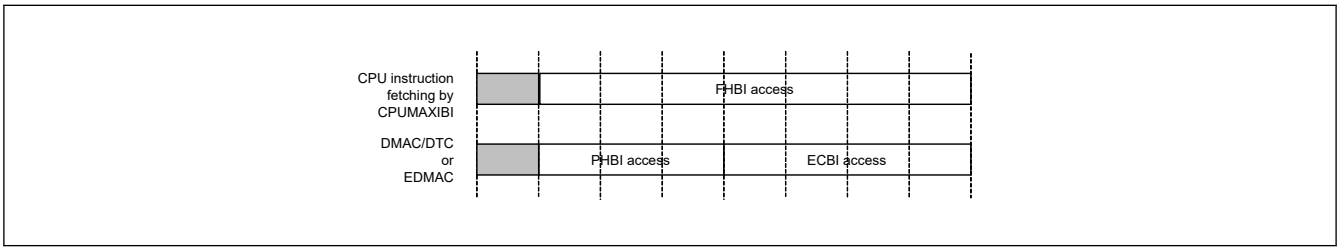


Figure 14.2 Example of parallel operations

14.2.4 Bus Settings

Set up the external bus with the following registers :

- Mode settings :
  CSn Mode Register (CSnMOD)
  CSn Wait Control Register 1 (CSnWCR1)
  CSn Wait Control Register 2 (CSnWCR2)
  CSn Control Register (CSnCR)
  CSn Recovery Cycle Register (CSnREC)
  CS Recovery Cycle Insertion Enable Register (CSRECEN)
  Bus Master Arbitration Control Register (BUSMABT)
  Bus Slave Arbitration Control Register 2 (BUSSABT2)
- I/O port assignments :
  PmnPFS.PMR = 1 and PmnPFS.PSEL[4:0] = 0x0B
- Frequency of the external bus clock (BCLK) and SDRAM clock (SDCLK) :
  SCKDIVCR register.

See section 19, I/O Ports, for information on PmnPFS and section Clock Generation Function for information on SCKDIVCR.

14.2.5 Restrictions

14.2.5.1 Endianness constraint

Memory space must be little-endian to execute code of the Arm® Cortex®-M85 core.

14.2.5.2 Bufferable write access

If an access violation occurs due to bufferable write access, an error response is not returned to the bus master, but an NMI or RESET request is issued depending on the setting of the BUSOAD register. For more information, see section 14.7.2. Operations When a Bus Error Occurs.

14.2.5.3 Access to the reserved area of FLBI

Access to the reserved area of FLBI is prohibited. Operation is not guaranteed if accessed.

14.3 Register Descriptions

14.3.1 BUSSARA : Bus Security Attribution Register A

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x100

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BUSSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BUSSA0	Bus Security Attribution A0 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

BUSSA0 bit (Bus Security Attribution A0)

Security attributes of registers for Slave Bus Control Register. The target registers are as follows:

- BUSMABT
- BUSSABT0<slave>
  <slave> = FLBI/STBYSBI/ECBI/EOBI/PBBI/PABI/PIBI/PSBI
- BUSSABT1<slave>
  <slave> = FHBI/S0BI/S1BI
- BUSDIVBYP

14.3.2 BUSSARB : Bus Security Attribution Register B

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x104

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BUSSB0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BUSSB0	Bus Security Attribution B0 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

BUSSB0 bit (Bus Security Attribution B0)

Security attributes of bus error related registers. The target registers are as follows:

- BUSnERRCLR
- MBWERRCLR
- SBWERRCLR
- BUSOAD
- BUSOADPT

14.3.3 BUSSARC : Bus Security Attribution Register C

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x110

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BUSSC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BUSSC0	Bus Security Attribution C0 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

BUSSC0 bit (Bus Security Attribution C0)

Security attributes of registers for bus (SDRAMC/CSC) Control. The target registers are as follows:

- BUS.CSnCR (n = 0 to 7)
- BUS.CSnREC (n = 0 to 7)
- BUS.CSRECEN
- BUS.CSnMOD (n = 0 to 7)
- BUS.CSnWCR1 (n = 0 to 7)
- BUS.CSnWCR2 (n = 0 to 7)
- BUS.SDCCR
- BUS.SDCMOD
- BUS.SDAMOD
- BUS.SDSELF

- BUS.SDRFCR
- BUS.SDRFEN
- BUS.SDICR
- BUS.SDIR
- BUS.SDADR
- BUS.SDTR
- BUS.SDMOD
- BUS.SDSR
- BUS.BUSPARC

14.3.4 BUSPARC : Bus Privileged Attribution Register C

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x114

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BUSPA0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	BUSPA0	External bus controller privilege attribution 0: Privileged 1: Unprivileged	R/W
31:1	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-2, P-TYPE-1

BUSPA0 bit (External bus controller privilege attribution)

Privileged attributes of registers for Bus (SDRAMC/CSC) Control. The target registers are as follows:

- BUS.CSnCR (n = 0 to 7)
- BUS.CSnREC (n = 0 to 7)
- BUS.CSRECEN
- BUS.CSnMOD (n = 0 to 7)
- BUS.CSnWCR1 (n = 0 to 7)
- BUS.CSnWCR2 (n = 0 to 7)
- BUS.SDCCR
- BUS.SDCMOD
- BUS.SDAMOD
- BUS.SDSELF
- BUS.SDRFCR
- BUS.SDRFEN
- BUS.SDICR
- BUS.SDIR

- BUS.SDADR
- BUS.SDTR
- BUS.SDMOD
- BUS.SDSR

14.3.5 CSnCR : CSn Control Register (n = 0 to 7)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x802 + 0x10 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	MPXEN	—	—	—	EMODE	—	—	BSIZE[1:0]		—	—	—	EXENB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EXENB	Operation Enable 0: Disable operation 1: Enable operation	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
5:4	BSIZE[1:0]	External Bus Width Select 0 0: 16-bit bus space 0 1: 32-bit bus space 1 0: 8-bit bus space Others: Setting prohibited	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	EMODE	Endian Mode 0: Little endian 1: Big endian	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
12	MPXEN	Address/Data Multiplexed I/O Interface Select 0: Separate bus interface is selected for area n. 1: Address/data multiplexed I/O interface is selected for area n.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Do not attempt to write the CSnCR register while the external bus is being accessed.

EXENB bit (Operation Enable)

The EXENB bit enables operation of the associated CS area. On MCU reset, operation is enabled (EXENB = 1) only for area 0. Operation in other areas is disabled (EXENB = 0). Attempts to access disabled areas have no effect.

When the CSC and SDRAMC are in use at the same time, EBCLK and SDCLK must operate at the same frequency.

BSIZE[1:0] bits (External Bus Width Select)

The BSIZE[1:0] bits specify the data bus width for the associated area.

EMODE bit (Endian Mode)

The EMODE bit specifies the endianness for the associated area. The Arm® Cortex®-M85 core is fixed at little-endian order, so instruction code can only be allocated to external spaces with little-endian specified. If an area is specified as big-endian, no instruction code can be allocated to it. Only CPU, DMAC and DTC can access to big-endian area. Memory type of big-endian area must be Device-Memory. For changing the memory type, see section 14.8. References[1]. CPU can't access to big-endian area when using M-profile Vector Extension(MVE).

MPXEN bit (Address/Data Multiplexed I/O Interface Select)

The MPXEN bit specifies separate bus interface or address/data multiplexed I/O interface of each area.

14.3.6 CSnREC : CSn Recovery Cycle Register (n = 0 to 7)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x80A + 0x10 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	WRCV[3:0]				—	—	—	—	RRCV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	RRCV[3:0]	Read Recovery 0 0 0 0: Do not insert any recovery cycles. 0 0 0 1: Insert 1 recovery cycle. 0 0 1 0: Insert 2 recovery cycles. 0 0 1 1: Insert 3 recovery cycles. 0 1 0 0: Insert 4 recovery cycles. 0 1 0 1: Insert 5 recovery cycles. 0 1 1 0: Insert 6 recovery cycles. 0 1 1 1: Insert 7 recovery cycles. 1 0 0 0: Insert 8 recovery cycles. 1 0 0 1: Insert 9 recovery cycles. 1 0 1 0: Insert 10 recovery cycles. 1 0 1 1: Insert 11 recovery cycles. 1 1 0 0: Insert 12 recovery cycles. 1 1 0 1: Insert 13 recovery cycles. 1 1 1 0: Insert 14 recovery cycles. 1 1 1 1: Insert 15 recovery cycles.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
11:8	WRCV[3:0]	Write Recovery 0 0 0 0: Do not insert any recovery cycles. 0 0 0 1: Insert 1 recovery cycle. 0 0 1 0: Insert 2 recovery cycles. 0 0 1 1: Insert 3 recovery cycles. 0 1 0 0: Insert 4 recovery cycles. 0 1 0 1: Insert 5 recovery cycles. 0 1 1 0: Insert 6 recovery cycles. 0 1 1 1: Insert 7 recovery cycles. 1 0 0 0: Insert 8 recovery cycles. 1 0 0 1: Insert 9 recovery cycles. 1 0 1 0: Insert 10 recovery cycles. 1 0 1 1: Insert 11 recovery cycles. 1 1 0 0: Insert 12 recovery cycles. 1 1 0 1: Insert 13 recovery cycles. 1 1 1 0: Insert 14 recovery cycles. 1 1 1 1: Insert 15 recovery cycles.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Do not attempt to write to the CSnREC register while the external bus is being accessed.

When the preceding bus access is from a separate bus, CSnREC is valid when the recovery cycle insertion is enabled in the Separate Bus Recovery Cycle Insertion Enable bit (RCVENi (i = 0 to 7)) in CSRECEN. When the preceding bus access is an address/data multiplexed bus access, CSnREC is valid when the recovery cycle insertion is enabled with the Multiplexed Bus Recovery Cycle Insertion Enable bit (RCVENMj (j = 0 to 7)) in CSRECEN. For more information, see section 14.5.4. Insertion of Recovery Cycles.

RRCV[3:0] bits (Read Recovery)

The RRCV[3:0] bits specify the number of recovery cycles inserted after a read access on the external bus for CSn (n = 0 to 7). When recovery cycle insertion is enabled and a value other than 0000 is set, 1 to 15 recovery cycles are inserted when:

- After a read access to the external bus, a read access is made to the external bus in the same area.
- After a read access to the external bus, a read access is made to the external bus in a different area.
- After a read access to the external bus, a write access is made to the external bus in the same area.
- After a read access to the external bus, a write access is made to the external bus in a different area.

WRCV[3:0] bits (Write Recovery)

The WRCV[3:0] bits specify the number of recovery cycles inserted after a write access on the external bus for CSn (n = 0 to 7). When recovery cycle insertion is enabled and a value other than 0000 is set, 1 to 15 recovery cycles are inserted when:

- After a write access to the external bus, a read access is made to the external bus in the same area.
- After a write access to the external bus, a read access is made to the external bus in a different area.
- After a write access to the external bus, a write access is made to the external bus in the same area.
- After a write access to the external bus, a write access is made to the external bus in a different area.

14.3.7 CSRECEN : CS Recovery Cycle Insertion Enable Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x880

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RCVENM7	RCVENM6	RCVENM5	RCVENM4	RCVENM3	RCVENM2	RCVENM1	RCVENM0	RCVEN7	RCVEN6	RCVEN5	RCVEN4	RCVEN3	RCVEN2	RCVEN1	RCVEN0
Value after reset:	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	0

Bit	Symbol	Function	R/W
0	RCVEN0	Separate Bus Recovery Cycle Insertion Enable 0 0: Disabled 1: Enabled	R/W
1	RCVEN1	Separate Bus Recovery Cycle Insertion Enable 1 0: Disabled 1: Enabled	R/W
2	RCVEN2	Separate Bus Recovery Cycle Insertion Enable 2 0: Disabled 1: Enabled	R/W
3	RCVEN3	Separate Bus Recovery Cycle Insertion Enable 3 0: Disabled 1: Enabled	R/W
4	RCVEN4	Separate Bus Recovery Cycle Insertion Enable 4 0: Disabled 1: Enabled	R/W
5	RCVEN5	Separate Bus Recovery Cycle Insertion Enable 5 0: Disabled 1: Enabled	R/W
6	RCVEN6	Separate Bus Recovery Cycle Insertion Enable 6 0: Disabled 1: Enabled	R/W
7	RCVEN7	Separate Bus Recovery Cycle Insertion Enable 7 0: Disabled 1: Enabled	R/W

Bit	Symbol	Function	R/W
8	RCVENM0	Multiplexed Bus Recovery Cycle Insertion Enable 0 0: Disabled 1: Enabled	R/W
9	RCVENM1	Multiplexed Bus Recovery Cycle Insertion Enable 1 0: Disabled 1: Enabled	R/W
10	RCVENM2	Multiplexed Bus Recovery Cycle Insertion Enable 2 0: Disabled 1: Enabled	R/W
11	RCVENM3	Multiplexed Bus Recovery Cycle Insertion Enable 3 0: Disabled 1: Enabled	R/W
12	RCVENM4	Multiplexed Bus Recovery Cycle Insertion Enable 4 0: Disabled 1: Enabled	R/W
13	RCVENM5	Multiplexed Bus Recovery Cycle Insertion Enable 5 0: Disabled 1: Enabled	R/W
14	RCVENM6	Multiplexed Bus Recovery Cycle Insertion Enable 6 0: Disabled 1: Enabled	R/W
15	RCVENM7	Multiplexed Bus Recovery Cycle Insertion Enable 7 0: Disabled 1: Enabled	R/W

Note: S-TYPE-3, P-TYPE-3

Do not attempt to write the CSRECEN register while the external bus is being accessed. For more information on insertion recovery cycles, see section 14.5.4. Insertion of Recovery Cycles.

RCVENi Bit (Separate Bus Recovery Cycle Insertion Enable i) (i = 0 to 7)

This bit enables the insertion of read or write recovery cycles when, after a read or write access on the external bus, a read or write access is made on the external bus to the same or different area.

RCVENMj Bit (Multiplexed Bus Recovery Cycle Insertion Enable j) (j = 0 to 7)

This bit enables the insertion of read or write recovery cycles when, after a read or write access on the external bus, a read or write access is made on the external bus to the same or different area.

Table 14.6 Access type associations with the RCVENn bits (1 of 2)

Access type	External Address Space	Insertion of recovery cycles	Associated bits (Separate/ Multiplexed)
Read access after read access	Same area	Recovery cycles specified in the RRCV[3:0] bits are inserted for the priority access area	RCVEN0/RCVENM0
	Different area	Recovery cycles specified in the RRCV[3:0] bits are inserted for the priority access area	RCVEN1/RCVENM1
Write access after read access	Same area	Recovery cycles specified in the RRCV[3:0] bits are inserted for the priority access area	RCVEN2/RCVENM2
	Different area	Recovery cycles specified in the RRCV[3:0] bits are inserted for the priority access area	RCVEN3/RCVENM3
Read access after write access	Same area	Recovery cycles specified in the WRCV[3:0] bits are inserted for the priority access area	RCVEN4/RCVENM4
	Different area	Recovery cycles specified in the WRCV[3:0] bits are inserted for the priority access area	RCVEN5/RCVENM5

Table 14.6 Access type associations with the RCVENn bits (2 of 2)

Access type	External Address Space	Insertion of recovery cycles	Associated bits (Separate/ Multiplexed)
Write access after write access	Same area	Recovery cycles specified in the WRCV[3:0] bits are inserted for the priority access area	RCVEN6/RCVENM6
	Different area	Recovery cycles specified in the WRCV[3:0] bits are inserted for the priority access area	RCVEN7/RCVENM7

14.3.8 CSnMOD : CSn Mode Register (n = 0 to 7)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x002 + 0x10 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PRMOD	—	—	—	—	—	PWENB	PRENB	—	—	—	—	EWENB	—	—	WRMOD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WRMOD	Write Access Mode Select 0: Byte strobe mode 1: Single-write strobe mode	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	EWENB	External Wait Enable 0: Disabled 1: Enabled	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	PRENB	Page Read Access Enable 0: Disabled 1: Enabled	R/W
9	PWENB	Page Write Access Enable 0: Disabled 1: Enabled	R/W
14:10	—	These bits are read as 0. The write value should be 0.	R/W
15	PRMOD	Page Read Access Mode Select 0: Normal access compatible mode 1: External data read continuous assertion mode	R/W

Note: S-TYPE-3, P-TYPE-3

Do not write to the CSnMOD register while access to the CSn area is in progress.

WRMOD bit (Write Access Mode Select)

The WRMOD bit selects the write access operating mode. Writing 0 selects byte strobe mode, in which data writes are controlled by the WRn signals (n = 0 to 3) associated with the respective byte positions. Writing 1 selects single-write strobe mode, in which data writes are controlled by the BCn (n = 0 to 3) and WR signals associated with the respective byte positions.

Note: Setting the external bus width to 8 bits is prohibited in single-write strobe mode.

Table 14.7 Control signals for write access mode

Mode	Pin name							
Write access mode	**WR3**	**WR2**	**WR1**	**WR0/WR**	**BC3**	**BC2**	**BC1**	**BC0**
Byte strobe mode	✓	✓	✓	✓ (WR0)	—	—	—	—
Single write strobe mode	—	—	—	✓ (WR)	✓	✓	✓	✓

✓ : Enabled, — : Disabled

EWENB bit (External Wait Enable)

The EWENB bit enables external waits. Writing 0 disables the WAIT signal. Writing 1 selects external wait and allows the WAIT signal to control the number of waits per cycle. In this state, wait cycles are inserted when the WAIT signal is low.

PRENB bit (Page Read Access Enable)

The PRENB bit enables page read accesses.

Note: When the address/data multiplexed I/O interface is selected with the CSnCR.MPXEN bit, PRENB should not be set to enable page read accesses. Page read accesses are not supported in the address/data multiplexed I/O interface.

PWENB bit (Page Write Access Enable)

The PWENB bit enables page write accesses.

Note: When the address/data multiplexed I/O interface is selected with the CSnCR.MPXEN bit, PWENB should not be set to enable page write accesses. Page write accesses are not supported in the address/data multiplexed I/O interface.

PRMOD bit (Page Read Access Mode Select)

The PRMOD bit selects the operating mode for page read accesses. Writing 0 selects normal access compatible mode, in which the RD signal is negated and an RD assert wait is inserted each time a unit of data is read. When there is no RD assert wait, the RD signal is negated only in the final transfer of the external bus access.

Writing 1 selects external data read continuous assertion mode, in which an RD assert wait is inserted and the RD signal is continuously asserted during the wait.

14.3.9 CSnWCR1 : CSn Wait Control Register 1 (n = 0 to 7)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x004 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	CSRWAIT[4:0]					—	—	—	CSWWAIT[4:0]				
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CSPRWAIT[2:0]			—	—	—	—	—	CSPWWAIT[2:0]		
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

Bit	Symbol	Function	R/W
2:0	CSPWWAIT[2:0]	Page Write Cycle Wait Select*1 0 0 0: Do not insert wait. 0 0 1: Insert wait of 1 clock cycle. 0 1 0: Insert wait of 2 clock cycles. 0 1 1: Insert wait of 3 clock cycles. 1 0 0: Insert wait of 4 clock cycles. 1 0 1: Insert wait of 5 clock cycles. 1 1 0: Insert wait of 6 clock cycles. 1 1 1: Insert wait of 7 clock cycles.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
10:8	CSPRWAIT[2:0]	Page Read Cycle Wait Select*2 0 0 0: Do not insert wait 0 0 1: Insert wait of 1 clock cycle. 0 1 0: Insert wait of 2 clock cycles. 0 1 1: Insert wait of 3 clock cycles. 1 0 0: Insert wait of 4 clock cycles. 1 0 1: Insert wait of 5 clock cycles. 1 1 0: Insert wait of 6 clock cycles. 1 1 1: Insert wait of 7 clock cycles.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
20:16	CSWWAIT[4:0]	Normal Write Cycle Wait Select 0 0 0 0 0: Do not insert wait. 0 0 0 0 1: Insert wait of 1 clock cycle. 0 0 0 1 0: Insert wait of 2 clock cycles. 0 0 0 1 1: Insert wait of 3 clock cycles. ⋮ 1 1 1 0 1: Insert wait of 29 clock cycles. 1 1 1 1 0: Insert wait of 30 clock cycles. 1 1 1 1 1: Insert wait of 31 clock cycles.	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
28:24	CSRWAIT[4:0]	Normal Read Cycle Wait Select 0 0 0 0 0: Do not insert wait. 0 0 0 0 1: Insert wait of 1 clock cycle. 0 0 0 1 0: Insert wait of 2 clock cycles. 0 0 0 1 1: Insert wait of 3 clock cycles. ⋮ 1 1 1 0 1: Insert wait of 29 clock cycles. 1 1 1 1 0: Insert wait of 30 clock cycles. 1 1 1 1 1: Insert wait of 31 clock cycles.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The CSPWWAIT[2:0] value is only valid when the CSnMOD.PWENB bit is set to 1.
Note 2. The CSPRWAIT[2:0] value is only valid when the CSnMOD.PRENB bit is set to 1.

Do not attempt to write to the CSnWCR1 register while the external bus is being accessed. Set each of these bits within a range of the restrictions described in (1) Constraints on using separate bus interface or (2) Constraints on using address/data multiplexed bus interface, according to the bus interface used.

CSPWWAIT[2:0] bits (Page Write Cycle Wait Select)

The CSPWWAIT[2:0] bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page write cycle. The setting is enabled when the CSnMOD.PWENB bit is set to 1.

Note: The settings must satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value, and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

CSPRWAIT[2:0] bits (Page Read Cycle Wait Select)

The CSPRWAIT[2:0] bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page read cycle. The setting is enabled when the CSnMOD.PRENB bit is set to 1.

Note: The settings must satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.

CSWWAIT[4:0] bits (Normal Write Cycle Wait Select)

The CSWWAIT[4:0] bits specify the number of wait cycles to be inserted into the first access during a normal write cycle or page write cycle.

Note: The settings must satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value, and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

CSRWAIT[4:0] bits (Normal Read Cycle Wait Select)

The CSRWAIT[4:0] bits specify the number of wait cycles to be inserted into the first access during a normal read cycle or page read cycle.

Note: The settings must satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

14.3.10 CSnWCR2 : CSn Wait Control Register 2 (n = 0 to 7)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x008 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	CSON[2:0]			—	WDON[2:0]			—	WRON[2:0]			—	RDON[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	AWAIT[1:0]		—	WDOFF[2:0]			—	CSWOFF[2:0]			—	CSROFF[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Function	R/W
2:0	CSROFF[2:0]	Read Access CS Extension Cycle Select 0 0 0: Do not insert wait. 0 0 1: Insert wait of 1 clock cycle. 0 1 0: Insert wait of 2 clock cycles. 0 1 1: Insert wait of 3 clock cycles. 1 0 0: Insert wait of 4 clock cycles. 1 0 1: Insert wait of 5 clock cycles. 1 1 0: Insert wait of 6 clock cycles. 1 1 1: Insert wait of 7 clock cycles.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	CSWOFF[2:0]	Write Access CS Extension Cycle Select 0 0 0: Do not insert wait. 0 0 1: Insert wait of 1 clock cycle. 0 1 0: Insert wait of 2 clock cycles. 0 1 1: Insert wait of 3 clock cycles. 1 0 0: Insert wait of 4 clock cycles. 1 0 1: Insert wait of 5 clock cycles. 1 1 0: Insert wait of 6 clock cycles. 1 1 1: Insert wait of 7 clock cycles.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	WDOFF[2:0]	Write Data Output Extension Cycle Select 0 0 0: Do not insert wait. 0 0 1: Insert wait of 1 clock cycle. 0 1 0: Insert wait of 2 clock cycles. 0 1 1: Insert wait of 3 clock cycles. 1 0 0: Insert wait of 4 clock cycles. 1 0 1: Insert wait of 5 clock cycles. 1 1 0: Insert wait of 6 clock cycles. 1 1 1: Insert wait of 7 clock cycles.	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
13:12	AWAIT[1:0]	Address Cycle Wait Select 0 0: Do not insert wait. 0 1: Insert wait of 1 clock cycle. 1 0: Insert wait of 2 clock cycles. 1 1: Insert wait of 3 clock cycles.	R/W

Bit	Symbol	Function	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
18:16	RDON[2:0]	RD Assert Wait Select 0 0 0: Do not insert wait. 0 0 1: Insert wait of 1 clock cycle. 0 1 0: Insert wait of 2 clock cycles. 0 1 1: Insert wait of 3 clock cycles. 1 0 0: Insert wait of 4 clock cycles. 1 0 1: Insert wait of 5 clock cycles. 1 1 0: Insert wait of 6 clock cycles. 1 1 1: Insert wait of 7 clock cycles.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	WRON[2:0]	WR Assert Wait Select 0 0 0: Do not insert wait. 0 0 1: Insert wait of 1 clock cycle. 0 1 0: Insert wait of 2 clock cycles. 0 1 1: Insert wait of 3 clock cycles. 1 0 0: Insert wait of 4 clock cycles. 1 0 1: Insert wait of 5 clock cycles. 1 1 0: Insert wait of 6 clock cycles. 1 1 1: Insert wait of 7 clock cycles.	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
26:24	WDON[2:0]	Write Data Output Wait Select 0 0 0: Do not insert wait. 0 0 1: Insert wait of 1 clock cycle. 0 1 0: Insert wait of 2 clock cycles. 0 1 1: Insert wait of 3 clock cycles. 1 0 0: Insert wait of 4 clock cycles. 1 0 1: Insert wait of 5 clock cycles. 1 1 0: Insert wait of 6 clock cycles. 1 1 1: Insert wait of 7 clock cycles.	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
30:28	CSON[2:0]	CS Assert Wait Select 0 0 0: Do not insert wait. 0 0 1: Insert wait of 1 clock cycle. 0 1 0: Insert wait of 2 clock cycles. 0 1 1: Insert wait of 3 clock cycles. 1 0 0: Insert wait of 4 clock cycles. 1 0 1: Insert wait of 5 clock cycles. 1 1 0: Insert wait of 6 clock cycles. 1 1 1: Insert wait of 7 clock cycles.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Do not attempt to write to the CSnWCR2 register while the external bus is being accessed. Set each of these bits within a range of the restrictions described in (1) Constraints on using separate bus interface or (2) Constraints on using address/data multiplexed bus interface, according to the bus interface used.

CSROFF[2:0] bits (Read Access CS Extension Cycle Select)

The CSROFF[2:0] bits specify the number of wait cycles to be inserted during the period from the end of a wait cycle (RD signal negated) until the CSn signal (n = 0 to 7) is negated in read access mode.

CSWOFF[2:0] bits (Write Access CS Extension Cycle Select)

The CSWOFF[2:0] bits specify the number of wait cycles to be inserted during the period from the end of a wait cycle (WRn signal (n = 0 to 3) negated) until the CSn signal (n = 0 to 7) is negated in write access mode.

Note: The settings must satisfy CSnWCR2.WDOFF[2:0] value ≤ CSnWCR2.CSWOFF[2:0] value.

WDOFF[2:0] bits (Write Data Output Extension Cycle Select)

The WDOFF[2:0] bits specify the number of wait cycles to be inserted during the period from the end of a wait cycle (WRn signal (n = 0 to 3) negated) until the write-data output is complete in write access mode.

Note: The settings must satisfy CSnWCR2.WDOFF[2:0] value ≤ CSnWCR2.CSWOFF[2:0] value.

AWAIT[1:0] bits (Address Cycle Wait Select)

AWAIT[1:0] bits specify the number of wait cycles to be inserted into an address output cycle with the address/data multiplexed I/O interface.

Note: CSnWCR2.CSON[2:0] value ≤ CSnWCR2.AWAIT[1:0] value.

For read access, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

For write access, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value and CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

RDON[2:0] bits (RD Assert Wait Select)

RDON[2:0] bits specify the number of wait cycles to be inserted before the RD signal is asserted.

Note: For normal read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

For page read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.

When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

WRON[2:0] bits (WR Assert Wait Select)

The WRON[2:0] bits specify the number of wait cycles to be inserted before the WRn signal (n = 0 to 3) is asserted.

Note: For normal write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value, and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

For page write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value, and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

WDON[2:0] bits (Write Data Output Wait Select)

The WDON[2:0] bits specify the number of wait cycles to be inserted before the write data is output.

Note: For normal write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

For page write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

CSON[2:0] bits (CS Assert Wait Select)

The CSON[2:0] bits specify the number of wait cycles to be inserted before the CSn signal (n = 0 to 7) is asserted.

Note: For normal read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

For page read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.

For normal write access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

For page write access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.AWAIT[1:0] value.

14.3.11 SDCCR : SDC Control Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	BSIZE[1:0]		—	—	—	EXENB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EXENB	Operation Enable 0: Disable 1: Enable	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
5:4	BSIZE[1:0]	SDRAM Bus Width Select 0 0: 16-bit bus space 0 1: 32-bit bus space 1 0: 8-bit bus space Others: Setting prohibited	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

EXENB bit (Operation Enable)

The EXENB bit enables the operation of the SDRAM address space. On reset, operation is disabled (EXENB = 0). Attempts to access disabled areas have no effect.

When CSC and SDRAMC are in use at the same time, EBCLK and SDCLK must operate at the same frequency.

BSIZE[1:0] bits (SDRAM Bus Width Select)

The BSIZE[1:0] bits specify the data bus width for the associated area.

14.3.12 SDCMOD : SDC Mode Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	EMODE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EMODE	Endian Mode 0: Endian order of SDRAM address space is the same as the endian order of the operating mode. 1: Endian order of SDRAM address space is not the endian order of the operating mode.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Writing to this register is possible only once after release from reset. The second and subsequent writes to this register are prohibited. SDRAM access operation is not guaranteed if more than one write access is attempted.

EMODE bit (Endian Mode)

The EMODE bit specifies the endianness for the SDRAM address space. The Arm® Cortex®-M85 core is fixed at little-endian order, so instruction code can only be allocated to external spaces with little-endian specified. If an area is specified as big-endian, no instruction code can be allocated to it.

14.3.13 SDAMOD : SDRAM Access Mode Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	BE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BE	Continuous Access Enable 0: Continuous access is disabled 1: Continuous access is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Set SDAMOD while the conditions listed in Table 14.34 are satisfied. The operation is not guaranteed if this register is set while these conditions are not satisfied.

BE bit (Continuous Access Enable)

This bit enables or disables continuous access to the SDRAM access space.

14.3.14 SDSELF : SDRAM Self-Refresh Control Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SFEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SFEN	SDRAM Self-Refresh Enable 0: Disable 1: Enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Set the SDSELF register only when the conditions in Table 14.34 are satisfied. Otherwise, operation of SDRAM I/F function is not guaranteed.

SFEN bit (SDRAM Self-Refresh Enable)

The SFEN bit controls the self-refresh operation. Setting this bit to 1 initiates an auto-refresh cycle, after which self-refresh begins. Clearing this bit to 0 ends the self-refresh, and auto-refresh resumes. When the bit is set to 1, the write value takes effect when the self-refresh operation starts. When it is cleared to 0, if the self-refresh operation has already finished, the

write value takes effect before the auto-refresh starts. If the self-refresh operation is in progress, the write value takes effect after the self-refresh is finished and before the auto-refresh is started.

14.3.15 SDRFCR : SDRAM Refresh Control Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC14

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	REFW[3:0]				RFC[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
11:0	RFC[11:0]	Auto-Refresh Request Interval Setting 0 0 0 0 0 0 0 0 0 0 0 0: Setting prohibited 0 0 0 0 0 0 0 0 0 0 0 1: 2 cycles 0 0 0 0 0 0 0 0 0 0 1 0: 3 cycles ⋮ 1 1 1 1 1 1 1 1 1 1 1 1: 4096 cycles	R/W
15:12	REFW[3:0]	Auto-Refresh Cycle/Self-Refresh Clearing Cycle Count Setting 0 0 0 0: 1 cycle 0 0 0 1: 2 cycles 0 0 1 0: 3 cycles 0 0 1 1: 4 cycles 0 1 0 0: 5 cycles 0 1 0 1: 6 cycles 0 1 1 0: 7 cycles 0 1 1 1: 8 cycles 1 0 0 0: 9 cycles 1 0 0 1: 10 cycles 1 0 1 0: 11 cycles 1 0 1 1: 12 cycles 1 1 0 0: 13 cycles 1 1 0 1: 14 cycles 1 1 1 0: 15 cycles 1 1 1 1: 16 cycles	R/W

Note: S-TYPE-3, P-TYPE-3

RFC[11:0] bits (Auto-Refresh Request Interval Setting)

The RFC[11:0] bits specify the auto-refresh request interval. They can be written to at any time, regardless of the state of the Auto-Refresh Operation Enable bit (RFEN) in SDRFEN. If auto-refresh is enabled, the write value takes effect after the end of the auto-refresh cycles. The refresh counter uses SDCLK.

REFW[3:0] bits (Auto-Refresh Cycle/Self-Refresh Clearing Cycle Count Setting)

The REFW[3:0] bits specify the number of auto-refresh cycles and the number of self-refresh clearing cycles. They can be written to at any time, regardless of the state of the Auto-Refresh Operation Enable bit (RFEN) in SDRFEN. If an auto-refresh cycle is in progress, the value written to the bits while auto-refresh is enabled takes effect after the cycle completes.

Note: Auto-refresh requests are not accepted while the SDRAM is being accessed. This means they must sometimes wait until the access completes for the auto-refresh interval to be extended. Set the RFC[11:0] bits to an auto-refresh request interval value that meets the specifications of the SDRAM being used. Additionally, make sure to set the auto-refresh request interval to a duration longer than the auto-refresh cycle. The auto-refresh interval cannot be automatically adjusted when the frequency is changed during operation. In this case, perform a self-refresh operation and set the auto-refresh interval to an appropriate value for the frequency again.

14.3.15.1 Auto-refresh Request Interval and RFC Set Value

The SDRAMC (SDRAM area controller) includes a 12-bit refresh counter that generates auto-refresh requests at fixed intervals. Use the following equation to calculate the set value for the RFC[11:0] bits from the auto-refresh request interval:

RFC = (Auto-refresh request interval / SDCLK cycle) - 1

Note: Auto-refresh requests are not accepted while the SDRAM is being accessed. They must wait until the access completes. However, the counter value is updated regardless of whether or not the request was accepted. If two or more auto-refresh requests are generated while the SDRAM is being accessed, the second and subsequent requests are ignored.

14.3.16 SDRFEN : SDRAM Auto-Refresh Control Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC16

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	RFEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFEN	Auto-Refresh Operation Enable 0: Disable 1: Enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

RFEN bit (Auto-Refresh Operation Enable)

The RFEN bit enables auto-refresh operation. When auto-refresh is required, set the RFEN bit to 1 before SDRAM access.

Clearing this bit to 0 while auto-refreshing is enabled causes RFEN to be cleared to 0 and auto-refresh operation to halt after the end of the auto-refresh cycle.The interval at which refresh requests are generated is determined by the value in the Auto-Refresh Request Interval Setting bits (RFC[11:0]) in the SDRAM Refresh Control Register (SDRFCR).

14.3.17 SDICR : SDRAM Initialization Sequence Control Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC20

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	INIRQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	INIRQ	Initialization Sequence Start 0: Invalid 1: Start initialization sequence	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Writing to this register is possible only once after release from reset. Operation is not guaranteed if more than one write access is attempted.

INIRQ bit (Initialization Sequence Start)

Setting the INIRQ bit to 1 starts the SDRAM initialization sequence and automatically sets the Initialization Status bit (INIST) in the SDRAM Status Register (SDSR) to 1. The INIST bit clears automatically after the initialization sequence ends. The value written to the INIRQ bit is not retained.

Note: Set the INIRQ bit to start the SDRAM initialization sequence only when the conditions in Table 14.34 are satisfied. Otherwise, operation is not guaranteed.

14.3.18 SDIR : SDRAM Initialization Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC24

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	PRC[2:0]			ARFC[3:0]				ARFI[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
3:0	ARFI[3:0]	Initialization Auto-Refresh Interval 0 0 0 0: 3 cycles 0 0 0 1: 4 cycles 0 0 1 0: 5 cycles 0 0 1 1: 6 cycles 0 1 0 0: 7 cycles 0 1 0 1: 8 cycles 0 1 1 0: 9 cycles 0 1 1 1: 10 cycles 1 0 0 0: 11 cycles 1 0 0 1: 12 cycles 1 0 1 0: 13 cycles 1 0 1 1: 14 cycles 1 1 0 0: 15 cycles 1 1 0 1: 16 cycles 1 1 1 0: 17 cycles 1 1 1 1: 18 cycles	R/W
7:4	ARFC[3:0]	Initialization Auto-Refresh Count 0 0 0 0: Setting prohibited 0 0 0 1: 1 time 0 0 1 0: 2 times 0 0 1 1: 3 times 0 1 0 0: 4 times 0 1 0 1: 5 times 0 1 1 0: 6 times 0 1 1 1: 7 times 1 0 0 0: 8 times 1 0 0 1: 9 times 1 0 1 0: 10 times 1 0 1 1: 11 times 1 1 0 0: 12 times 1 1 0 1: 13 times 1 1 1 0: 14 times 1 1 1 1: 15 times	R/W

Bit	Symbol	Function	R/W
10:8	PRC[2:0]	Initialization Precharge Cycle Count 0 0 0: 3 cycles 0 0 1: 4 cycles 0 1 0: 5 cycles 0 1 1: 6 cycles 1 0 0: 7 cycles 1 0 1: 8 cycles 1 1 0: 9 cycles 1 1 1: 10 cycles	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Writing to this register is possible only once after release from reset. Operation is not guaranteed if more than one write access is attempted.

ARFI[3:0] bits (Initialization Auto-Refresh Interval)

The ARFI[3:0] bits specify the interval at which the auto-refresh commands are issued in the SDRAM initialization sequence.

ARFC[3:0] bits (Initialization Auto-Refresh Count)

The ARFC[3:0] bits specify the number of times auto-refresh is to be performed in the SDRAM initialization sequence.

PRC[2:0] bits (Initialization Precharge Cycle Count)

The PRC[2:0] bits specify the number of precharged cycles in the SDRAM initialization sequence.

Note: Implement settings that satisfy the specifications of the connected SDRAM before starting the initialization sequence.

14.3.19 SDADR : SDRAM Address Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC40

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	MXC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	MXC[1:0]	Address Multiplex Select 0 0: 8-bit shift 0 1: 9-bit shift 1 0: 10-bit shift 1 1: 11-bit shift	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Set SDADR only when the conditions in Table 14.34 are satisfied. Otherwise, operation is not guaranteed.

MXC[1:0] bits (Address Multiplex Select)

The MXC[1:0] bits select the size of the shift towards the lower half of the row address in row address/column address multiplexing. For details, see Table 14.39.

14.3.20 SDTR : SDRAM Timing Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC44

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	RAI[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	RCD[1:0]		RP[2:0]			WR	—	—	—	—	—	CL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
2:0	CL[2:0]	SDRAMC Column Latency 0 0 0: Setting prohibited 0 0 1: 1 cycle 0 1 0: 2 cycles 0 1 1: 3 cycles 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	WR	Write Recovery Interval 0: 1 cycle 1: 2 cycles	R/W
11:9	RP[2:0]	Row Precharge Interval 0 0 0: 1 cycle 0 0 1: 2 cycles 0 1 0: 3 cycles 0 1 1: 4 cycles 1 0 0: 5 cycles 1 0 1: 6 cycles 1 1 0: 7 cycles 1 1 1: 8 cycles	R/W
13:12	RCD[1:0]	Row Column Latency 0 0: 1 cycle 0 1: 2 cycles 1 0: 3 cycles 1 1: 4 cycles	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
18:16	RAI[2:0]	Row Active Interval 0 0 0: 1 cycle 0 0 1: 2 cycles 0 1 0: 3 cycles 0 1 1: 4 cycles 1 0 0: 5 cycles 1 0 1: 6 cycles 1 1 0: 7 cycles 1 1 1: Setting prohibited	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The SDTR register specifies the timing for read and write accesses to the SDRAM. For more information, see section 14.6.12.3. Timing register settings and access timing.

Set the SDTR register only when the conditions in Table 14.34 are satisfied. Otherwise, operation is not guaranteed.

Writing to this register is possible only once after release from reset. Operation is not guaranteed if more than one write access is attempted

CL[2:0] bits (SDRAMC Column Latency)

The CL[2:0] bits specify the column latency of the SDRAM controller. This setting only affects the latency setting on the SDRAM controller side. To specify the column latency for externally connected SDRAM, use the SDRAM mode register (SDMOD).

WR bit (Write Recovery Interval)

The WR bit specifies the interval that must elapse between the SDRAM write command (WRIT) and deactivation (PALL).

RP[2:0] bits (Row Precharge Interval)

The RP[2:0] bits specify the minimum number of cycles that must elapse between the SDRAM deactivation command (PALL) and the next valid command.

RAI[2:0] bits (Row Active Interval)

The RAI[2:0] bits specify the minimum interval that must elapse between the SDRAM row activation command (ACTV) and deactivation (PALL). The value specified in these bits must be less than or equal to the sum of the row column latency (RCD[1:0]) and column latency (CL[2:0]) settings.

14.3.21 SDMOD : SDRAM Mode Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC48

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MR[14:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
14:0	MR[14:0]	Mode Register Setting Writing to these bits triggers a mode register set command	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The SDMOD register specifies the value to be written to the SDRAM mode register. Writing to SDMOD causes a mode register set command to be issued automatically to the SDRAM. Set SDMOD only when the conditions in Table 14.34 are satisfied. Otherwise, operation is not guaranteed.

Writing to this register is possible only once after release from reset. Operation is not guaranteed if more than one write access is attempted.

MR[14:0] bits (Mode Register Setting)

Writing to the MR[14:0] bits causes a mode register set command to be issued to the SDRAM, and the setting in the MR[14:0] bits is output to the lower bits of the address. For more information, see section 14.6.11. Setting the Mode Register.

Note:
1. Set a burst length of 1 for the SDRAM. Operation is not guaranteed with any other burst length setting.
2. The SDRAM column latency must match the setting in the SDRAMC Column Latency setting (CL[2:0]) in the SDRAM Timing Register (SDTR). Operation is not guaranteed if the latency settings do not agree.
3. Make sure the SRFST, INIST, and MRSST status bits in the SDRAM Status Register (SDSR) are all 0.

14.3.22 SDSR : SDRAM Status Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC50

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SRFST	INIST	—	—	MRSST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MRSST	Mode Register Setting Status 0: Mode register setting not in progress 1: Mode register setting in progress	R
2:1	—	These bits are read as 0.	R
3	INIST	Initialization Status 0: Initialization sequence not in progress 1: Initialization sequence in progress	R
4	SRFST	Self-Refresh Transition/Recovery Status 0: Transition/recovery not in progress 1: Transition/recovery in progress	R
7:5	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

MRSST bit (Mode Register Setting Status)

When set to 1, the MRSST bit indicates that SDRAM mode register setting is in progress.

INIST bit (Initialization Status)

When set to 1, the INIST bit indicates that the SDRAM initialization sequence is in progress.

SRFST bit (Self-Refresh Transition/Recovery Status)

When set to 1, the SRFST bit indicates that a transition to or recovery from a self-refresh operation is in progress for the SDRAM. The in progress interval begins when the bits in Table 14.8 are written to and lasts until the associated commands are issued.

Note: Execution of a self-refresh, initialization sequence, or mode register setting can only be performed when all the status bits are 0. Do not rewrite the registers and bits in Table 14.8 when any of the SRFST, INIST, or MRSST status bits is set to 1.

Table 14.8 Registers and bits requiring status bit checking

Function	Register	Bits
Self-refresh	SDSELF	SFEN
Initialization sequence	SDICR	INIRQ
Mode register setting	SDMOD	MR[14:0]

14.3.23 BUSDIVBYP : Bus Divider Bypass Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1300

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPU0 SBPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	GDSS BPE	—	—	EDMA BPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EDMABPE	Divider for EDMACBI bypass enable 0: Disable 1: Enable	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	GDSSBPE	Divider for GDSSBI bypass enable 0: Disable 1: Enable	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
16	CPU0SBPE	Divider for CPUSAHBI bypass enable 0: Disable 1: Enable	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

EDMABPE bit (Divider for EDMACBI bypass enable)

This bit enables the through transfer between master and slave by bypassing the frequency divider implemented for EDMACBI. It can reduce the waiting time of 1 PCLKA cycle. This bit can be set to 1 only when ICLK frequency is the same as PCLKA frequency. This bit cannot be switched during EDMACBI operation.

GDSSBPE bit (Divider for GDSSBI bypass enable)

This bit enables the through transfer between master and slave by bypassing the frequency divider implemented for GDSSBI. It can reduce the waiting time of 1 PCLKA cycle. This bit can be set to 1 only when ICLK frequency is the same as PCLKA frequency. This bit cannot be switched during GDSSBI operation.

CPU0SBPE bit (Divider for CPUSAHBI bypass enable)

This bit enables the through transfer between master and slave by bypassing the frequency divider implemented for CPUSAHBI. It can reduce the waiting time of 1 ICLK cycle. This bit can be set to 1 only when CPUCLK frequency is the same as ICLK frequency. This bit cannot be switched during CPUSAHBI operation.

14.3.24 BUSMABT : Bus Master Arbitration Control Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1100

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARBS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ARBS	Arbitration Select for GDSSBI 0: Fixed priority 1: Round-robin	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

This register specifies the arbitration method for GDSSBI that DRW (texture), DRW (data) and MIPI connect to.

ARBS bit (Arbitration Select for GDSSBI)

This bit specifies the priority between bus masters connected to GDSSBI . For fixed priority, see Table 14.9. For round-robin, see Table 14.10.

Table 14.9 Bus priorities with fixed-priority arbitration (ARBS = 0)

Bus Master Arbitration Control Register	Bus interface	Priority order*1
BUSMABT	GDSSBI	DRW (texture) > DRW (data) > MIPI

Note 1. “>” represents left side has high priority.

Table 14.10 Bus priorities with round-robin arbitration (ARBS = 1)

Bus Master Arbitration Control Register	Bus interface	Priority order*1
BUSMABT	GDSSBI	DRW (texture) <-> DRW (data) <-> MIPI

Note 1. “<->” represents round-robin.

14.3.25 BUSSABT0<slave> : Bus Slave Arbitration Control Register 0(<slave> = FLBI, STBYSBI, ECBI, EOBI, PBBI, PABI, PIBI, PSBI)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1210 (BUSSABT0FLBI)
0x1248 (BUSSABT0STBYSBI)
0x1250(BUSSABT0ECBI)
0x1258(BUSSABT0EOBI)
0x1260 (BUSSABT0PBBI)
0x1268 (BUSSABT0PABI)
0x1270 (BUSSABT0PIBI)
0x1278 (BUSSABT0PSBI)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARBS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ARBS	Arbitration Select for <slave> 0: Fixed priority 1: Round-robin	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

<slave> is bus interface unit name for slave.

This register specifies the arbitration method for <slave> buses.

(<slave> =FLBI/STBYSBI/PBBI/PABI/PIBI/PSBI)

ARBS bit (Arbitration Select for <slave>)

The ARBS bit specifies the priority between bus masters connected to <slave>. For fixed priority, see Table 14.11. For round-robin, see Table 14.12.

Table 14.11 Bus priorities with fixed-priority arbitration (ARBS = 0)

Bus Slave Arbitration Control Register	Slave interface	Priority order*1
BUSSABT0FLBI	FLBI	DMAC/DTC > CPU M-AXI > CPU P-AHB
BUSSABT0STBYSBI	STBYSBI	EDMAC > DMAC/DTC > CPU M-AXI
BUSSABT0ECBI	ECBI	CEU > GLCDC0 > GLCDC1 > GDSSBI > EDMAC > DMAC/DTC > CPU
BUSSABT0EOBI	EOBI	CEU > GLCDC0 > GLCDC1 > GDSSBI > EDMAC > DMAC/DTC > CPU
BUSSABT0PBBI	PBBI	DMAC/DTC > CPU P-AHB
BUSSABT0PABI	PABI	DMAC/DTC > CPU P-AHB
BUSSABT0PIBI	PIBI	DMAC/DTC > CPU P-AHB
BUSSABT0PSBI	PSBI	DMAC/DTC > CPU P-AHB

Note 1. “>” represents left side has high priority.

Table 14.12 Bus priorities with round-robin arbitration (ARBS = 1)

Bus Slave Arbitration Control Register	Slave interface	Priority order*1
BUSSABT0FLBI	FLBI	DMAC/DTC <-> CPU M-AXI <-> CPU P-AHB
BUSSABT0STBYSBI	STBYSBI	EDMAC <-> DMAC/DTC <-> CPU M-AXI
BUSSABT0ECBI	ECBI	CEU <-> GLCDC0 <-> GLCDC1 <-> GDSSBI <-> EDMAC <-> DMAC/DTC <-> CP
BUSSABT0EOBI	EOBI	CEU <-> GLCDC0 <-> GLCDC1 <-> GDSSBI <-> EDMAC <-> DMAC/DTC <-> CP
BUSSABT0PBBI	PBBI	DMAC/DTC <-> CPU P-AHB
BUSSABT0PABI	PABI	DMAC/DTC <-> CPU P-AHB
BUSSABT0PIBI	PIBI	DMAC/DTC <-> CPU P-AHB
BUSSABT0PSBI	PSBI	DMAC/DTC <-> CPU P-AHB

Note 1. “<->” represents round-robin.

14.3.26 BUSSABT1<slave> : Bus Slave Arbitration Control Register 1(<slave> = FHBI, S0BI, S1BI)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1200 (BUSSABT1FHBI)
0x1218 (BUSSABT1S0BI)
0x1220 (BUSSABT1S1BI)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARBS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ARBS[1:0]	Arbitration Select for <slave> 0 0: Fixed priority 0 1: Setting prohibited 1 0: Combination of round-robin and fixed priority 1 1: Round-robin	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

<slave> is bus interface unit name for slave.

This register specifies the arbitration method for <slave> buses.

(<slave> = FHBI/S0BI/S1BI)

ARBS[1:0] bits (Arbitration Select for <slave>)

The ARBS bits specify the priority between bus masters connected to <slave>. For priority of each setting, see Table 14.13, Table 14.14 and Table 14.15.

Table 14.13 Bus priorities with fixed-priority arbitration (ARBS[1:0] = 00) (1 of 2)

Bus Slave Arbitration Control Register	Slave interface	Priority order*1
BUSSABT1FHBI	FHBI	GLCDC (graphic 1) > GLCDC (graphic 2) > GDSSBI > DMAC/DTC > CPU M-AXI

Table 14.13 Bus priorities with fixed-priority arbitration (ARBS[1:0] = 00) (2 of 2)

Bus Slave Arbitration Control Register	Slave interface	Priority order*1
BUSSABT1S0BI	S0BI	CEU > GLCDC (graphic 1) > GLCDC (graphic 2) > GDSSBI > EDMAC > DMAC/DTC > CPU M-AXI
BUSSABT1S1BI	S1BI	CEU > GLCDC (graphic 1) > GLCDC (graphic 2) > GDSSBI > EDMAC > DMAC/DTC > CPU M-AXI

Note 1. “>” represents left side has high priority.

Table 14.14 Bus priorities with combination of round-robin and fixed priority arbitration (ARBS[1:0] = 10)

Bus Slave Arbitration Control Register	Slave interface	Priority order*1
BUSSABT1FHBI	FHBI	(GLCDC (graphic 1) <-> GLCDC (graphic 2) <-> GDSSBI) > DMAC/DTC > CPU M-AXI
BUSSABT1S0BI	S0BI	CEU > (GLCDC (graphic 1) <-> GLCDC (graphic 2) <-> GDSSBI <-> EDMAC> DMAC/DTC > CPU M-AXI
BUSSABT1S1BI	S1BI	CEU > (GLCDC (graphic 1) <-> GLCDC (graphic 2) <-> GDSSBI <-> EDMAC > DMAC/DTC > CPU M-AXI

Note 1. “<->” represents round-robin. “>” represents left side has high priority.

Table 14.15 Bus priorities with round-robin arbitration (ARBS[1:0] = 11)

Bus Slave Arbitration Control Register	Slave interface	Priority order*1
BUSSABT1FHBI	FHBI	GLCDC (graphic 1) <-> GLCDC (graphic 2) <-> GDSSBI <-> DMAC/DTC <-> CPU M-AXI
BUSSABT1S0BI	S0BI	CEU <-> GLCDC (graphic 1) <-> GLCDC (graphic 2) <-> GDSSBI <-> EDMAC) <-> DMAC/DTC <-> CPU M-AXI
BUSSABT1S1BI	S1BI	CEU <-> GLCDC (graphic 1) <-> GLCDC (graphic 2) <-> GDSSBI <-> EDMAC) <-> DMAC/DTC <-> CPU M-AXI

Note 1. “<->” represents round-robin.

14.3.27 BUSnERRADD : BUS Error Address Register (n = 4 to 9)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1800 + 0x10 × (n-1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BERAD[31:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BERAD[31:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	BERAD[31:0]	Bus Error Address When a bus error occurs, it stores an error address.	R

Note: S-TYPE-5, P-TYPE-2

Table 14.16 shows the register numbers associated with each bus type.

This register is cleared by set the BUSnERRCLR.ILERRCLR, BUSnERRCLR.MMERRCLR, or BUSnERRCLR.SLERRCLR bits or reset other than Bus error reset and Common memory error reset.

BERAD[31:0] bits (Bus Error Address)

The BERAD[31:0] bits indicates the address, when an error occurs on the associated bus. For detail of error that occurs by bus, see section 14.3.31. BUSnERRSTAT : BUS Error Status Register (n = 1 to 12) and section 14.7. Bus Error Monitoring Section.

When an error occurs on the bus, the corresponding bit of ILERRSTAT, MMERRSTAT, SLERRSTAT in BUSnERRSTAT (n = 1 to 12) is set to 1, at the same time, the BERAD[31:0] bits stores the address of the bus error access.

When the error address is set once, it is not updated until the value is cleared by the BUSnERRCLR Register.

When all of ILERRSTAT, MMERRSTAT, SLERRSTAT, MSERRSTAT in BUSnERRSTAT (n =1 to 12) are set to 0, the BERAD[31:0] bits are indefinite.

Table 14.16 Associations between bus types and registers

Bus type	Bus error status register	Bus error clear register	Bus error address register	Bus error RW register
CPUMAXIBI_R channel	BUS1ERRSTAT	BUS1ERRCLR	—	—
CPUMAXIBI_W channel	BUS2ERRSTAT	BUS2ERRCLR	—	—
CPUPAHBI	BUS3ERRSTAT	BUS3ERRCLR	—	—
DMAC/DTCBI	BUS4ERRSTAT	BUS4ERRCLR	BUS4ERRADD BMSA4ERRADD	BUS4ERRRW BMSA4ERRRW
EDMACBI	BUS5ERRSTAT	BUS5ERRCLR	BUS5ERRADD BMSA5ERRADD	BUS5ERRRW BMSA5ERRRW
GLCDC0BI	BUS6ERRSTAT	BUS6ERRCLR	BUS6ERRADD BMSA6ERRADD	BUS6ERRRW BMSA6ERRRW
GLCDC1BI	BUS7ERRSTAT	BUS7ERRCLR	BUS7ERRADD BMSA7ERRADD	BUS7ERRRW BMSA7ERRRW
DRW0BI	BUS8ERRSTAT	BUS8ERRCLR	BUS8ERRADD BMSA8ERRADD	BUS8ERRRW BMSA8ERRRW
DRW1BI	BUS9ERRSTAT	BUS9ERRCLR	BUS9ERRADD BMSA9ERRADD	BUS9ERRRW BMSA9ERRRW
CEUBI_W channel	BUS10ERRSTAT	BUS10ERRCLR	—	—
MIPIBI_R channel	BUS11ERRSTAT	BUS11ERRCLR	—	—
MIPIBI_W channel	BUS12ERRSTAT	BUS12ERRCLR	—	—

14.3.28 BUSnERRRW : BUS Error Read Write (n = 4 to 9)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1804 + 0x10 × (n-1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	RWSTAT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RWSTAT	Error access Read/Write Status The status at the time of the error. 0: Read access 1: Write access	R
7:1	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-2

This register is cleared by a reset other than the Bus error reset and Common memory error reset, and the BUSnERRCLR.ILERRCLR, BUSnERRCLR.MMERRCLR and BUSnERRCLR.SLERRCLR.

Table 14.16 shows the register numbers associated with each bus type.

RWSTAT bit (Error access Read/Write Status)

The RWSTAT bit indicates the access status, write access or read access, when an error occurs on the associated bus.

For detail of error that occurs by bus, see section 14.3.31. BUSnERRSTAT : BUS Error Status Register (n = 1 to 12) and section 14.7. Bus Error Monitoring Section.

When an error occurs on the bus, the corresponding bit of ILERRSTAT, MMERRSTAT, SLERRSTAT, in BUSnERRSTAT (n = 1 to 12) is set to 1, at the same time, the RWSTAT bits stores the read/write status of the bus error access.

When all of ILERRSTAT, MMERRSTAT, SLERRSTAT in BUSnERRSTAT (n = 1 to 12) are set to 0, the RWSTAT bit is indefinite.

When the RWSTAT bit is set once, it is not updated until the value is cleared by the BUSnERRCLR Register.

14.3.29 BMSAnERRADD : Bus Master Security Attribution Unit Error Address (n = 4 to 9)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1900 + 0x10 × (n-1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSERAD[31:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MSERAD[31:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	MSERAD[31:0]	Bus Master Security Attribution Unit Error Address When a Master Security Attribution Unit error occurs, It stores an error address.	R

Note: S-TYPE-5, P-TYPE-2
Note: This register is cleared by set the BUSnERRCLR.MSERRCLR bit or reset other than Bus error reset and Common memory error reset.

Table 14.16 lists the registers associated with each bus type.

MSERAD[31:0] bits (Bus Master Security Attribution Unit Error Address)

The MSERAD[31:0] bits indicates the address, when an error occurs on the associated bus.

For detail of error that occurs by bus, see section 14.3.31. BUSnERRSTAT : BUS Error Status Register (n = 1 to 12) and section 14.7. Bus Error Monitoring Section.

When an error occurs on the bus, the corresponding bit of MSERRSTAT in BUSnERRSTAT (n = 1 to 12) is set to 1, at the same time, the MSERAD[31:0] bits stores the address of the bus error access.

When the error address is set once, it is not updated until the value is cleared by the BUSnERRCLR (n = 1 to 12).

When all of MSERRSTAT in BUSnERRSTAT (n = 1 to 12) is set to 0, the MSERAD[31:0] bits are indefinite.

14.3.30 BMSAnERRRW : BUS Master Security Attribution Unit Error Read Write (n = 4 to 9)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1904 + 0x10 × (n-1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MSARWSTAT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MSARWSTAT	Master Security Attribution Unit error access Read/Write Status The status at the time of the error. 0: Read access 1: Write access	R
7:1	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-2
Note: This register is cleared by set the BUSnERRCLR.MSERRCLR bits or reset other than Bus error reset and Common memory error reset.

Table 14.16 shows the register numbers associated with each bus type.

MSARWSTAT bit (Master Security Attribution Unit error access Read/Write Status)

The MSARWSTAT bit indicates the access status, write access or read access, when an error occurs on the associated bus.

For detail of error that occurs by bus, see section 14.3.31. BUSnERRSTAT : BUS Error Status Register (n = 1 to 12) and section 14.7. Bus Error Monitoring Section.

When an error occurs on the bus, the corresponding bit of MSERRSTAT in BUSnERRSTAT (n = 1 to 12) is set to 1, at the same time, the MSARWSTAT bits stores the read/write status of the bus error access.

When the MSARWSTAT bit is set once, it is not updated until the value is cleared by the BUSnERRCLR (n = 1 to 12).

When all of MSERRSTAT in BUSnERRSTAT (n = 1 to 12) are set to 0, the MSARWSTAT bit is indefinite.

14.3.31 BUSnERRSTAT : BUS Error Status Register (n = 1 to 12)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1A00 + 0x10 × (n - 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	MSERRSTAT	ILERRSTAT	MMERRSTAT	—	—	SLERRSTAT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SLERRSTAT	Slave Bus Error Status 0: No error occurred 1: Error occurred	R
2:1	—	These bits are read as 0.	R
3	MMERRSTAT	Master MPU Error Status 0: No error occurred 1: Error occurred	R
4	ILERRSTAT	Illegal Address Access Error Status 0: No error occurred 1: Error occurred	R

Bit	Symbol	Function	R/W
5	MSERRSTAT	Master Security Attribution Unit Error Status 0: No error occurred 1: Error occurred	R
7:6	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-2

This register is cleared by reset other than Bus error reset and Common memory error reset.

Table 14.16 shows the register numbers associated with each bus type.

When the bus error occurred to plural at the same time, the STAT bit becomes effective only 1 by the following order of priority.The left side is high priority.

Master Security attribution unit > Master-MPU Error > Illegal access Error, Slave TrustZone filter, Slave bus Error

Illegal access Error, Slave TrustZone filter Error and Slave bus Error don’t occur at the same time.

After one of MSERRSTAT, ILERRSTAT, MMERRSTAT, and SLERRSTAT are set once, MSERRSTAT, ILERRSTAT, MMERRSTAT, and SLERRSTAT are not renewed.

SLERRSTAT bit (Slave Bus Error Status)

When slave error or TrustZone filter error occurs by bus, BUSnERRSTAT.SLERRSTAT (n = 1 to 12) is set to 1.

There is clearing conditions.

- Set BUSnERRCLR.SLERRCLR to 1.

Slave error is an error that occurs on a slave such as Time-out.

The SLERRSTAT bit is not set when debugger violate.

For detail of slave error that occurs by bus, see section 14.7. Bus Error Monitoring Section.

MMERRSTAT bit (Master MPU Error Status)

When master mpu error occurs by bus, BUSnERRSTAT. MMERRSTAT (n = 1 to 12) is set to 1.

There is clearing conditions.

- Set BUSnERRCLR.MMERRCLR to 1.

For detail of master mpu error that occurs by bus, see section 15, Memory Protection Unit (MPU).

ILERRSTAT bit (Illegal Address Access Error Status)

When illegal address access error occurs by bus and write access to FHBI occurs, BUSnERRSTAT.ILERRSTAT (n = 1 to 12) is set to 1.

There is clearing conditions.

- Set BUSnERRCLR.ILLERRCLR to 1.

The ILERRSTAT bit is not set when debugger violate.

For detail of illegal address access error that occurs by bus, see section 14.7. Bus Error Monitoring Section.

MSERRSTAT bit (Master Security Attribution Unit Error Status)

When MSAU error occurs by bus, BUSnERRSTAT.MSERRSTAT (n = 1 to 12) is set to 1.

There is clearing conditions.

- Set BUSnERRCLR.MSERRCLR to 1.

For detail of MSAU error that occurs by bus, see section 14.7. Bus Error Monitoring Section.

14.3.32 BUSnERRCLR : BUS Error Clear Register (n = 1 to 12)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1A08 + 0x10 × (n - 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	MSERRCLR	ILERRCLR	MMERRCLR	—	—	SLERRCLR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SLERRCLR	Slave Bus Error Clear Writing 1 to this bit clears the BUSnERRSTAT.SLERRSTAT, BUSnERRADD.BERAD and BUSnERRRW.RWSTAT.	R/W*1
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	MMERRCLR	Master MPU Error Clear Writing 1 to this bit clears the BUSnERRSTAT.MMERRSTAT, BUSnERRADD.BERAD and BUSnERRRW.RWSTAT .	R/W*1
4	ILERRCLR	Illegal Address Access Error Clear Writing 1 to this bit clears the BUSnERRSTAT.ILERRSTAT, BUSnERRADD.BERAD and BUSnERRRW.RWSTAT.	R/W*1
5	MSERRCLR	Master Security Attribution Unit Error Clear Writing 1 to this bit clears the BUSnERRSTAT.MSERRSTAT, BUSnERRADD.BERAD and BUSnERRRW.RWSTAT.	R/W*1
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note 1. Only 1 can be written to this bit. This bit is read as 0. Writing 0 to this bit has no effect.

Table 14.16 shows the register numbers associated with each bus type.

When writing 1 to BUSnERRCLR (n = 1 to 12), stop the bus access that causes an error in the corresponding bus master.

SLERRCLR bit (Slave Bus Error Clear)

Writing 1 to the SLERRCLR bit clears the BUSnERRSTAT.SLERRSTAT, BMSAnERRADD.MSERAD and BMSAnERRRW.MSARWSTAT (n = 1 to 12).

This bit is read as 0.

MMERRCLR bit (Master MPU Error Clear)

Writing 1 to the MMERRCLR bit clears the BUSnERRSTAT.MMERRSTAT, BUSnERRADD.BERAD and BUSnERRRW.RWSTAT (n = 1 to 12).

This bit is read as 0.

ILERRCLR bit (Illegal Address Access Error Clear)

Writing 1 to the MSERRCLR bit clears the BUSnERRSTAT.MSERRSTAT, BMSAnERRADD.MSERAD and BMSAnERRRW.MSARWSTAT (n = 4 to 9).

This bit is read as 0.

MSERRCLR bit (Master Security Attribution Unit Error Clear)

Writing 1 to the MSERRCLR bit clears the BUSnERRSTAT.MSERRSTAT, BUSnERRADD.BERAD and BUSnERRRW.RWSTAT (n = 1 to 12).

This bit is read as 0.

14.3.33 BUSOAD : BUS Operation After Detection Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1000

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	BWERROAD	SLERROAD	ILERROAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ILERROAD	Illegal address access error operation after detection CPU 0: Only error response is returned. 1: Reset Other master 0: NMI 1: Reset	R/W
1	SLERROAD	Slave bus error operation after detection CPU 0: Only error response is returned. 1: Reset Other master 0: NMI 1: Reset	R/W
2	BWERROAD	Bufferable write error operation after detection All bus master 0: NMI 1: Reset	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: When an error occurs in a speculative access issued by CPU with OAD bit is set to 1, a system reset request may occur. In that case, it can be avoided by using Arm-MPU. See section 14.8. References[3] for the detail.

ILERROAD bit (Illegal address access error operation after detection)

The OAD bit specify operation when the illegal address access error is detected. When OAD bit is 0, the operation differs depending on the bus master. If CPU causes access violation, only error response is returned and NMI is not generated. If master other than CPU causes access violation, error response is returned and NMI is generated.

When OAD bit is 1, reset request is generated.

SLERROAD bit (Slave bus error operation after detection)

The OAD bit specify operation when the slave bus error is detected. When OAD bit is 0, the operation differs depending on the bus master. If CPU causes access violation, only error response is returned and NMI is not generated. If master other than CPU causes access violation, error response is returned and NMI is generated.

When OAD bit is 1, reset request is generated.

BWERROAD bit (Bufferable write error operation after detection)

The OAD bit specify operation when the bufferable write error is detected.

When OAD bit is 0, error response is returned and NMI is generated.

When OAD bit is 1, reset request is generated.

14.3.34 BUSOADPT : BUS Operation After Detection Protect Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1004

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: BUSOAD register writing is possible. 1: BUSOAD register writing is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key code These bits enable or disable writes to the PROTECT bit.	W

Note: S-TYPE-3, P-TYPE-2

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writes to the associated registers to be protected.

BUSOADPT.PROTECT controls the following registers:

- BUSOAD

When the PROTECT bit is set simultaneously, write 0xA5 to the KEY[7:0] bits using half word access.

KEY[7:0] bits (Key code)

The KEY[7:0] bits enable or disable writes to the PROTECT bit. When writing to the PROTECT bit simultaneously, write 0xA5 to the KEY[7:0] bits. When other values are written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

14.3.35 MBWERRSTAT : Master Bufferable Write Error Status Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1B00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	MBWERR23	MBWERR22	MBWERR21	MBWERR20	MBWERR19	MBWERR18	MBWERR17	MBWERR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MBWERR8	—	—	—	—	—	—	MBWERR1	MBWERR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBWERR0	Master Bufferable Write Error 0: No bufferable write error in Master #0 1: Bufferable write error occurs in Master #0	R
1	MBWERR1	Master Bufferable Write Error 0: No bufferable write error in Master #1 1: Bufferable write error occurs in Master #1	R
7:2	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
8	MBWERR8	Master Bufferable Write Error 0: No bufferable write error in Master #8 1: Bufferable write error occurs in Master #8	R
15:9	—	These bits are read as 0.	R
16	MBWERR16	Master Bufferable Write Error 0: No bufferable write error in Master #16 1: Bufferable write error occurs in Master #16	R
17	MBWERR17	Master Bufferable Write Error 0: No bufferable write error in Master #17 1: Bufferable write error occurs in Master #17	R
18	MBWERR18	Master Bufferable Write Error 0: No bufferable write error in Master #18 1: Bufferable write error occurs in Master #18	R
19	MBWERR19	Master Bufferable Write Error 0: No bufferable write error in Master #19 1: Bufferable write error occurs in Master #19	R
20	MBWERR20	Master Bufferable Write Error 0: No bufferable write error in Master #20 1: Bufferable write error occurs in Master #20	R
21	MBWERR21	Master Bufferable Write Error 0: No bufferable write error in Master #21 1: Bufferable write error occurs in Master #21	R
22	MBWERR22	Master Bufferable Write Error 0: No bufferable write error in Master #22 1: Bufferable write error occurs in Master #22	R
23	MBWERR23	Master Bufferable Write Error 0: No bufferable write error in Master #23 1: Bufferable write error occurs in Master #23	R
31:24	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-2
Note: This register is cleared by reset other than Bus error reset and Common memory error reset.

MBWERRn bit (Master Bufferable Write Error Clear n)

When bufferable write access error occurs by MBWERRSTAT.MBWERRn is set to 1.

There are clearing conditions.

- Set MBWERRCLR.MBWERRCn to 1

Table 14.17 lists the registers associated with each bus master type.

Table 14.17 Associations between bus master types and bits (1 of 2)

Bus master type	Master bufferable write error status bit	Master bufferable write error clear bit
CPUMAXIBI_W channel	MBWERR0	MBWECLR0
CPUPAHBI	MBWERR1	MBWECLR1
DMAC/DTCBI	MBWERR8	MBWECLR8
EDMACBI	MBWERR16	MBWECLR16
GLCDC0BI	MBWERR17	MBWECLR17
GLCDC1BI	MBWERR18	MBWECLR18
DRW0BI	MBWERR19	MBWECLR19
DRW1BI	MBWERR20	MBWECLR20
CEUBI_W channel	MBWERR21	MBWECLR21
MIPIBI_W channel	MBWERR22	MBWECLR22

Table 14.17 Associations between bus master types and bits (2 of 2)

Bus master type	Master bufferable write error status bit	Master bufferable write error clear bit
GDSSBI	MBWERR23	MBWECLR23

Note: GDSSBI is a BUS interface that combines DRW0BI, DRW1BI and MIPIBI. See Table 14.2 for details.

14.3.36 MBWERRCLR : Master Bufferable Write Error Clear Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1B08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	MBWECLR23	MBWECLR22	MBWECLR21	MBWECLR20	MBWECLR19	MBWECLR18	MBWECLR17	MBWECLR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MBWECLR8	—	—	—	—	—	—	MBWECLR1	MBWECLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBWECLR0	Master Bufferable Write Error Clear Writing 1 to this bit clears the MBWERRSTAT.MBWERR0 flag.	R/W*1
1	MBWECLR1	Master Bufferable Write Error Clear Writing 1 to this bit clears the MBWERRSTAT.MBWERR1 flag.	R/W*1
7:2	—	These bits are read as 0.	R/W*1
8	MBWECLR8	Master Bufferable Write Error Clear Writing 1 to this bit clears the MBWERRSTAT.MBWERR8 flag.	R/W*1
15:9	—	These bits are read as 0.	R/W*1
16	MBWECLR16	Master Bufferable Write Error Clear Writing 1 to this bit clears the MBWERRSTAT.MBWERR16 flag.	R/W*1
17	MBWECLR17	Master Bufferable Write Error Clear Writing 1 to this bit clears the MBWERRSTAT.MBWERR17 flag.	R/W*1
18	MBWECLR18	Master Bufferable Write Error Clear Writing 1 to this bit clears the MBWERRSTAT.MBWERR18 flag.	R/W*1
19	MBWECLR19	Master Bufferable Write Error Clear Writing 1 to this bit clears the MBWERRSTAT.MBWERR19 flag.	R/W*1
20	MBWECLR20	Master Bufferable Write Error Clear Writing 1 to this bit clears the MBWERRSTAT.MBWERR20 flag.	R/W*1
21	MBWECLR21	Master Bufferable Write Error Clear Writing 1 to this bit clears the MBWERRSTAT.MBWERR21 flag.	R/W*1
22	MBWECLR22	Master Bufferable Write Error Clear Writing 1 to this bit clears the MBWERRSTAT.MBWERR22 flag.	R/W*1
23	MBWECLR23	Master Bufferable Write Error Clear Writing 1 to this bit clears the MBWERRSTAT.MBWERR23 flag.	R/W*1
31:24	—	These bits are read as 0.	R/W*1

Note: S-TYPE-3, P-TYPE-2
Note 1. Only 1 can be written to this bit.

MBWECLRn bit (Master Bufferable Write Error Clear n)

Writing 1 to this bit clears the MBWERRSTAT. MBWERRn flag.

This bit is read as 0.

14.3.37 SBWERRSTAT : Slave Bufferable Write Error Status Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1B20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SBWERR12	SBWERR11	SBWERR10	SBWERR9	SBWERR8	SBWERR7	SBWERR6	SBWERR5	SBWERR4	SBWERR3	SBWERR2	SBWERR1	SBWERR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SBWERR0	Slave Bufferable Write Error 0: No bufferable write error in Slave #0 1: Bufferable write error occurs in Slave #0	R
1	SBWERR1	Slave Bufferable Write Error 0: No bufferable write error in Slave #1 1: Bufferable write error occurs in Slave #1	R
2	SBWERR2	Slave Bufferable Write Error 0: No bufferable write error in Slave #2 1: Bufferable write error occurs in Slave #2	R
3	SBWERR3	Slave Bufferable Write Error 0: No bufferable write error in Slave #3 1: Bufferable write error occurs in Slave #3	R
4	SBWERR4	Slave Bufferable Write Error 0: No bufferable write error in Slave #4 1: Bufferable write error occurs in Slave #4	R
5	SBWERR5	Slave Bufferable Write Error 0: No bufferable write error in Slave #5 1: Bufferable write error occurs in Slave #5	R
6	SBWERR6	Slave Bufferable Write Error 0: No bufferable write error in Slave #6 1: Bufferable write error occurs in Slave #6	R
7	SBWERR7	Slave Bufferable Write Error 0: No bufferable write error in Slave #7 1: Bufferable write error occurs in Slave #7	R
8	SBWERR8	Slave Bufferable Write Error 0: No bufferable write error in Slave #8 1: Bufferable write error occurs in Slave #8	R
9	SBWERR9	Slave Bufferable Write Error 0: No bufferable write error in Slave #9 1: Bufferable write error occurs in Slave #9	R
10	SBWERR10	Slave Bufferable Write Error 0: No bufferable write error in Slave #10 1: Bufferable write error occurs in Slave #10	R
11	SBWERR11	Slave Bufferable Write Error 0: No bufferable write error in Slave #11 1: Bufferable write error occurs in Slave #11	R
12	SBWERR12	Slave Bufferable Write Error 0: No bufferable write error in Slave #12 1: Bufferable write error occurs in Slave #12	R
31:13	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-2
Note: This register is cleared by reset other than Bus error reset and Common memory error reset.

SBWERRn bit (Slave Bufferable Write Error n)

When bufferable write access error occurs by SBWERRSTAT.SBWERRn is set to 1.

There are clearing conditions.

- Set SBWERRCLR.SBWECLRn to 1.

Table 14.18 lists the registers associated with each bus slave type.

Table 14.18 Associations between bus slave types and bits

Bus slave type	Slave bufferable write error status bit	Slave bufferable write error clear bit
FHBI	SBWERR0	SBWECLR0
FLBI	SBWERR1	SBWECLR1
CPUSAHBI	SBWERR2	SBWECLR2
S0BI	SBWERR3	SBWECLR3
S1BI	SBWERR4	SBWECLR4
STBYSBI	SBWERR5	SBWECLR5
ECBI	SBWERR6	SBWECLR6
EOBI	SBWERR7	SBWECLR7
PBBI	SBWERR8	SBWECLR8
PABI	SBWERR9	SBWECLR9
PIBI	SBWERR10	SBWECLR10
ICUBI	SBWERR11	SBWECLR11
PSBI	SBWERR12	SBWECLR12

14.3.38 SBWERRCLR : Slave Bufferable Write Error Clear Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1B28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SBWECLR12	SBWECLR11	SBWECLR10	SBWECLR9	SBWECLR8	SBWECLR7	SBWECLR6	SBWECLR5	SBWECLR4	SBWECLR3	SBWECLR2	SBWECLR1	SBWECLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SBWECLR0	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR0 flag.	R/W*1
1	SBWECLR1	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR1 flag.	R/W*1
2	SBWECLR2	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR2 flag.	R/W*1
3	SBWECLR3	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR3 flag.	R/W*1

Bit	Symbol	Function	R/W
4	SBWECLR4	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR4 flag.	R/W*1
5	SBWECLR5	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR5 flag.	R/W*1
6	SBWECLR6	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR6 flag.	R/W*1
7	SBWECLR7	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR7 flag.	R/W*1
8	SBWECLR8	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR8 flag.	R/W*1
9	SBWECLR9	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR9 flag.	R/W*1
10	SBWECLR10	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR10 flag.	R/W*1
11	SBWECLR11	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR11 flag.	R/W*1
12	SBWECLR12	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR12 flag.	R/W*1
31:13	—	These bits are read as 0.	R/W*1

Note: S-TYPE-3, P-TYPE-2
Note 1. Only 1 can be written to this bit.

SBWECLRn bit (Slave Bufferable Write Error Clear n)

Writing 1 to this bit clears the SBWERRSTAT.SBWERRn flag.

This bit is read as 0.

14.4 Endian and Data Alignment

The external bus has a data alignment function to control which byte of the data bus (D31 to D24, D23 to D16, D15 to D08, D07 to D00) is used when accessing the external address space (the CS and SDRAM areas). Alignment is based on the bus specifications of the area to be accessed (8-bit, 16-bit, or 32-bit bus space), the data size, and the endian order.

14.4.1 Data Alignment Control for the CS Area

14.4.1.1 32-Bit Bus Space

When a 32-bit width is selected for a bus space by the BSIZE[1:0] bits in CSnCR, the address buses A23 to A2 are enabled to output address signals in units of 32 bits, and the address buses A1 and A0 are disabled (always output low).

When byte strobe mode is selected (the WRMOD bit = 0 in CSnMOD), the WR0 to WR3 pins are enabled. The BC0 to BC3 pins are not used.

When single-write strobe mode is selected (the WRMOD bit = 1 in CSnMOD), only the WR0 pin is enabled and always outputs the low level during write access, regardless of the data size. The WR1 to WR3 pins are invalid (always output high). The valid byte position is indicated by the BC0 to BC3 pins.

In 32-bit bus space, the valid positions of data external to the MCU and of control signals differ according to whether the endian is big or little.

Table 14.19 Data alignment (little-endian) in 32-bit bus space

Data size	Access address	Number of access	Bus cycle	Unit of data	Address	WR3/BC3 RD Data bus D31 D24	WR2/BC2 RD Data bus D23 D16	WR1/BC1 RD Data bus D15 D8	WR0/BC0 RD Data bus D7 D0
8 bits	4n	One	First	8 bits	4n				7 0
	4n + 1	One			4n			7 0	
	4n + 2	One	First	8 bits	4n		7 0		
	4n + 3	One			4n	7 0			
16 bits	4n	One	First	16 bits	4n			15 8	7 0
	4n + 2	One	First	16 bits	4n	15 8	7 0		
32 bits	4n	One	First	32 bits	4n	31 24	23 16	15 8	7 0

Table 14.20 Data alignment (big-endian) in 32-bit bus space

Data size	Access address	Number of access	Bus cycle	Unit of data	Address	WR3/BC3 RD Data bus D31 D24	WR2/BC2 RD Data bus D23 D16	WR1/BC1 RD Data bus D15 D8	WR0/BC0 RD Data bus D7 D0
8 bits	4n	One	First	8 bits	4n	7 0			
	4n + 1	One			4n		7 0		
	4n + 2	One	First	8 bits	4n			7 0	
	4n + 3	One			4n				7 0
16 bits	4n	One	First	16 bits	4n	15 8	7 0		
	4n + 2	One	First	16 bits	4n			15 8	7 0
32 bits	4n	One	First	32 bits	4n	31 24	23 16	15 8	7 0

14.4.1.2 16-Bit Bus Space

When a 16-bit bus space is selected in the BSIZE[1:0] bits in CSnCR, address buses A23 to A01 are enabled to output address signals in 16-bit units, and the address bus A00 is disabled (always outputs low).

When byte strobe mode is selected (WRMOD = 0 in CSnMOD), the WR0 and WR1 pins are enabled, and the WR2 and WR3 pins are disabled (fixed high). The BC0 to BC3 pins are not used.

When single-write strobe mode is selected (WRMOD = 1 in CSnMOD), only the WR0 pin is enabled, and it always outputs low during write access, regardless of the data size. The WR1 to WR3 pins are invalid (always outputs high). The valid byte position is indicated by the BC0 and BC1 pins.

The valid positions of control signals and data external to the MCU differ depending on the endian order. See Table 14.21 and Table 14.22.

Page access can occur for accesses to data in 32-bit units. Page access can only occur when an access does not extend over a 32-bit boundary and causes no change in the BC0 and BC1 signals. The conditions in which page access occurs are indicated by the letter (p) in Table 14.21 and Table 14.22.

Table 14.21 Data alignment (little-endian) in 16-bit bus space

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address		WR1/BC1 RD Data bus D15 D08	WR0/BC0 RD Data bus D07 D00
8 bits	4n	One	First	8 bits	4n			7 0
	4n + 1	One	First	8 bits	4n		7 0	
	4n + 2	One	First	8 bits	4n + 2			7 0
	4n + 3	One	First	8 bits	4n + 2		7 0	
16 bits	4n	One	First	16 bits	4n		15 8	7 0
	4n+2	One	First	16 bits	4n + 2		15 8	7 0
32 bits	4n	Two	First	16 bits	4n		15 8	7 0
			Second	16 bits	4n + 2	(p)	31 24	23 16

(p):Page access (only when page access is enabled in the PRENB and PWENB bits in CSnMOD)

Table 14.22 Data alignment (big-endian) in 16-bit bus space (1 of 2)

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address	WR1/BC1 RD Data bus D15 D08	WR0/BC0 RD Data bus D07 D00
8 bits	4n	One	First	8 bits	4n	7 0	
	4n + 1	One	First	8 bits	4n		7 0
	4n + 2	One	First	8 bits	4n + 2	7 0	
	4n + 3	One	First	8 bits	4n + 2		7 0

Table 14.22 Data alignment (big-endian) in 16-bit bus space (2 of 2)

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address		WR1/BC1 RD Data bus D15 D08	WR0/BC0 RD Data bus D07 D00
16 bits	4n	One	First	16 bits	4n		15 8	7 0
	4n+2	One	First	16 bits	4n + 2		15 8	7 0
32 bits	4n	Two	First	16 bits	4n		31 24	23 16
			Second	16 bits	4n + 2	(p)	15 8	7 0

(p):Page access (only when page access is enabled in the PRENB and PWENB bits in CSnMOD)

14.4.1.3 8-Bit Bus Space

When an 8-bit bus space is selected by the BSIZE[1:0] bits in CSnCR, the address buses A23 to A0 are enabled to output address signals in byte units.

In 8-bit bus space, only the WR0# pin is valid regardless of write access mode, and always outputs the low level during write access. The WR1# to WR3# pins and the BC0# to BC3# pins are not used.

Page access can occur in access to data in 16- or 32-bit units. Specifically, page access can occur when an access does not spread over a 32-bit boundary. The conditions in which page access occurs are indicated by the letter (p) in Table 14.23 and Table 14.24.

In 8-bit bus space, the valid positions of data external to the MCU are D7 to D0 and WR0# is used as the control signal, regardless of the endian mode.

When an 8-bit bus space is selected in the BSIZE[1:0] bits in CSnCR, the address buses A23 to A00 are enabled to output address signals in byte units.

In 8-bit bus space, only the WR0 pin is valid, regardless of the write access mode, and it always outputs low during write access. The WR1 to WR3 pins and the BC0 to BC3 pins are not used.

The valid positions of data external to the MCU are D07 to D00, and WR0 is used as the control signal, regardless of the endian mode. See Table 14.23 and Table 14.24.

Page access can occur for accesses to data in 16-bit or 32-bit units. Page access can only occur when an access does not extend over a 32-bit boundary. The conditions in which page access occurs are indicated by the letter (p) in Table 14.23 and Table 14.24.

Table 14.23 Data alignment (little-endian) in 8-bit bus space

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address		WR1/BC1 RD Data bus D15 D08	WR0/BC0 RD Data bus D07 D00
8 bits	4n	One	First	8 bits	4n			7 0
	4n + 1	One	First	8 bits	4n + 1			7 0
	4n + 2	One	First	8 bits	4n + 2			7 0
	4n + 3	One	First	8 bits	4n + 3			7 0
16 bits	4n	Two	First	8 bits	4n			7 0
			Second	8 bits	4n + 1	(p)		15 8
	4n + 2	Two	First	8 bits	4n + 2			7 0
			Second	8 bits	4n + 3	(p)		15 8
32 bits	4n	Four	First	8 bits	4n			7 0
			Second	8 bits	4n + 1	(p)		15 8
			Third	8 bits	4n + 2	(p)		23 16
			Fourth	8 bits	4n + 3	(p)		31 24

(p):Page access (only when page access is enabled in the PRENB and PWENB bits in CSnMOD)

Table 14.24 Data alignment (big-endian) in 8-bit bus space (1 of 2)

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address	WR1/BC1 RD Data bus D15 D08	WR0/BC0 RD Data bus D07 D00
8 bits	4n	One	First	8 bits	4n		7 0
	4n + 1	One	First	8 bits	4n + 1		7 0
	4n + 2	One	First	8 bits	4n + 2		7 0
	4n + 3	One	First	8 bits	4n + 3		7 0

Table 14.24 Data alignment (big-endian) in 8-bit bus space (2 of 2)

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address		WR1/BC1 RD Data bus D15 D08	WR0/BC0 RD Data bus D07 D00
16 bits	4n	Two	First	8 bits	4n			15 8
			Second	8 bits	4n + 1	(p)		7 0
	4n + 2	Two	First	8 bits	4n + 2			15 8
			Second	8 bits	4n + 3	(p)		7 0
32 bits	4n	Four	First	8 bits	4n			31 24
			Second	8 bits	4n + 1	(p)		23 16
			Third	8 bits	4n + 2	(p)		15 8
			Fourth	8 bits	4n + 3	(p)		7 0

(p):Page access (only when page access is enabled in the PRENB and PWENB bits in CSnMOD)

14.4.2 Data Alignment Control for the SDRAM Area

14.4.2.1 32-Bit Bus Space

When a 32-bit width is selected for a bus space by the BSIZE[1:0] bits in SDCCR, the address buses A26 to A02 are enabled to output address signals in units of 32 bits, and the address buses A01 and A00 are disabled (always output low). The valid byte position is indicated by DQM0 to DQM3 signals.

The external data is accessed using the D31 to D24, D23 to D16, D15 to D8, and D7 to D0 pins. Data can be accessed in either 8-bit, 16-bit, or 32-bit units at a time.

In 32-bit bus space, the valid positions of data external to the chip and of SDRAM control signals (DQM0 to DQM3) differ according to whether the endian is big or little. Table 14.25 and Table 14.26 show data alignment control when the endian is little and big, respectively.

Table 14.25 Data alignment (little-endian) in 32-bit bus space

Data size	Access address	Number of access	Bus cycle	Unit of data	Address	DQM3		DQM2		DQM1		DQM0	
						WE							
						Data bus							
						D31	D24	D23	D16	D15	D8	D7	D0
8 bits	4n	One	First	8 bits	4n							7	0
	4n + 1	One			4n					7	0		
	4n + 2	One	First	8 bits	4n			7	0				
	4n + 3	One			4n	7	0						
16 bits	4n	One	First	16 bits	4n					15	8	7	0
	4n + 2	One	First	16 bits	4n	15	8	7	0				
32 bits	4n	One	First	32 bits	4n	31	24	23	16	15	8	7	0

Table 14.26 Data alignment (big-endian) in 32-bit bus space

Data size	Access address	Number of access	Bus cycle	Unit of data	Address	DQM3		DQM2		DQM1		DQM0	
						WE							
						Data bus							
						D31	D24	D23	D16	D15	D8	D7	D0
8 bits	4n	One	First	8 bits	4n	7	0						
	4n + 1	One			4n			7	0				
	4n + 2	One	First	8 bits	4n					7	0		
	4n + 3	One			4n							7	0
16 bits	4n	One	First	16 bits	4n	15	8	7	0				
	4n + 2	One	First	16 bits	4n					15	8	7	0
32 bits	4n	One	First	32 bits	4n	31	24	23	16	15	8	7	0

14.4.2.2 16-Bit Bus Space

When a 16-bit bus space is selected in the BSIZE[1:0] bits in SDCCR, address buses A26 to A01 are enabled to output address signals in 16-bit units, and the address bus A00 is disabled (always outputs low). The valid byte position is indicated by the DQM0 and DQM1 signals.

External data is accessed using the DQ15 to DQ08 and DQ07 to DQ00 pins and DQM0 and DQM1 control signals. Data can be accessed in either 8-bit or 16-bit units at a time.

The valid positions of control signals and data external to the MCU differ depending on the endian order. See Table 14.27 and Table 14.28.

For divided access, continuous access is possible by setting SDAMOD.BE = 1.

Table 14.27 Data alignment in 16-bit bus space with little-endian order for SDRAM area

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address	DQM1		DQM0	
						WE			
						Data bus			
						DQ15	DQ08	DQ07	DQ00
8 bits	4n	One	First	8 bits	4n			7	0
	4n + 1	One	First	8 bits	4n	7	0		
	4n + 2	One	First	8 bits	4n + 2			7	0
	4n + 3	One	First	8 bits	4n + 2	7	0		
16 bits	4n	One	First	16 bits	4n	15	8	7	0
	4n+2	One	First	16 bits	4n + 2	15	8	7	0
32 bits	4n	Two	First	16 bits	4n	15	8	7	0
			Second	16 bits	4n + 2	31	24	23	16

Table 14.28 Data alignment in 16-bit bus space with big-endian order for SDRAM area (1 of 2)

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address	DQM1		DQM0	
						WE			
						Data bus			
						DQ15	DQ08	DQ07	DQ00
8 bits	4n	One	First	8 bits	4n	7	0		
	4n + 1	One	First	8 bits	4n			7	0
	4n + 2	One	First	8 bits	4n + 2	7	0		
	4n + 3	One	First	8 bits	4n + 2			7	0
16 bits	4n	One	First	16 bits	4n	15	8	7	0
	4n+2	One	First	16 bits	4n + 2	15	8	7	0

Table 14.28 Data alignment in 16-bit bus space with big-endian order for SDRAM area (2 of 2)

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address	DQM1		DQM0	
						WE			
						Data bus			
						DQ15	DQ08	DQ07	DQ00
32 bits	4n	Two	First	16 bits	4n	31	24	23	16
			Second	16 bits	4n + 2	15	8	7	0

14.4.2.3 8-Bit Bus Space

When an 8-bit width is selected in the BSIZE[1:0] bits in SDCCR, address buses A26 to A00 are enabled to output address signals in 8-bit units.

External data is accessed using the DQ07 to DQ00 pins and DQM0 control signal. Eight-bit data is accessed in single 8-bit access, 16-bit data with two 8-bit accesses, and 32-bit data with four 8-bit accesses.

The valid positions of control signals and data external to the MCU differ depending on the endian order. See Table 14.29 and Table 14.30.

For divided access, continuous access is possible by setting SDAMOD.BE = 1.

Table 14.29 Data alignment in 8-bit bus space with little-endian order for SDRAM area (1 of 2)

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address	DQM1		DQM0	
						WE			
						Data bus			
						DQ15	DQ08	DQ07	DQ00
8 bits	4n	One	First	8 bits	4n			7	0
	4n + 1	One	First	8 bits	4n + 1			7	0
	4n + 2	One	First	8 bits	4n + 2			7	0
	4n + 3	One	First	8 bits	4n + 3			7	0
16 bits	4n	Two	First	8 bits	4n			7	0
			Second	8 bits	4n + 1			15	8
	4n + 2	Two	First	8 bits	4n + 2			7	0
			Second	8 bits	4n + 3			15	8

Table 14.29 Data alignment in 8-bit bus space with little-endian order for SDRAM area (2 of 2)

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address	DQM1 / WE / Data bus DQ15–DQ08	DQM0 / WE / Data bus DQ07–DQ00
32 bits	4n	Four	First	8 bits	4n		7 0
			Second	8 bits	4n + 1		15 8
			Third	8 bits	4n + 2		23 16
			Fourth	8 bits	4n + 3		31 24

Table 14.30 Data alignment in 8-bit bus space with big-endian order for SDRAM area

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address	DQM1 / WE / Data bus DQ15–DQ08	DQM0 / WE / Data bus DQ07–DQ00
8 bits	4n	One	First	8 bits	4n		7 0
	4n + 1	One	First	8 bits	4n + 1		7 0
	4n + 2	One	First	8 bits	4n + 2		7 0
	4n + 3	One	First	8 bits	4n + 3		7 0
16 bits	4n	Two	First	8 bits	4n		15 8
			Second	8 bits	4n + 1		7 0
	4n + 2	Two	First	8 bits	4n + 2		15 8
			Second	8 bits	4n + 3		7 0
32 bits	4n	Four	First	8 bits	4n		31 24
			Second	8 bits	4n + 1		23 16
			Third	8 bits	4n + 2		15 8
			Fourth	8 bits	4n + 3		7 0

14.5 Operation of CS Area Controller

14.5.1 Separate Bus

This section describes the periods shown in the timing diagrams. The CS area controller (CSC) operates in synchronization with the external bus clock, BCLK. Operation cycles, such as wait cycles, specified in the CSC register, are counted on BCLK. In the following description, the frequencies of BCLK and EBCLK pin output are the same, unless otherwise noted. Access through the external bus starts at the same point as the output of a rising edge on the EBCLK pin. However, if the external bus clock, BCLK, and the output on the EBCLK pin are at different frequencies, the wait settings can cause the start of access for the second and subsequent rounds to coincide with the falling edge of the output on the EBCLK pin. See Figure 14.10 to Figure 14.14. If recovery cycles are inserted for bus access, the setting for the number of recovery cycles can also cause the start of access for the second and subsequent rounds to coincide with the falling edge of the output on the EBCLK pin. See Figure 14.32.

Tw1 to Twn (Clock Cycles of Wait for a Normal Read Cycle or Normal Write Cycle)

The period from Tw1 to Twn is the number of clock cycles from the start of access through the external bus clock to 1 cycle before the strobe signal is valid. The number of cycles is selectable from 0 to 31. Within this period, the timing of CSn, RD, and WRn assertion (driving the signals low) is determined by the respective wait settings. The wait periods are controlled by the following bits in the CSn Wait Control Register 2 (CSnWCR2) .

- CS Assert Wait Select bits (CSON)
- RD Assert Wait Select bits (RDON)
- WR Assert Wait Select bits (WRON)
- Write-Data Output Wait Select bits (WDON)

The number of clock cycles for each of these wait periods is selectable as a value from 0 to 7, counted from the start of external bus access. The selectable number of cycles is also within the overall number of clock cycles required for waiting read or write.

Tend (Clock Cycle where the Strobe Signal is Valid)

Tend is the next clock cycle after completion of the wait period for a normal cycle of read or write, or for a cycle of page read or page write. If the wait select bit for these cycles is 0, bus access starts on the clock cycle where the strobe signal is valid. The RD and WRn signals are negated in the next clock cycle. For a read access, the clock cycle where the strobe signal is valid is where the data to be read is sampled. If an external wait is enabled, the wait signal is sampled on the cycle where the strobe signal is valid. The bus cycle is extended if the wait signal is low. The bus cycle completes in the next clock cycle if the wait signal is high. Tend indicates the cycle where sampling of the wait signal starts.

After the first cycle where the strobe signal is valid during page access, second and subsequent page access operations (see Tpw1 to Tpwn (Page-Read Cycle Wait or Page-Write Cycle Wait)) start in the next cycle, except during write access with a setting other than 0 for write-data output extension clock cycles (see Tdw1 to Tdwn (Write-Data Output Extension Clock Cycles)). If the setting for the RD or WR assertion wait is any value other than 0, the RD and WRn signals are negated in the next clock cycle. If the setting is 0, assertion continues. Additionally, the CSn signal continues to be asserted rather than negated.

Tn1 to Tnm (Clock Cycles of CS Extension)

For normal access, Tn1 to Tnm represent the clock cycles of the period following the cycle where the strobe signal is valid (Tend) up to negation of the CSn signal. For read or write access, the negation timing can be controlled by the read-access CS Extension Cycle Select bits (CSROFF) and the write-access CS Extension Cycle Select bits (CSWOFF) in the CSn Wait Control Register 2 (CSnWCR2). The number of cycles is counted from the cycle following the cycle where the strobe signal is valid.

For page access, Tn1 to Tnm represent the clock cycles of the period following the last cycle where the strobe signal is valid up to negation of the CSn signal.

For write access, setting the Write Data Output Extension Cycle Select bits (WDOFF) controls extension of the period where the address and output data is valid.

Tdw1 to Tdwn (Write-Data Output Extension Clock Cycles)

For write access, if the wait setting for the write-data output extension is any value other than 0, the specified clock cycles are inserted from the cycle following the cycle where the strobe signal is valid (Tend).

For normal access, this period is inserted within the clock cycle period for CS extension (Tn1 to Tnm).

For page access, this period is inserted within the clock cycle period where the strobe signal is valid and subsequent page accesses, or within the clock cycle period for the CS extension (Tn1 to Tnm). Valid address and data output are extended over this period, and the WRn signal is negated.

Tpw1 to Tpwn (Page-Read Cycle Wait or Page-Write Cycle Wait)

For the second and subsequent bus cycles during page access, the values for a page-read cycle wait or page-write cycle wait are used instead of the settings for a normal read or write cycle wait. The settings in the WR Assert Wait Select bits become enabled in the same way as for the first access. The RD assertion control operation depends on the page read access mode setting (the PRMOD bit in CSnMOD) as follows:

- CSnMOD.PRMOD = 0: A wait for RD assertion is inserted in the same way as for the first access, and the RD signal is negated
- CSnMOD.PRMOD = 1: Although a wait for RD assertion is inserted in the same way as for normal-access compatibility mode, the RD signal continues to be asserted over this period.

Tr1 to Trn (Recovery Cycles)

Recovery cycles can be inserted from the point where a bus cycle is complete (CSn signal negation). The number of recovery cycles can be controlled by setting the Read Recovery (RRCV) or Write Recovery (WRCV) bits in the CSn Recovery Cycle Register (CSnREC). Both numbers of recovery cycles are counted from the end of a bus cycle (CSn negation) and can be selected from 0 to 15 cycles. For more information, see section 14.5.4. Insertion of Recovery Cycles.

(1) Normal Access

When the PRENB and PWENB bits in CSnMOD are set to 0 to disable page read and page write access, all bus accesses take the form of normal read and write operations. Even when these bits are set to 1 to enable page read and page write access, bus access other than page access takes the form of normal read and write operations. Figure 14.3 to Figure 14.5 show the normal access operations.

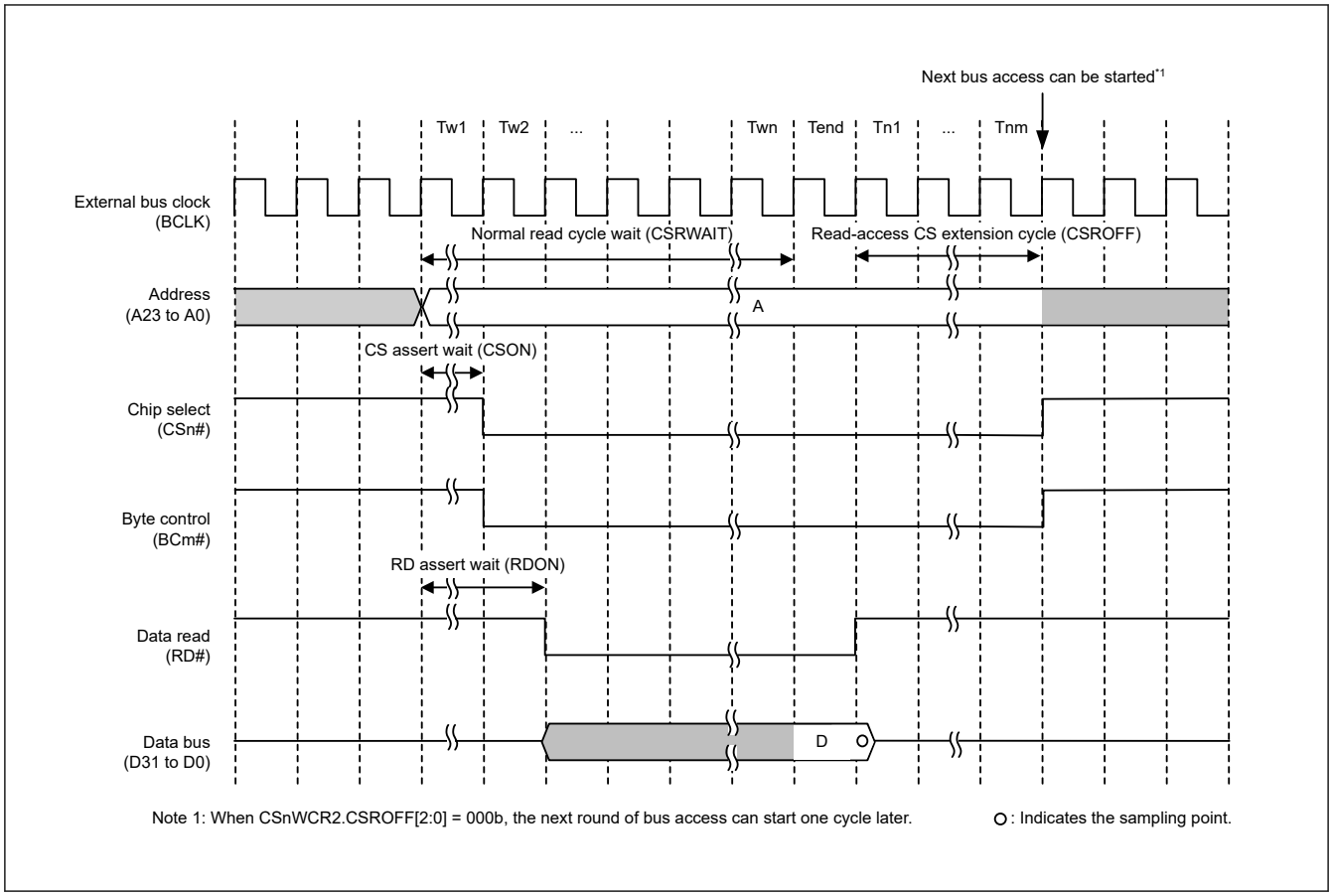


Figure 14.3 Bus timing for normal read operation (n = 0 to 7, m = 0 to 3)

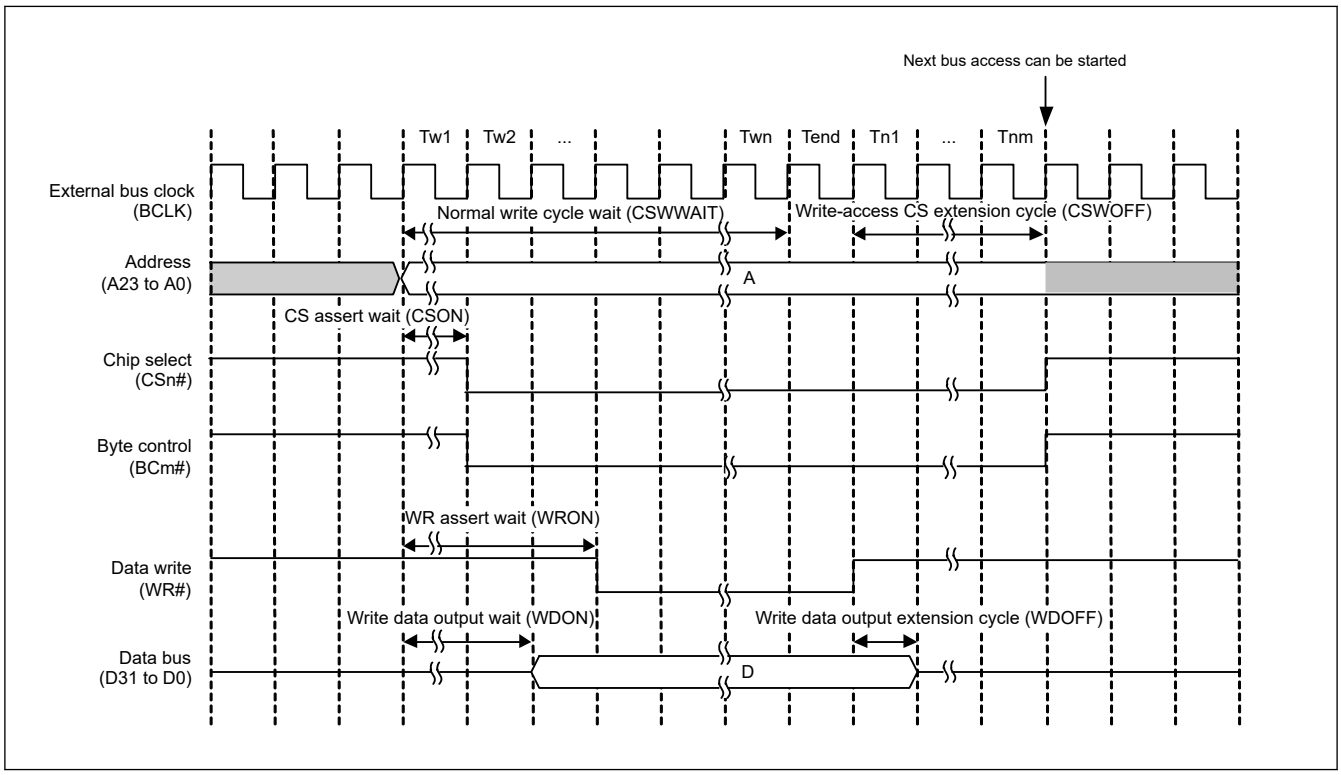


Figure 14.4 Bus timing for normal write operation in single-write strobe mode (n = 0 to 7, m = 0 to 3)

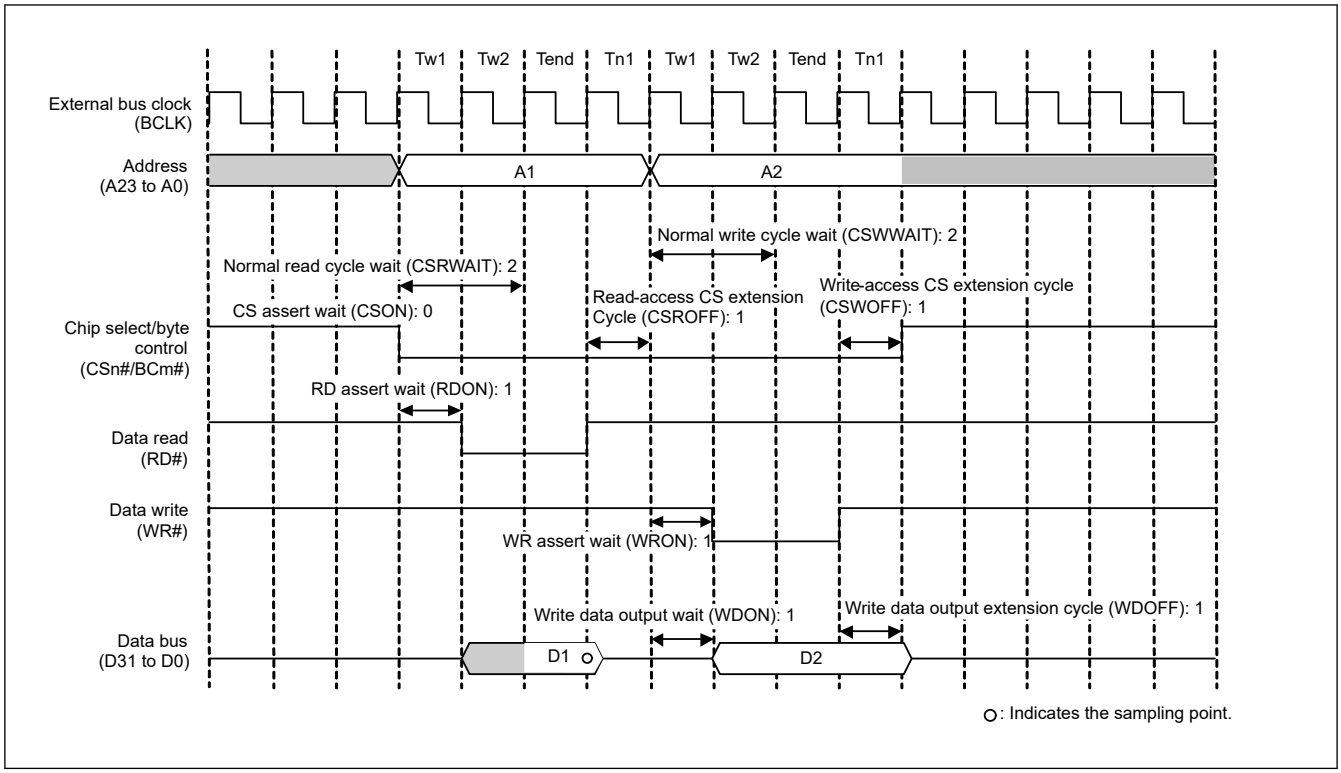


Figure 14.5 Example of normal access operation for read and write (n = 0 to 7, m = 0 to 3)

When two or more rounds of external bus access are required in response to a single request for transfer from a bus master, normal access operations are repeated. See Tw1 to Twn (Clock Cycles of Wait for a Normal Read Cycle or Normal Write Cycle) to Tdw1 to Tdwn (Write-Data Output Extension Clock Cycles). Figure 14.6 and Figure 14.7 show examples of operations when two rounds of bus access are generated in response to a single transfer request. If the recovery cycle insertion condition is satisfied, recovery cycles Tr1 to Trn (Recovery Cycles) are also inserted in the second and subsequent external bus accesses. See Figure 14.30.

The values in the wait control registers shown in the figures are example settings. In your application, set the registers appropriately for the specifications of connected devices.

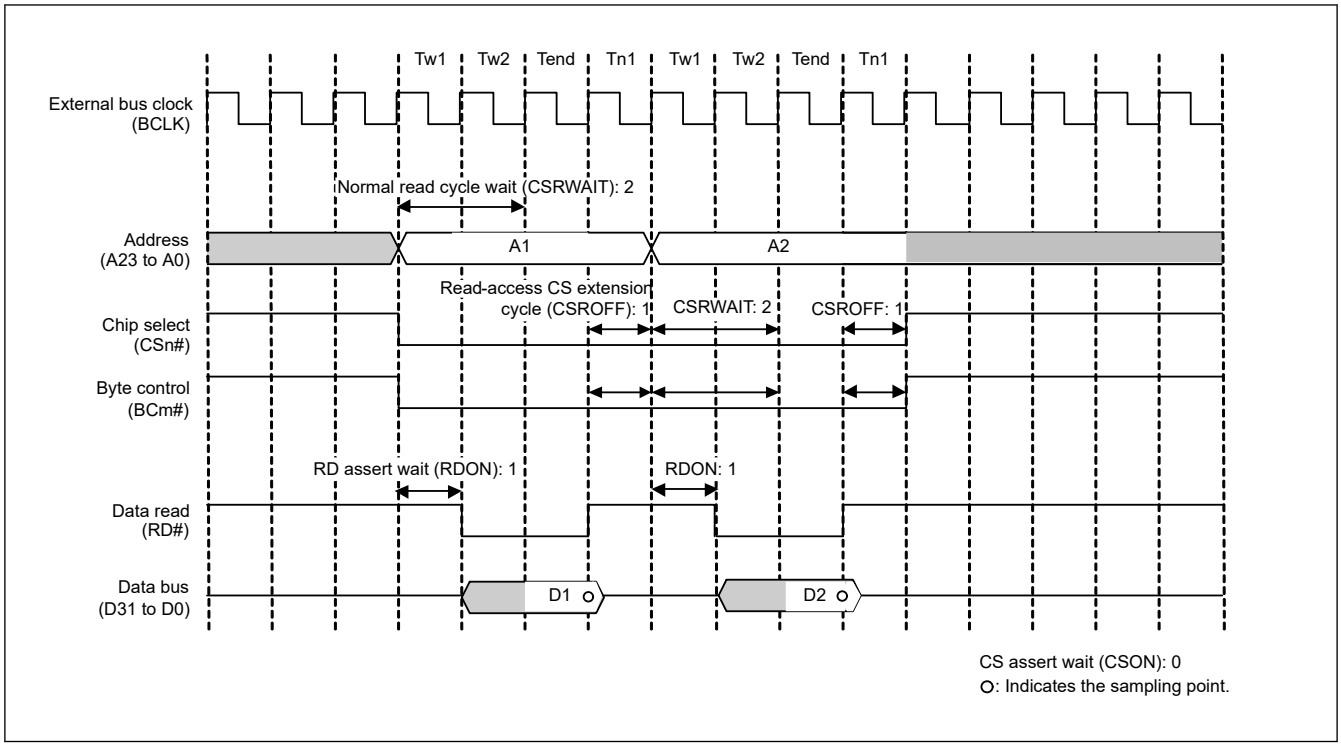


Figure 14.6 Example of normal read operation when two rounds of bus access are generated in response to a single transfer request (n = 0 to 7, m = 0 to 3)

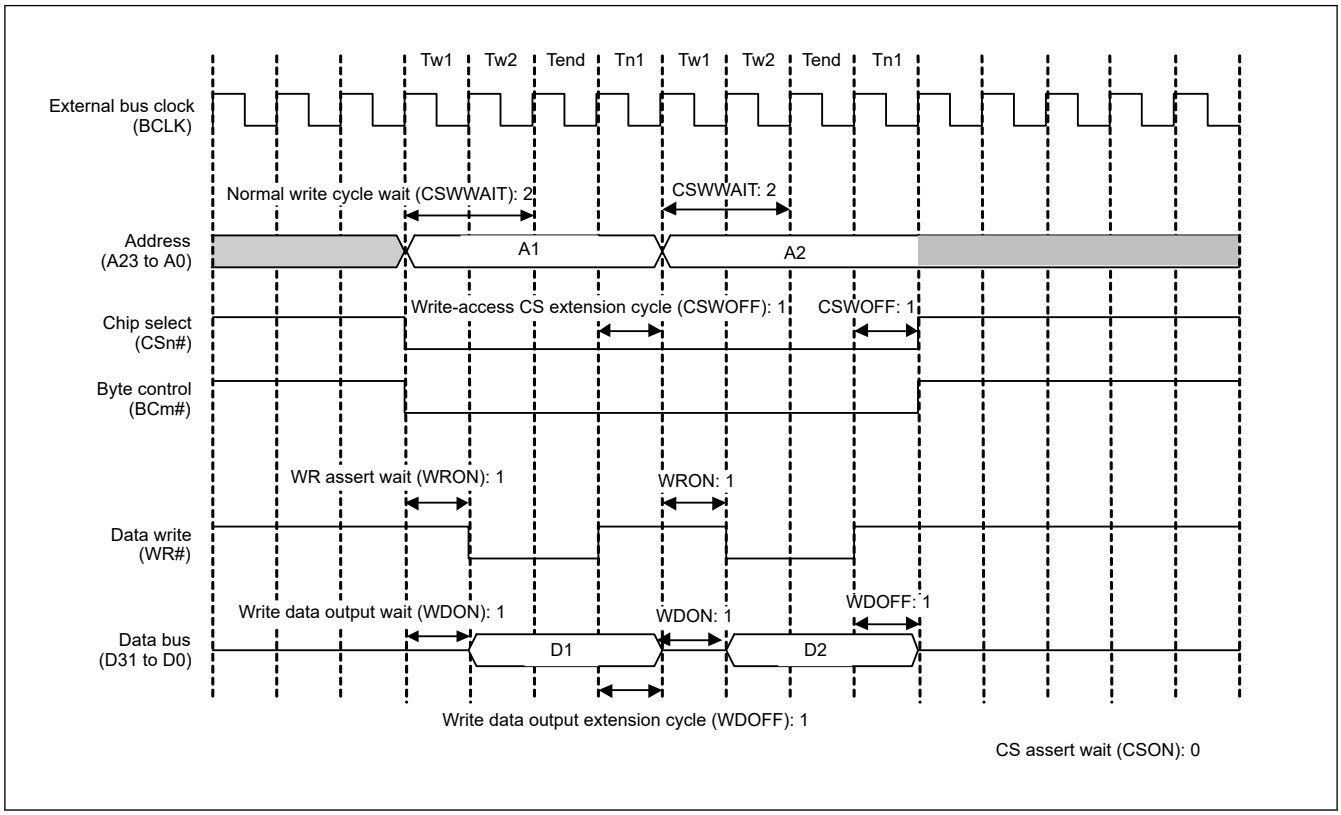


Figure 14.7 Example of normal write operation when two rounds of bus access are generated in response to a single transfer request in single-write strobe mode (n = 0 to 7, m = 0 to 3)

Figure 14.8 and Figure 14.9 show examples of normal read and write accesses to a 32-bit bus space in 16 bits.

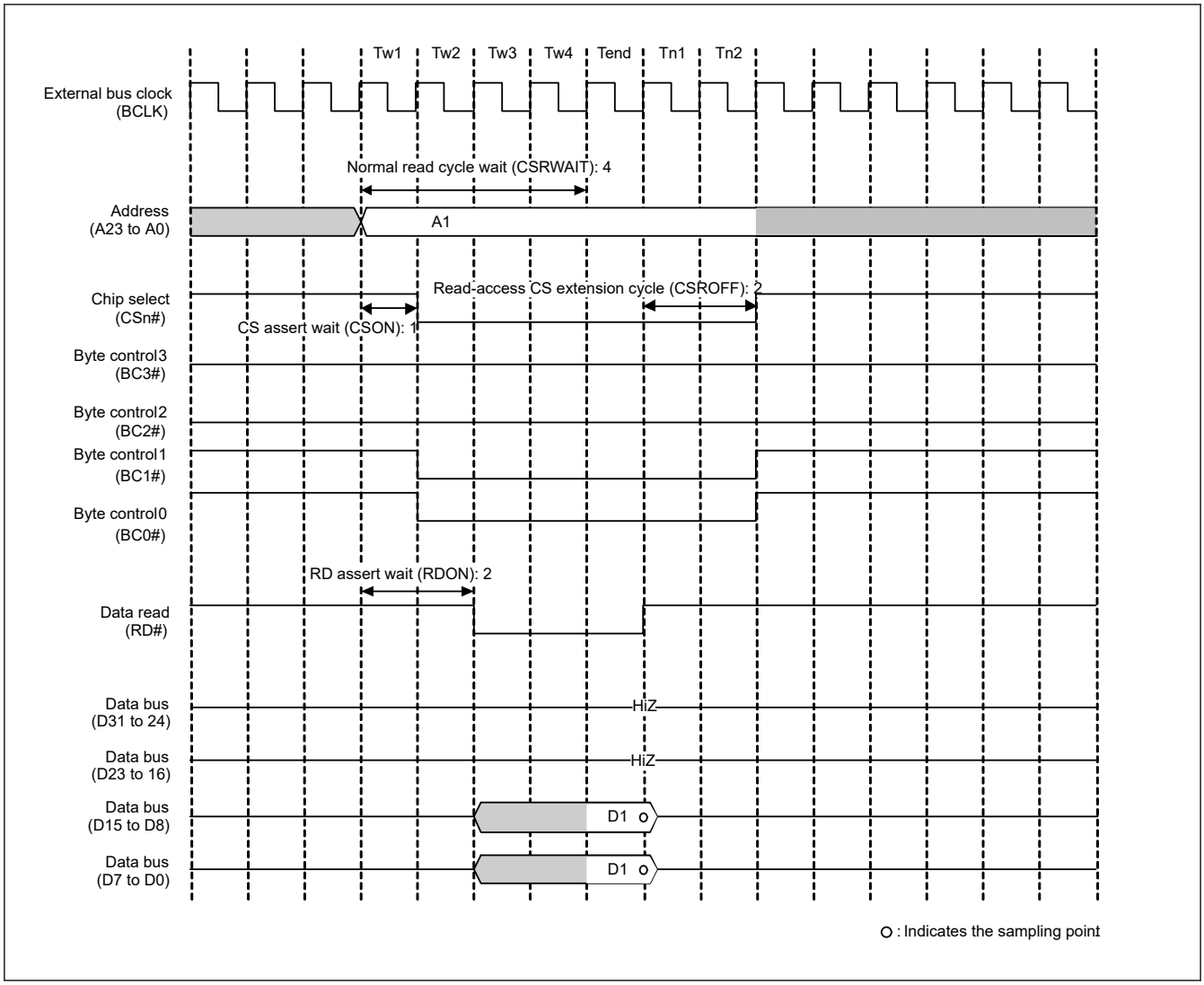


Figure 14.8 Example of normal read operation when 32-bit bus space is accessed in 16 bits (n = 0 to 7)

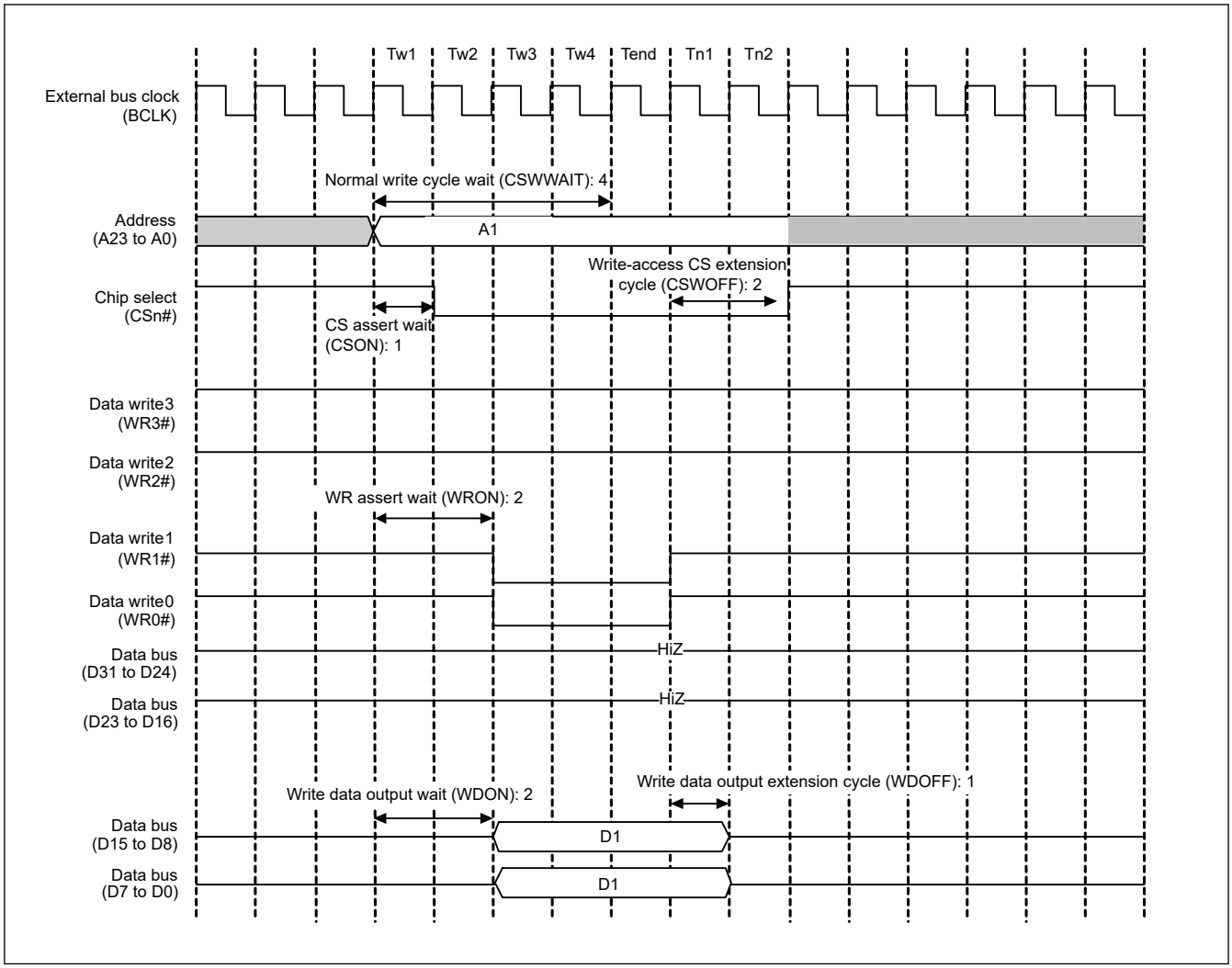


Figure 14.9 Example of normal write operation when 32-bit bus space is accessed in 16 bits, in byte strobe mode (n = 0 to 7)

Figure 14.10 to Figure 14.14 show examples of normal accesses made with the 1/2 BCLK selected with the EBCLK pin output select bit.

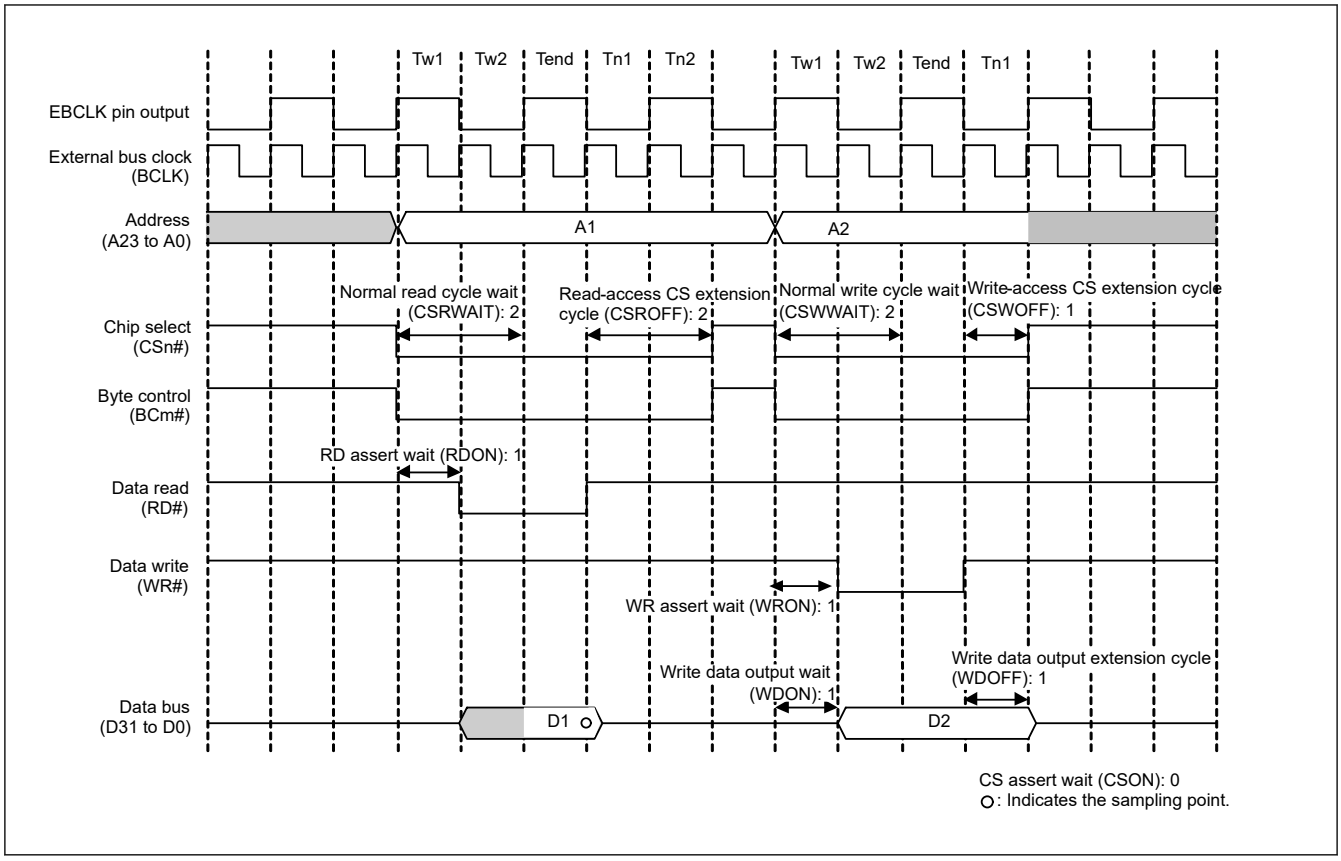


Figure 14.10 Example of normal access when BCLK/2 is selected in the EBCLK Pin Output Select bit (n = 0 to 7, m = 0 to 3)

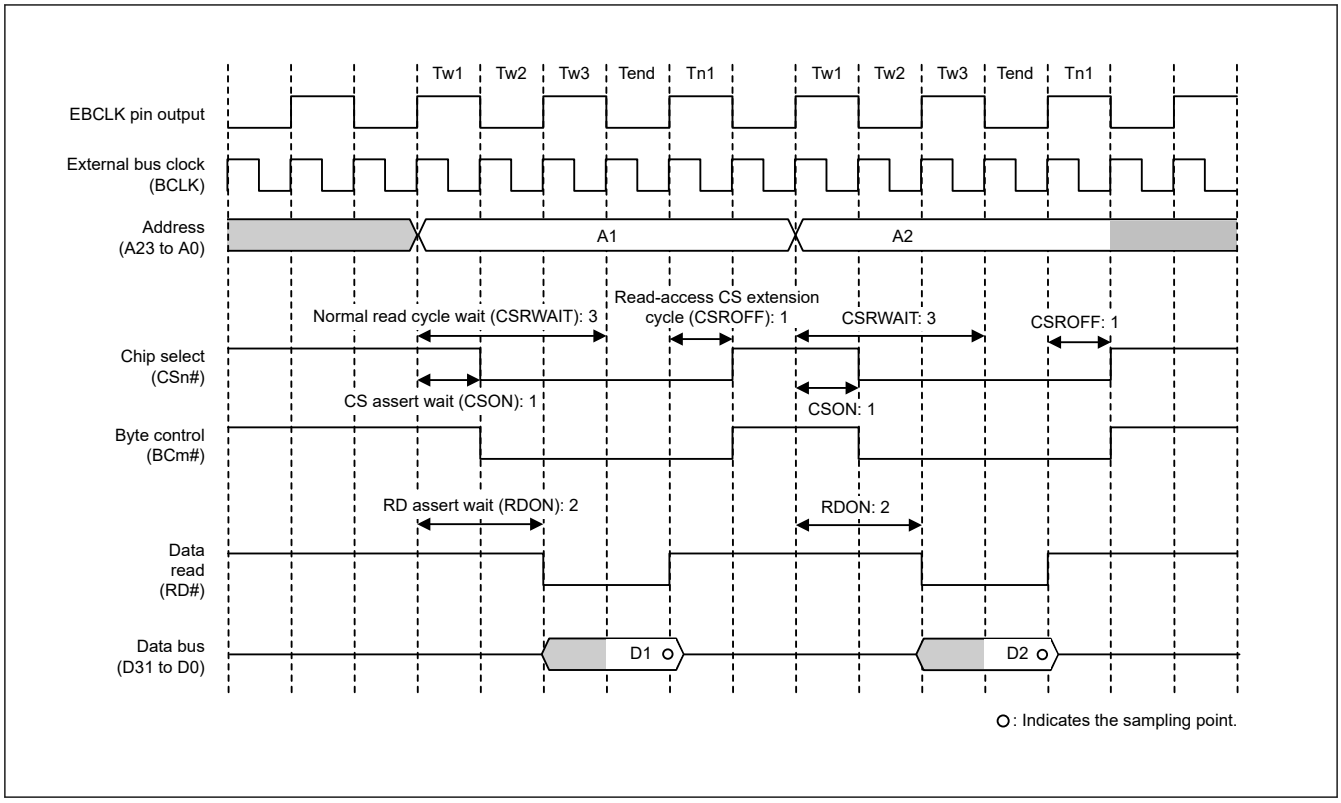


Figure 14.11 Example of normal read operation when BCLK/2 is selected in the EBCLK Pin Output Select bit (n = 0 to 7, m = 0 to 3)

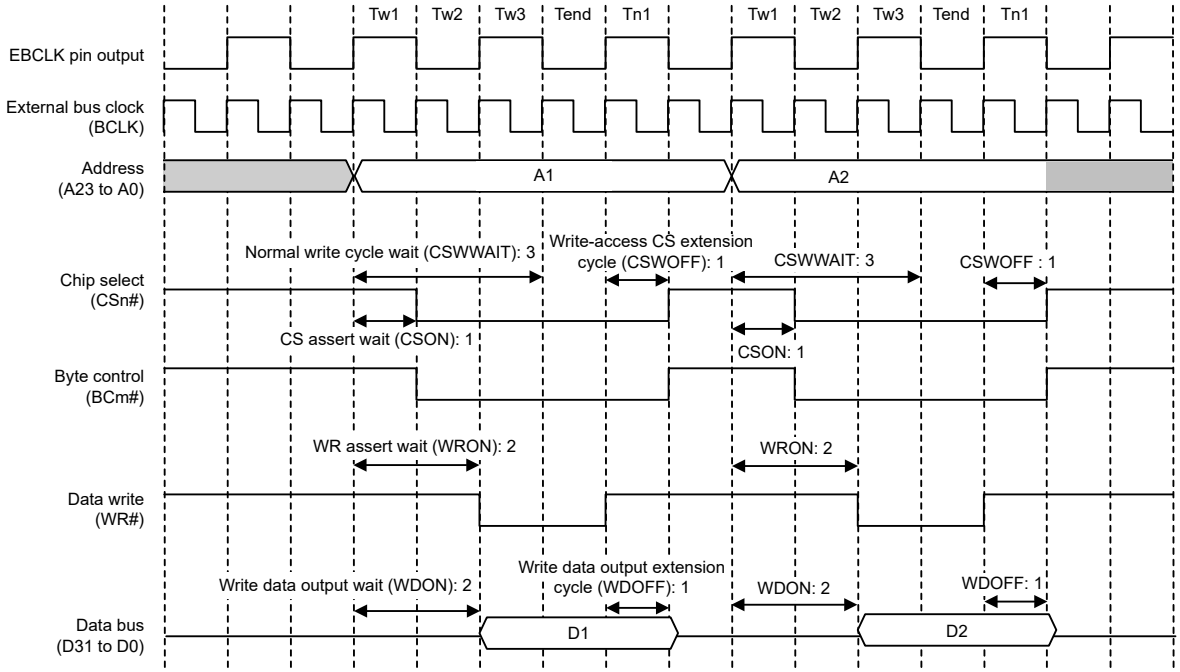


Figure 14.12 Example of normal write operation when BCLK/2 is selected in the EBCLK Pin Output Select bit (n = 0 to 7, m = 0 to 3)

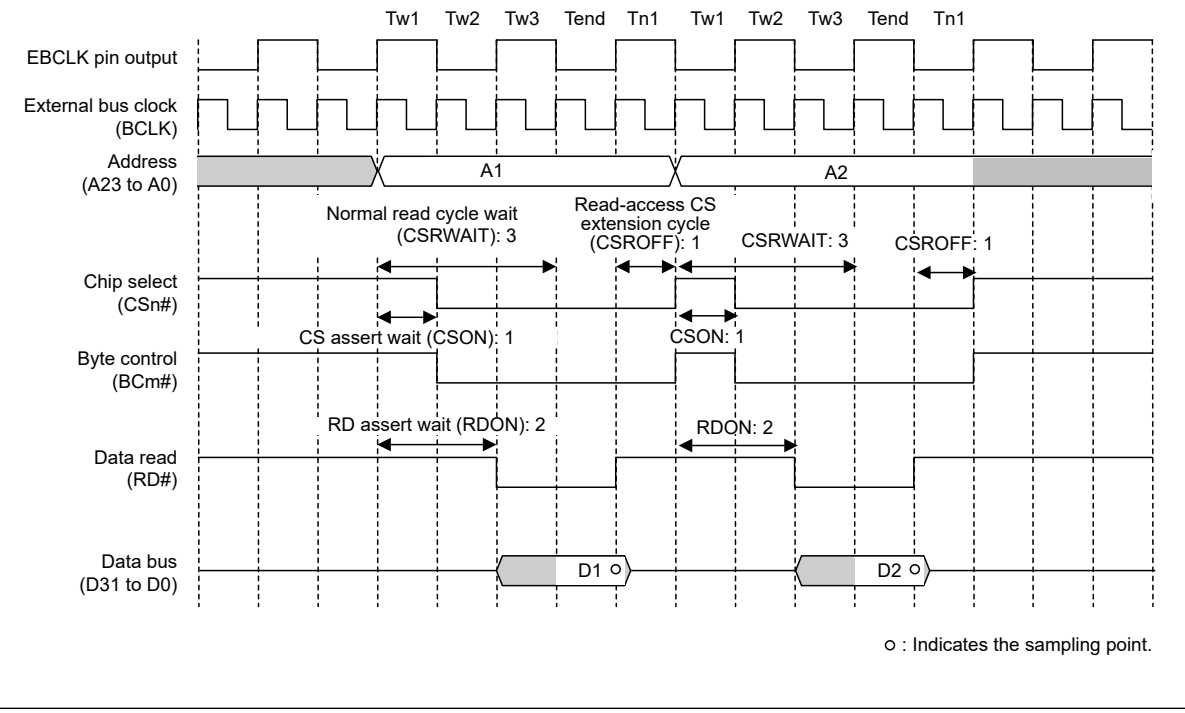


Figure 14.13 Example of normal read operation when BCLK/2 is selected in the EBCLK Pin Output Select bit and two rounds of bus access are generated in response to a single transfer request (n = 0 to 7, m = 0 to 3)

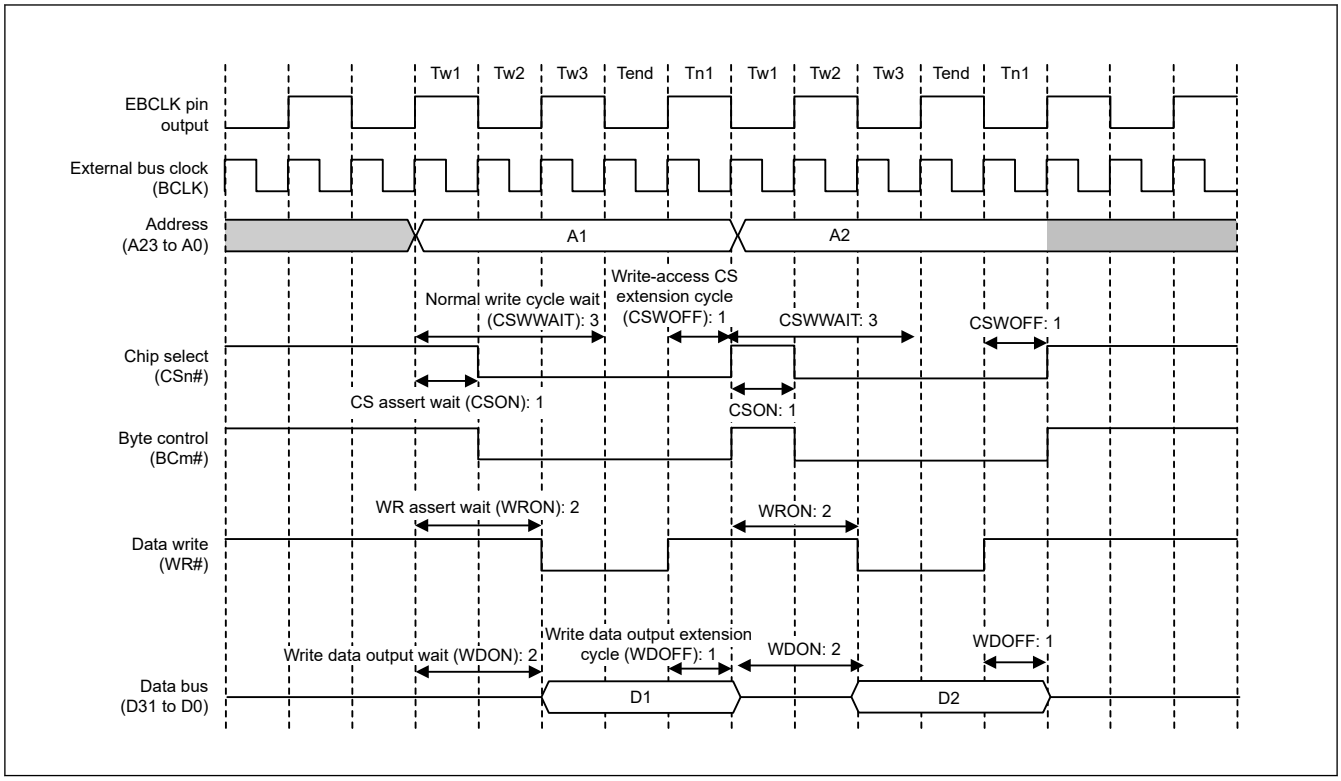


Figure 14.14 Example of normal write operation when BCLK/2 is selected in the EBCLK Pin Output Select bit and two rounds of bus access are generated in response to a single transfer request (n = 0 to 7, m = 0 to 3)

(2) Page Access

When the PRENB and PWENB bits in CSnMOD are set to 1 to enable page read and page write access, the bus access for page access operations becomes page read and write. Page access can only occur when two or more rounds of external bus access are required for a single transfer request from the bus master. However, normal access is made when split accesses are not aligned or access extends across the 32-bit boundary. See Table 14.21 to Table 14.24 for the conditions under which page access occurs.

Figure 14.15 and Figure 14.16 show examples of page access operations.

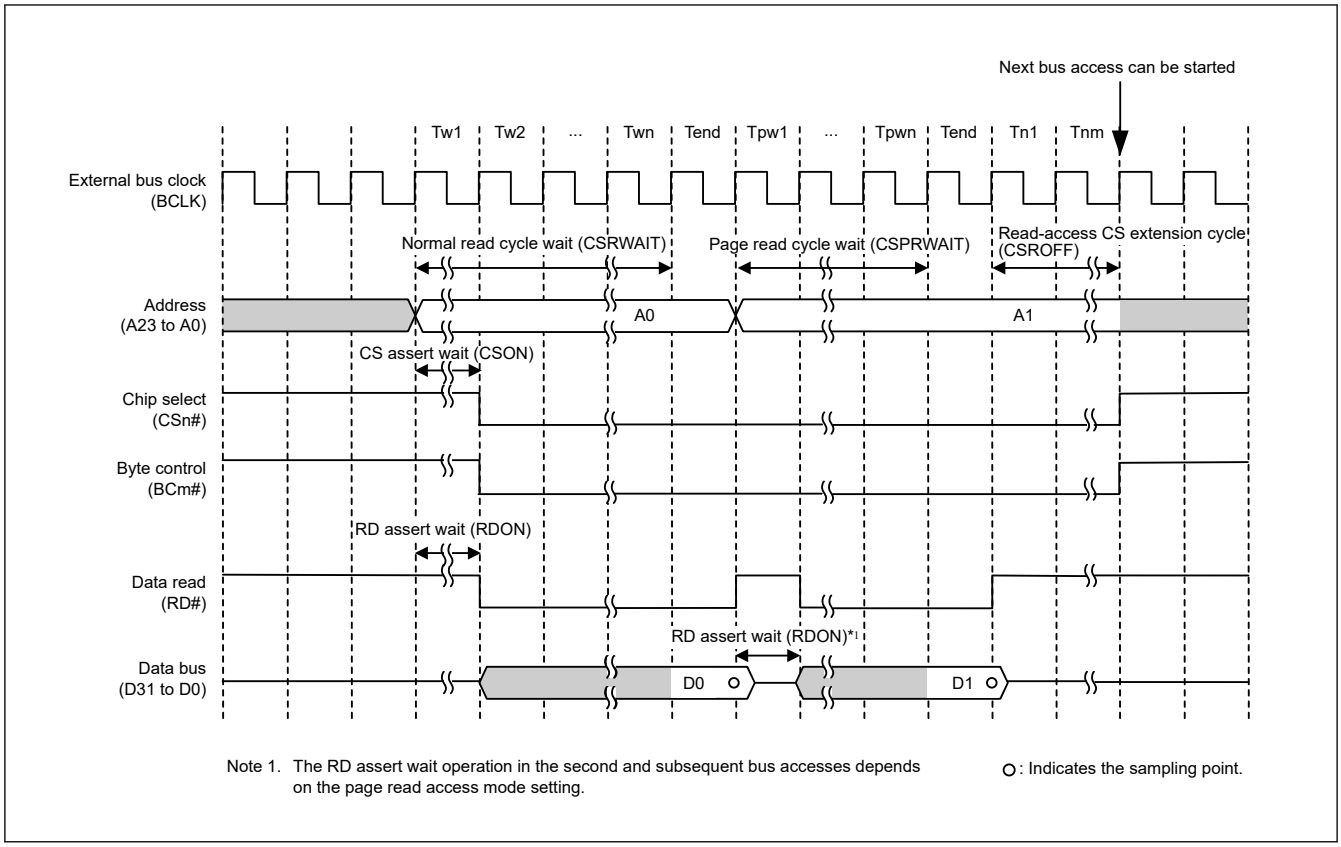


Figure 14.15 Page read access timing (n = 0 to 7, m = 0 to 3)

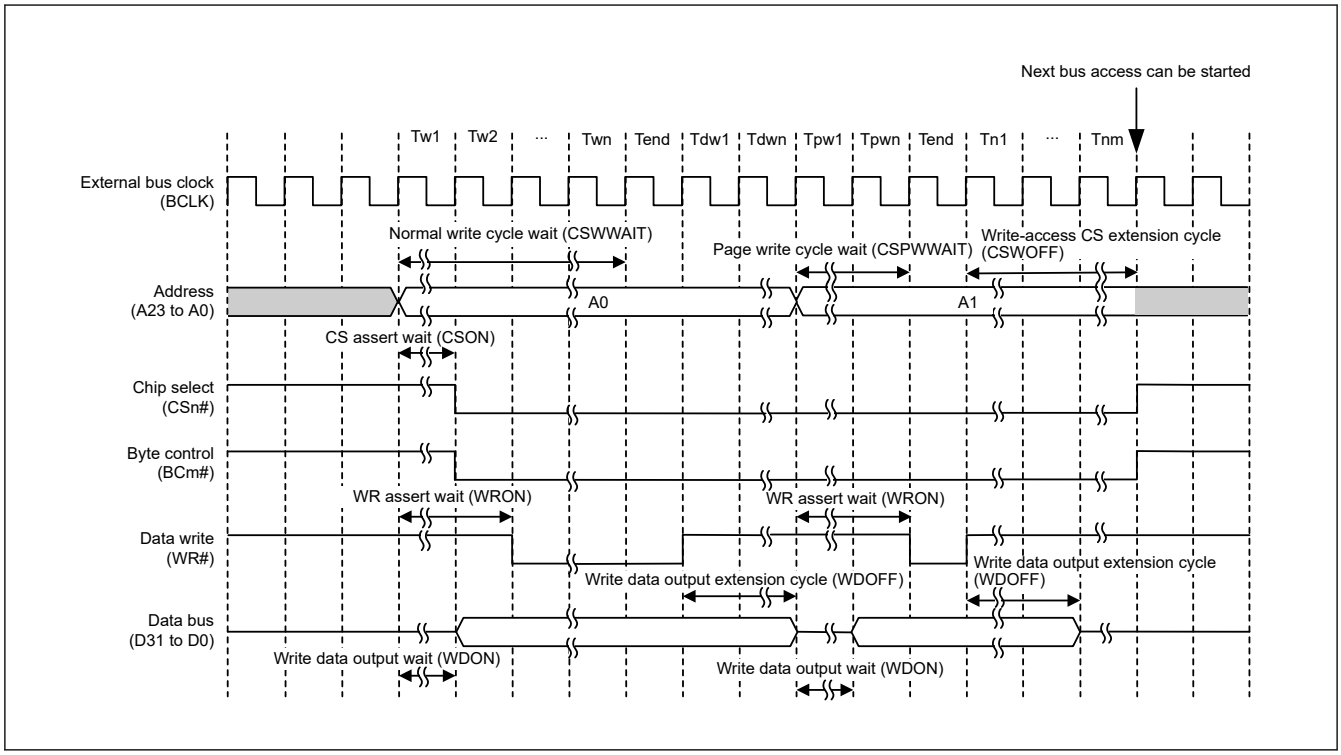


Figure 14.16 Page write access timing (n = 0 to 7, m = 0 to 3)

Figure 14.17 and Figure 14.18 show examples of operations for access to a 16-bit bus space in 32 bits. The values of the wait control registers shown in the figures are example settings. In your application, set the registers appropriately for the specifications of connected devices.

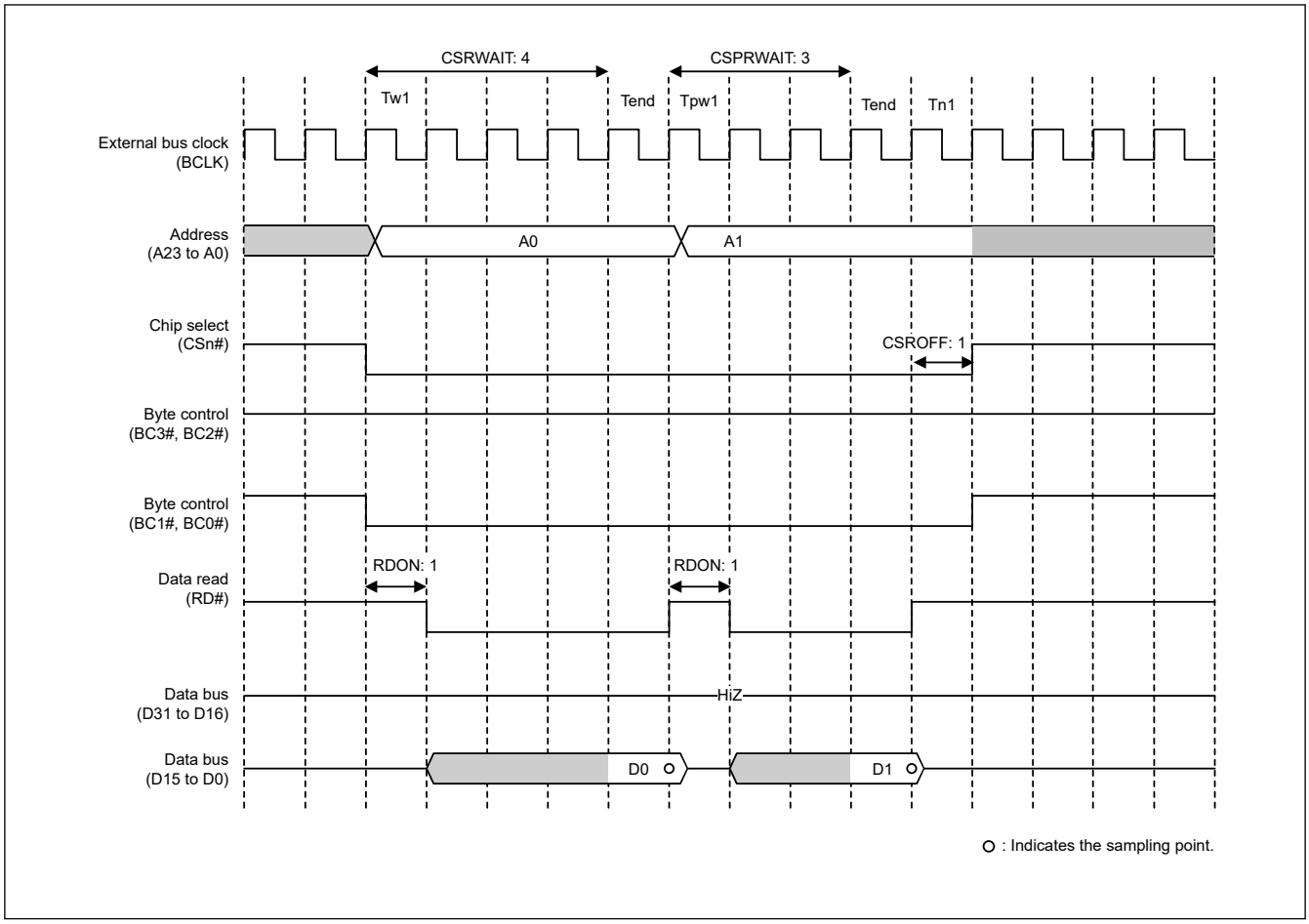


Figure 14.17 Example page read access operation when 16-bit bus space is accessed in 32 bits (n = 0 to 7)

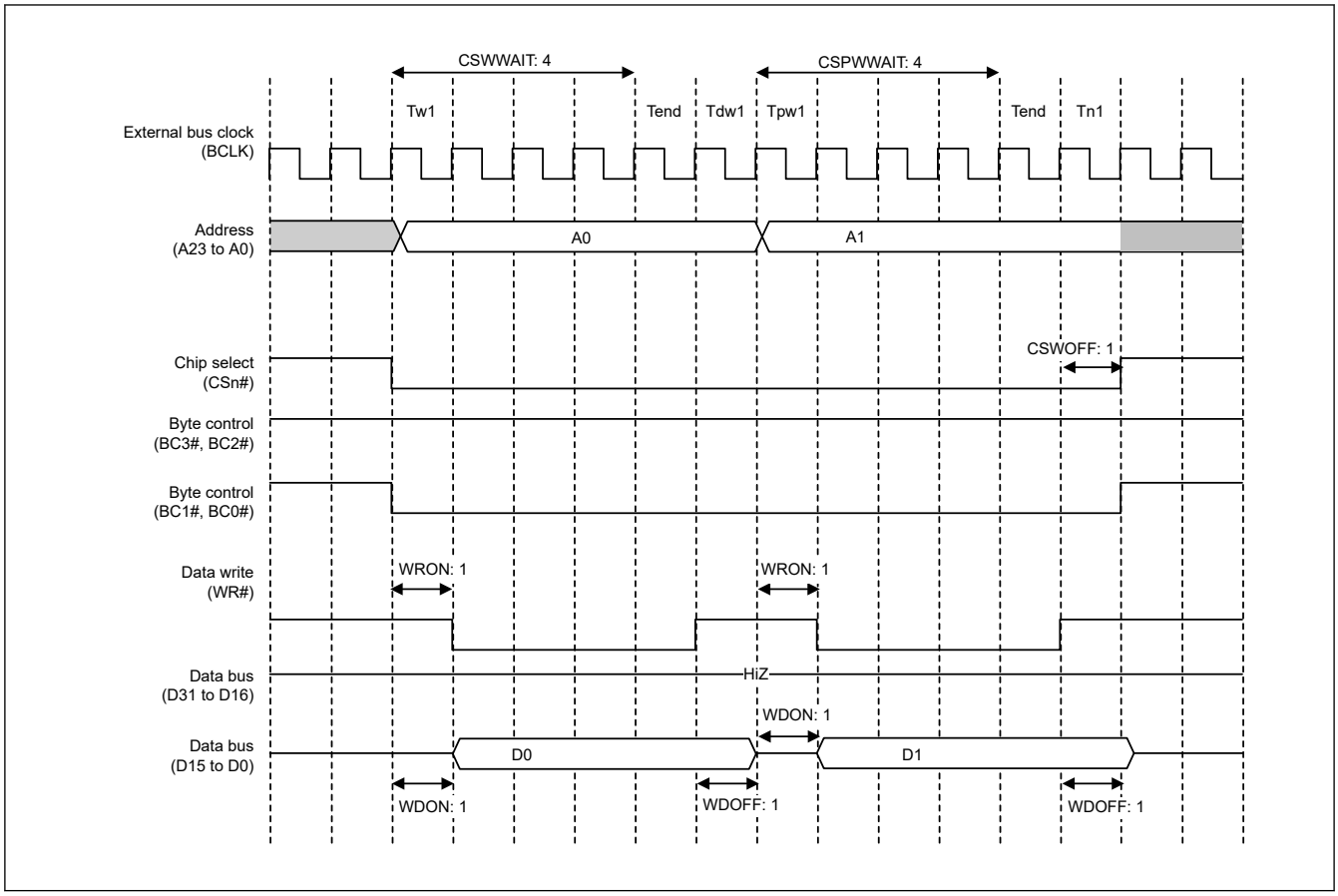


Figure 14.18 Example page write access operation when 16-bit bus space is accessed in 32 bits in single-write strobe mode (n = 0 to 7)

Figure 14.19 and Figure 14.20 show examples of page access operations when BCLK/2 is selected as the frequency division in the EBCLK Pin Output Select bit.

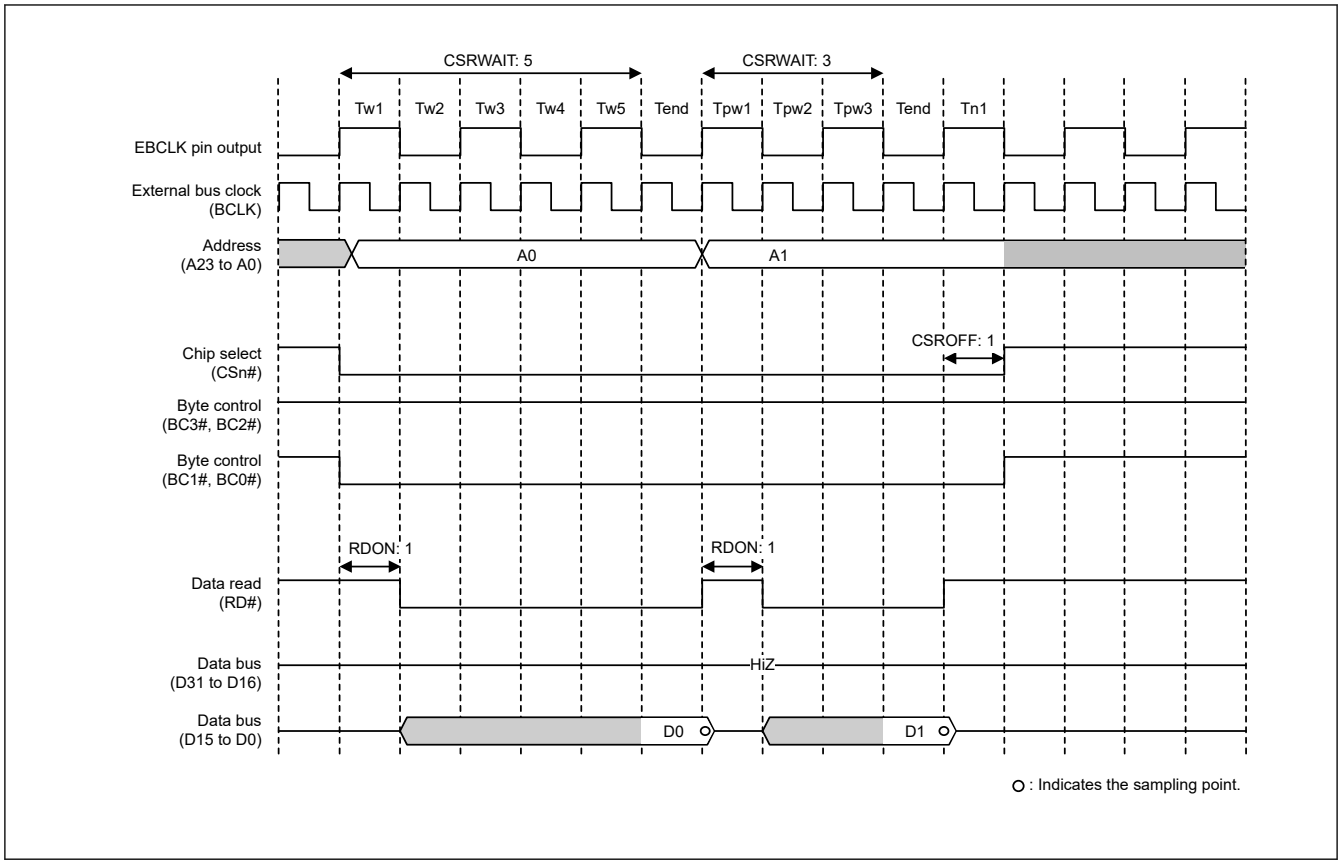


Figure 14.19 Example page read access operation when BCLK/2 is selected in the EBCLK Pin Output Select bit and two rounds of bus access are generated in response to a single transfer request (n = 0 to 7)

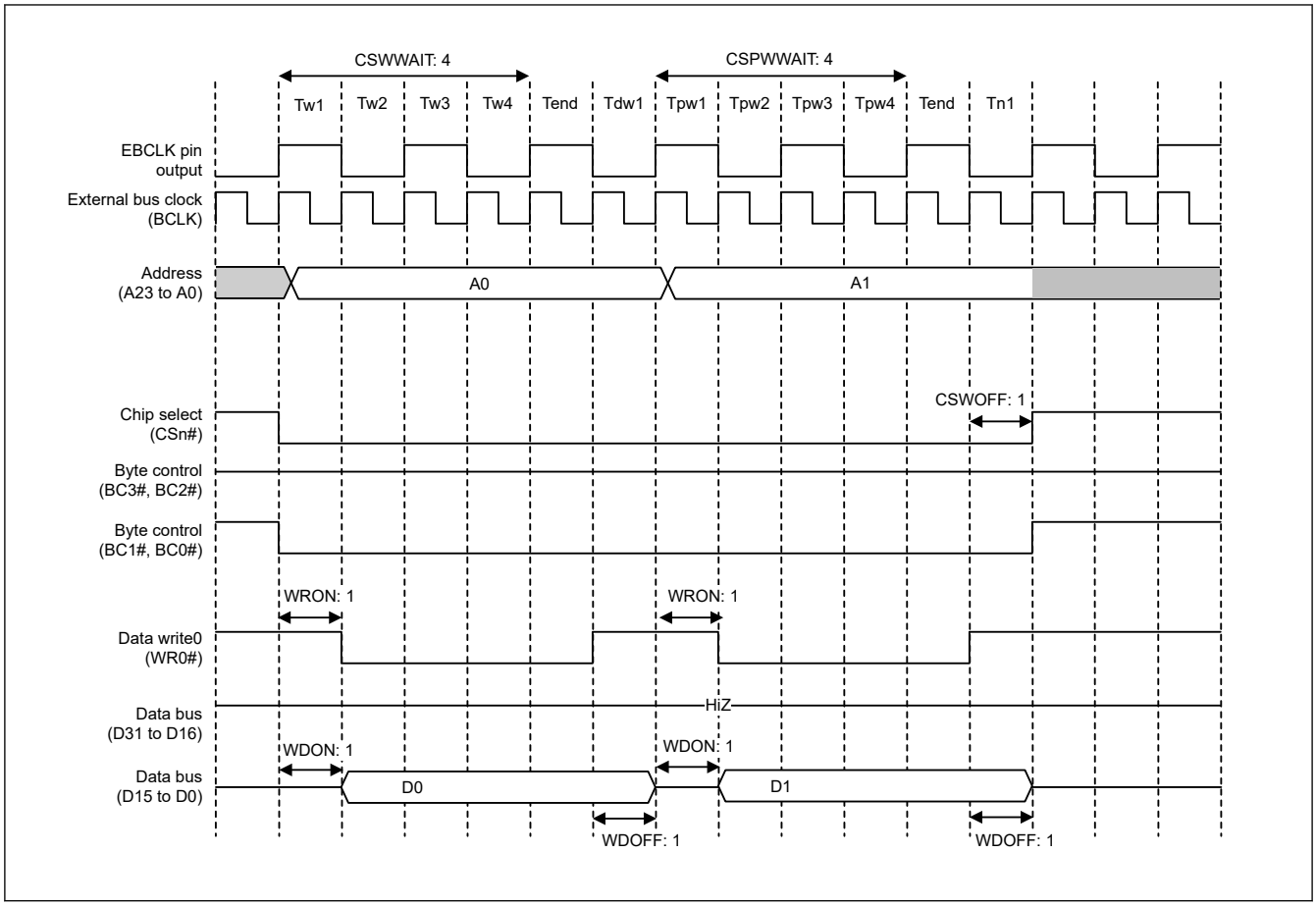


Figure 14.20 Example page write access operation when BCLK/2 is selected in the EBCLK Pin Output Select bit and two rounds of bus access are generated in response to a single transfer request, in single-write strobe mode (n = 0 to 7)

14.5.2 Address/Data Multiplexed Bus

When the address/data Multiplexed I/O Interface Select bit (MPXEN) in CSnCR is set to 1, addresses and data can be multiplexing input/output to/from the D15 to D00 pins in the corresponding area. Using this function enables direct connection of this MCU to peripherals of the MCU requiring address/data multiplexing. When 8-bit width is selected with the BSIZE[1:0] bits in CSnCR, D7 to D00 are multiplexed with A07 to A00. When 16-bit width is selected, D15 to D00 are multiplexed with A15 to A00. In the address/data multiplexed I/O space, accesses are controlled with the ALE, RD, WRn, and BCn signals.

Byte strobe mode or single-write strobe mode is selectable in the same way as for a separate bus. However, with regard to the BCn signals within the address cycle, the byte-control signal is output for the data being read or written.

During the address/data multiplexed I/O space access, after the number of wait cycles specified by the Address Cycle Wait Select bits (AWAIT[1:0]) in CSnWCR2 is inserted in the address output cycle, data access is performed.

Ta1 to Tan (Address Cycle Wait)

The period Ta1 to Tan is valid only when the address/data multiplexed I/O space is specified. This period is made up of the number of clock cycles between the start of external bus access and 1 cycle before the address latch (ALE) signal is negated. The number of cycles are selectable within the range from 0 to 3. Addresses are output until the next cycle of ALE signal negation (address cycle). The timing of ALE signal is the same as that of CS assertion. After the address cycle, a data cycle is started. CSnWCR1 and CSnWCR2 should be set so that an address cycle and a data cycle do not overlap.

Page access to the address/data multiplexed I/O space is invalid. When the PRENB or PWENB bit in CSnMOD is set to 1 to enable page-read or page-write access, these settings are ignored and normal read or write operation is performed.

Figure 14.21 to Figure 14.23 show examples of operations with the address/data multiplexed I/O interface

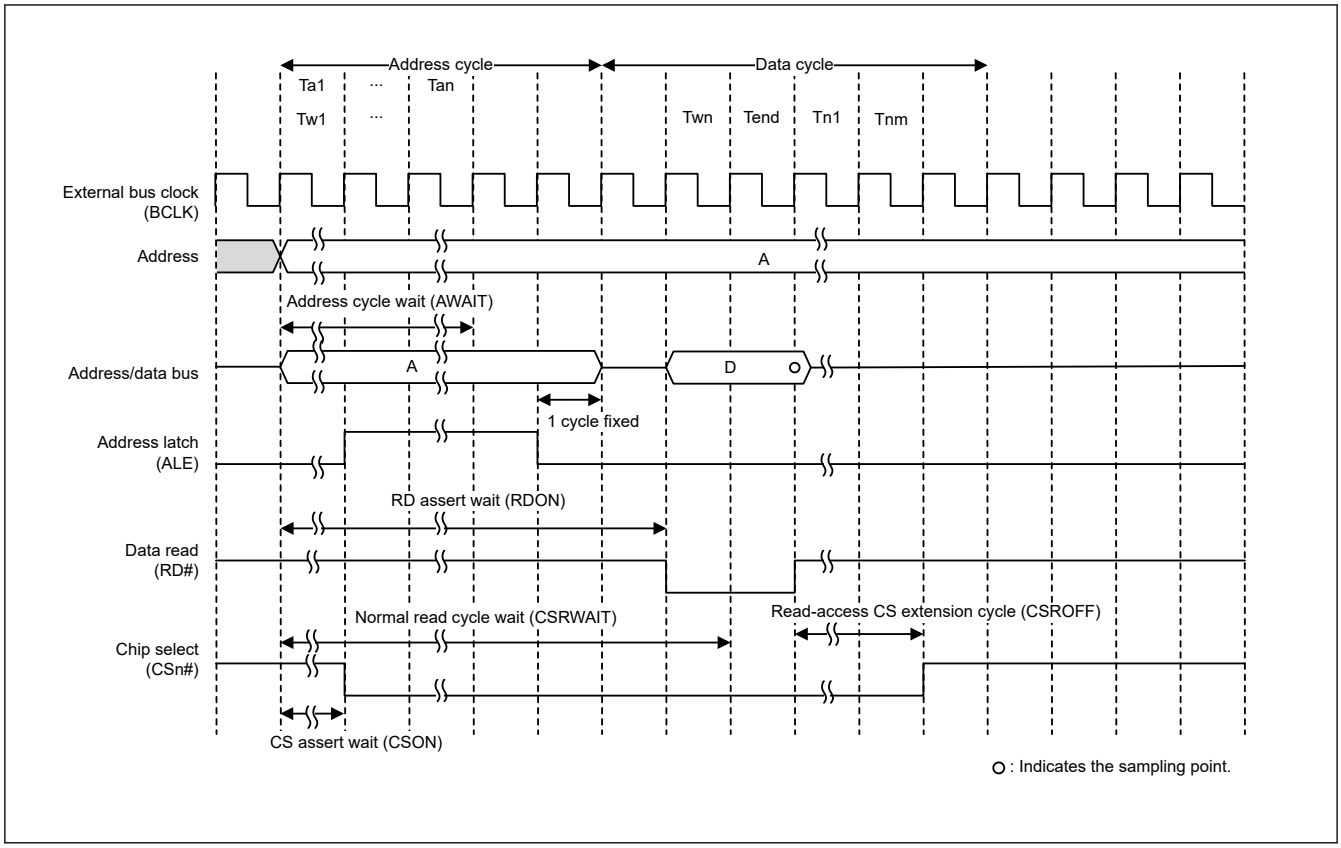


Figure 14.21 Example of read access operation with address/data multiplexed I/O interface (n = 0 to 7)

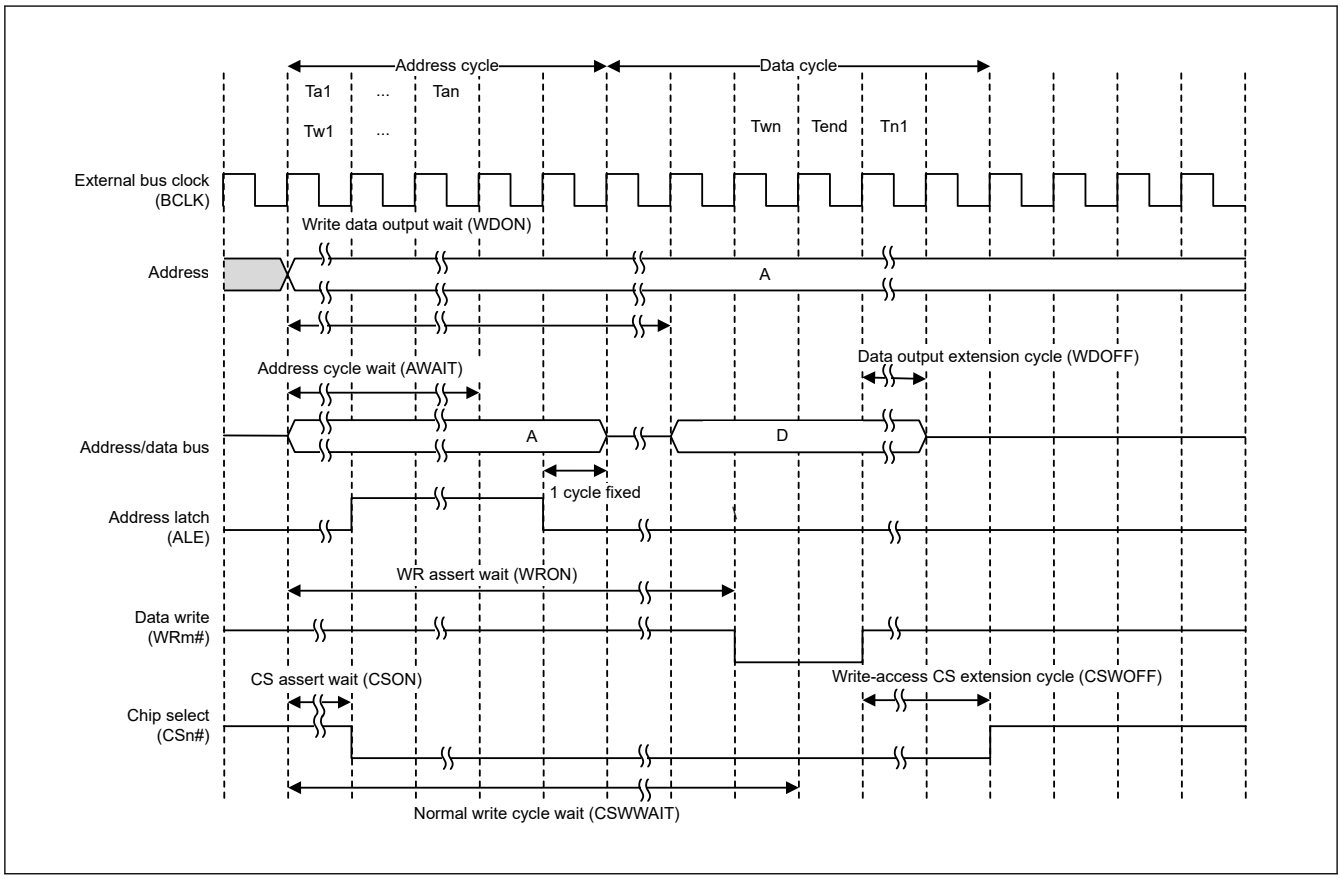


Figure 14.22 Example of write access operation with address/data multiplexed I/O interface (n = 0 to 7, m = 0 to 3)

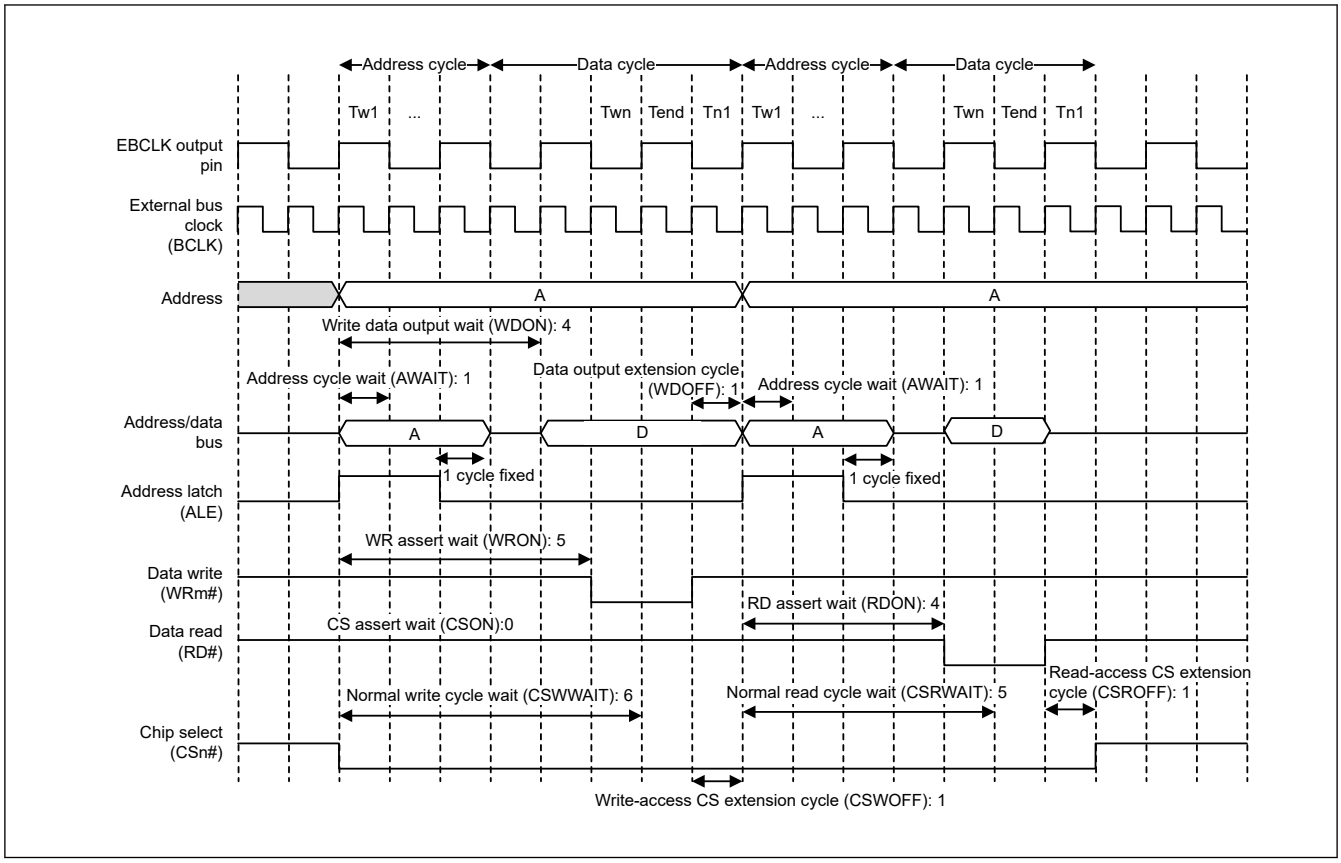


Figure 14.23 Example of bus timing with address/data multiplexed I/O interface (n = 0 to 7, m = 0 to 3)

14.5.3 External Wait Function

Wait cycles can be extended by the WAIT signal beyond the length of the normal access cycle wait specified in the CSRWAIT[4:0] and CSWWAIT[4:0] bits in CSnWCR1, and the page access cycle wait specified in the CSPRWAIT[2:0] and CSPWWAIT[2:0] bits in CSnWCR1.

When external wait is enabled (EWENB = 1 in CSnMOD), wait cycles are inserted while the WAIT signal is held low. When external wait is disabled (EWENB = 0 in CSnMOD), the WAIT signal has no effect. All wait cycles specified in CSnWCR1 are inserted independently of the WAIT signal.

(1) Normal Access

Sampling of the WAIT signal begins on completion of the wait cycle (Tend) specified in CSnWCR1. The bus cycle is extended while the WAIT signal is held low. The wait cycle ends (Tend) at the next cycle after the WAIT signal goes high.

(2) Page Access

The first access operation is the same as the normal access operation. Sampling of the WAIT signal begins on completion of the wait cycle (Tend) specified in the CSnWCR1 register. The bus cycle is extended while the WAIT signal is held low. The wait cycle (Tend) ends at the next cycle after the WAIT signal goes high.

For the second and subsequent accesses, sampling of the WAIT signal begins on completion of the page access wait cycle (Tend). The page access wait cycle is extended while the WAIT signal is held low, and ends (Tend) at the next cycle after the WAIT signal goes high.

Figure 14.24 to Figure 14.27 show examples of external wait insertion timing with the separate bus interface.

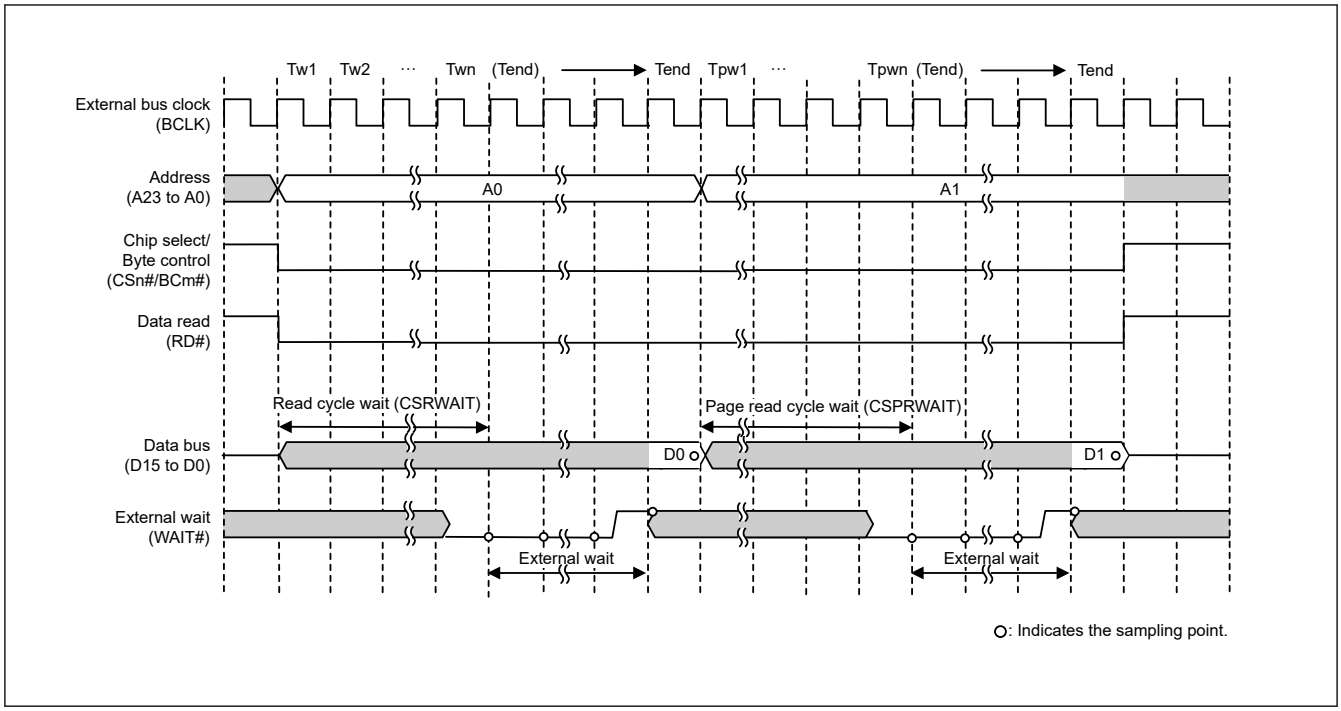


Figure 14.24 Example external wait timing for page read access to 16-bit bus space (when 1/1 BCLK is selected with the BCLK Pin Output Select bit) (n = 0 to 7, m = 0, 1)

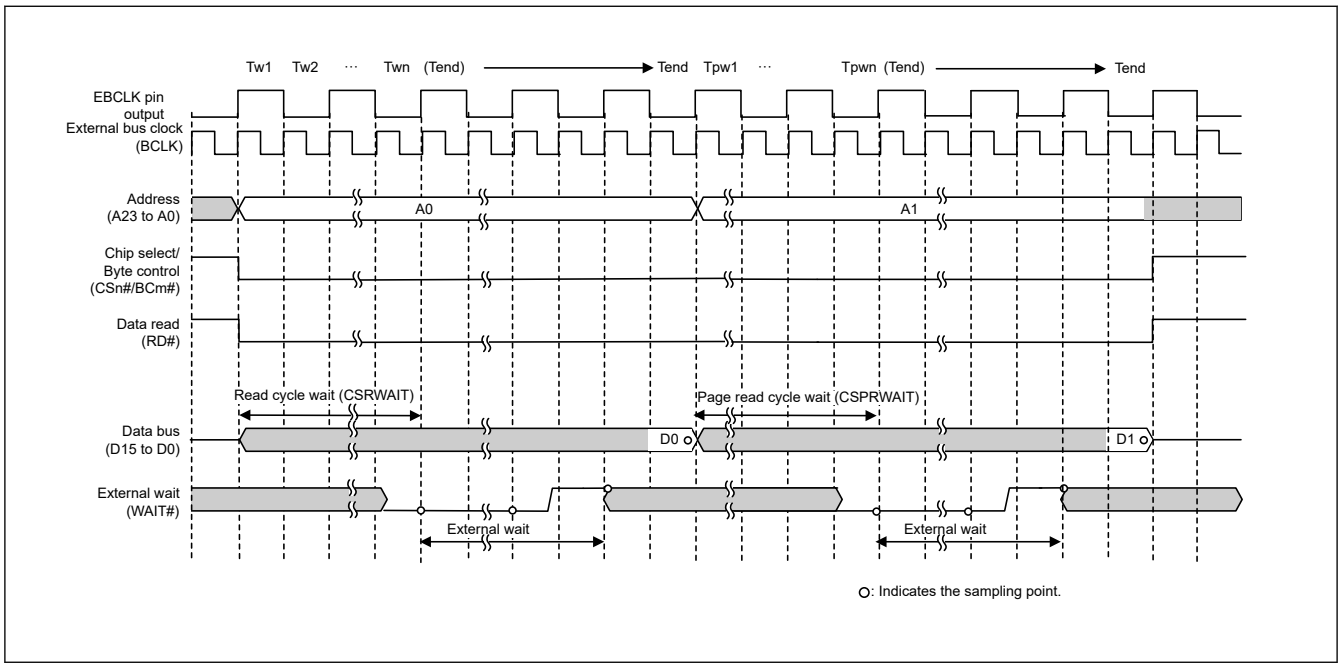


Figure 14.25 Example external wait timing for page read access to 16-bit bus space (when 1/2 BCLK is Selected with the EBCLK Pin Output Select bit) (n = 0 to 7, m = 0, 1)

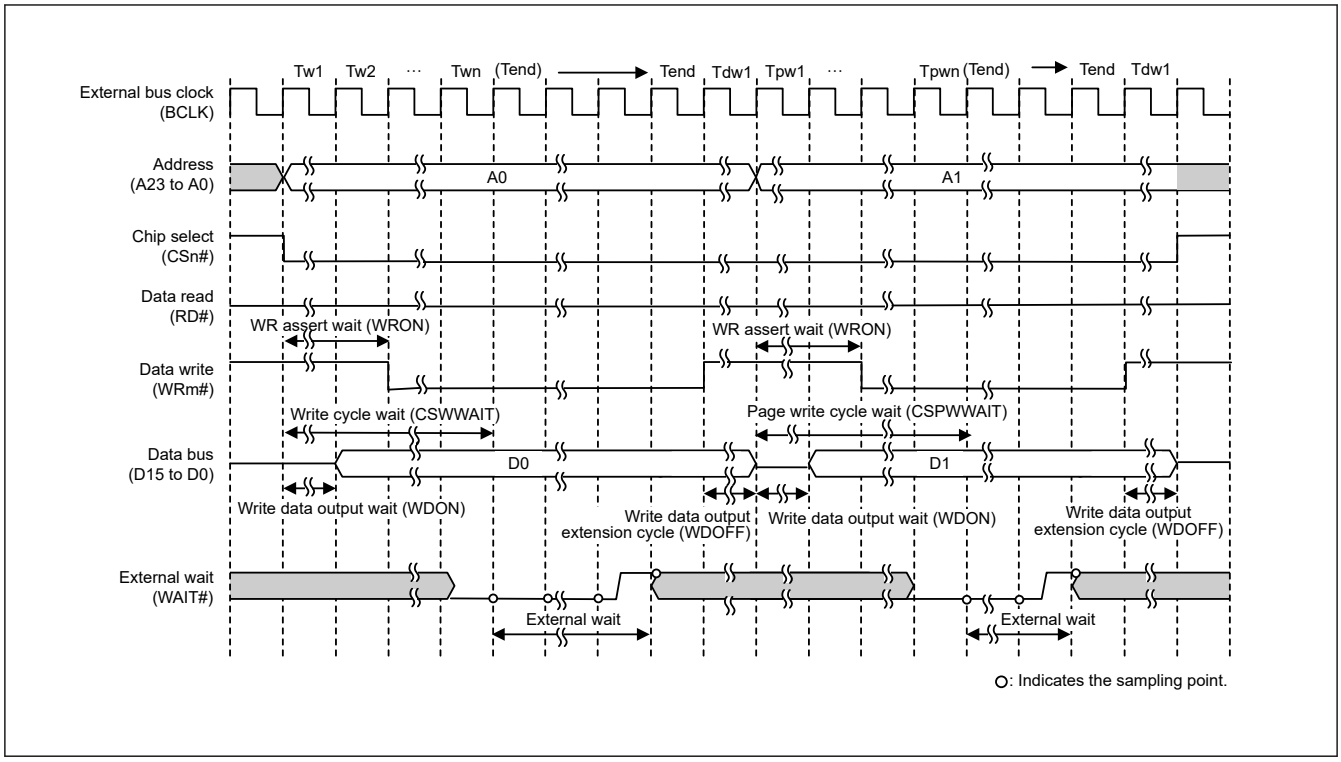


Figure 14.26 Example external wait timing for page write access to 16-bit bus space in byte strobe mode (when 1/1 BCLK is selected with the BCLK Pin Output Select bit) (n = 0 to 7, m = 0, 1)

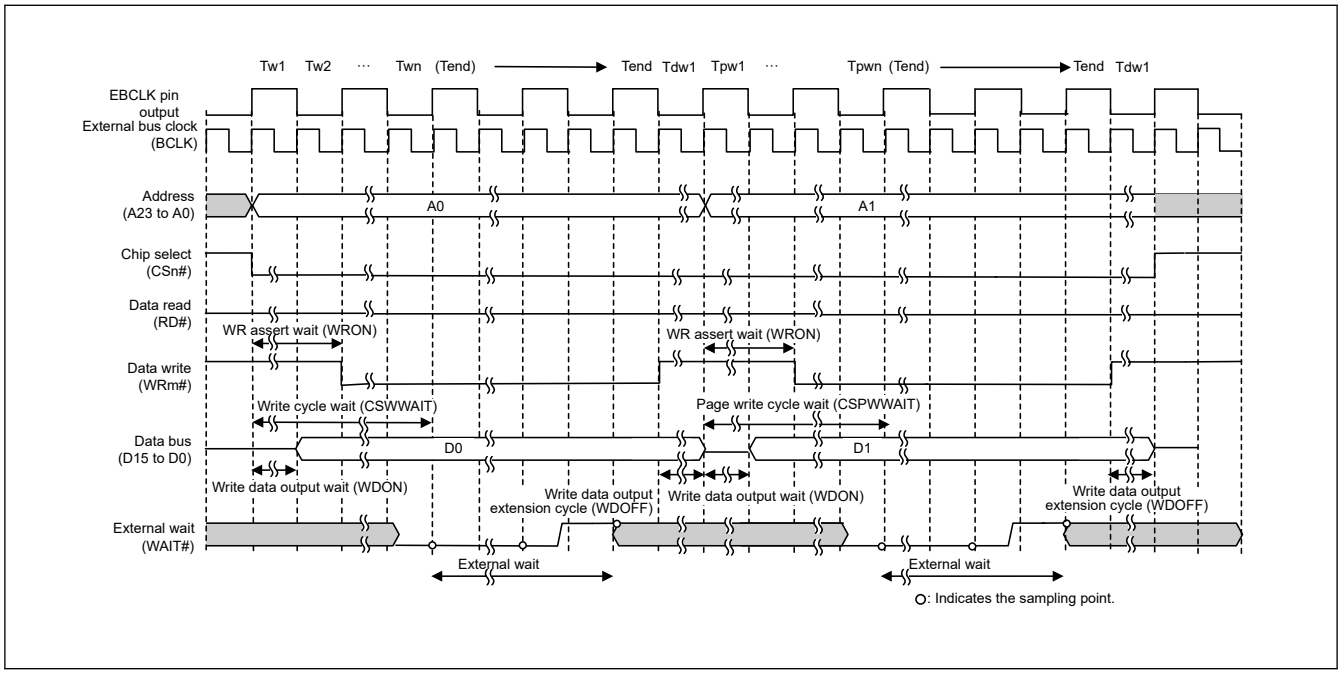


Figure 14.27 Example external wait timing for page write access to 16-bit bus space in byte strobe mode (when 1/2 BCLK is selected with the EBCLK Pin Output Select bit) (n = 0 to 7, m = 0, 1)

(3) Address/Data Multiplexed I/O Interface

In a data cycle with the address/data multiplexed I/O interface, programmed waits and pin waits using the WAIT pin can be inserted in the same way as that with the separate bus interface.

Address cycles are not affected by the wait control settings. Figure 14.28 shows an example of external wait insertion timing with the address/data multiplexed I/O interface.

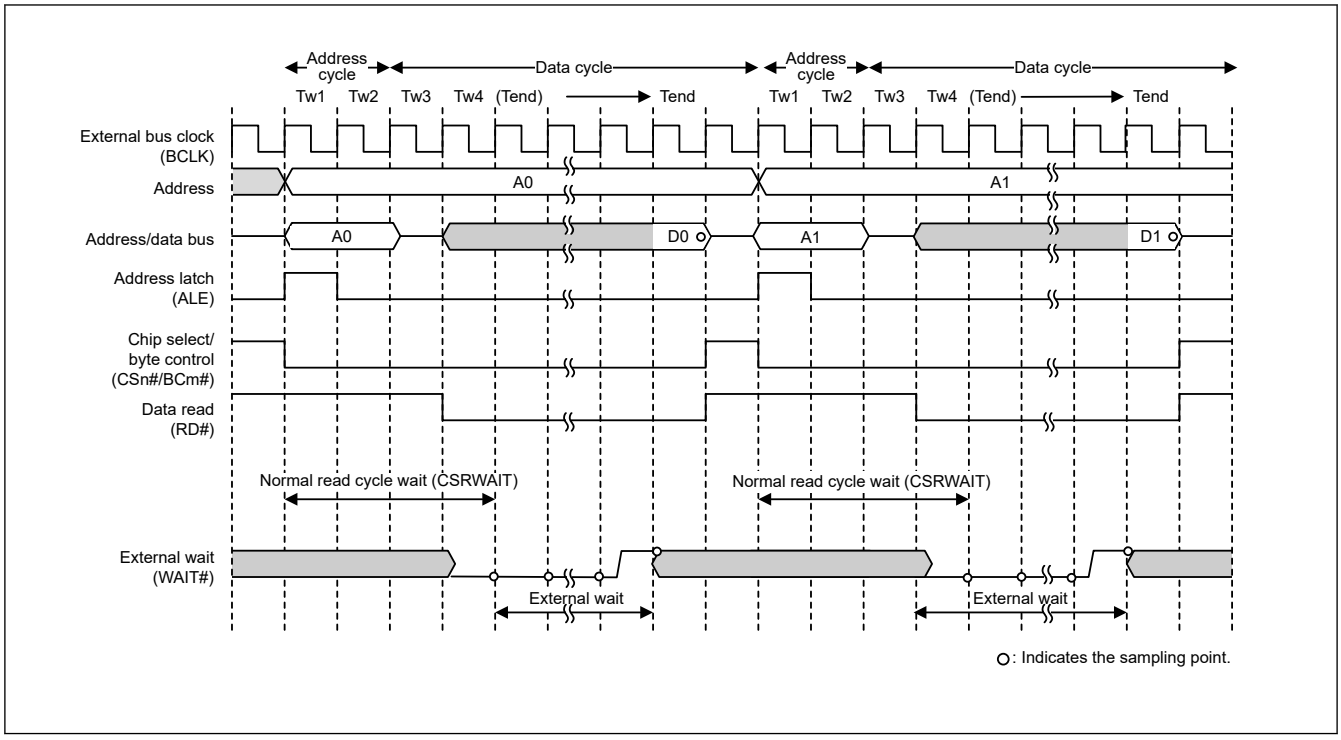


Figure 14.28 Example external wait insertion timing with address/data multiplexed I/O interface (n = 0 to 7, m = 0, 1)

14.5.4 Insertion of Recovery Cycles

Recovery cycles can be inserted between consecutive rounds of external bus access by setting the Recovery Cycle Insertion Enable bit in CSRECEN to 1.

The number of recovery cycles to be inserted after read cycles and write cycles can be independently set for each area using CSnREC. When the preceding bus cycle is a write access, the number of write recovery cycles must be set with the WRCV[3:0] bits for the associated area. When the preceding bus cycle is a read access, the number of read recovery cycles must be set with the RRCV[3:0] bits for the associated area. For example, when a CS1 read access occurs after a CS0 read access, the number of recovery cycles to be inserted between them is set in the RRCV[3:0] bits in CS0REC.

Recovery cycle insertion can be enabled or disabled with RCVENi (i = 0 to 7) in CSRECEN when the preceding bus access is a separate bus access, and with RCVENMj (j = 0 to 7) when the preceding bus access is an address/data multiplexed bus access.

Recovery cycles can be inserted on any of the following conditions:

- After a read access to the external bus, a read access is made to the external bus in the same area.
- After a read access to the external bus, a read access is made to the external bus in a different area.
- After a read access to the external bus, a write access is made to the external bus in the same area.
- After a read access to the external bus, a write access is made to the external bus in a different area.
- After a write access to the external bus, a read access is made to the external bus in the same area.
- After a write access to the external bus, a read access is made to the external bus in a different area.
- After a write access to the external bus, a write access is made to the external bus in the same area.
- After a write access to the external bus, a write access is made to the external bus in a different area.

The recovery cycle starts at the end of the preceding bus cycle, for example when the CSn signal (n = 0 to 7) is negated. A high-level period of the CSn signal is inserted for the specified recovery cycle period starting from this point.

In the fastest case, the CSn signal for the next round of bus access is asserted immediately after the end of the recovery cycles. Even if the next request for access to an external address space is generated during the recovery period, the next access over the external bus starts immediately after the end of the recovery cycles.

When two or more external bus access cycles are required for a single transfer request from a bus master, and the recovery cycle insertion condition is satisfied, recovery cycles are also inserted between these bus access cycles. However, when page read access is enabled (CSnMOD.PRENB = 1) or page write access is enabled (CSnMOD.PWENB = 1), recovery cycles are not inserted except after the last bus access cycle of the transfer, even if the recovery cycle insertion condition is satisfied. See Figure 14.32.

Similarly, during normal access with page access enabled, recovery cycles are not inserted between bus access cycles but only after the last bus access cycle of the transfer. With the address/data multiplexed I/O interface, when the recovery cycle insertion condition is satisfied, recovery cycles are inserted between bus access cycles regardless of the page access enable setting.

Figure 14.29 to Figure 14.32 show examples of recovery cycle insertion with the separate bus interface.

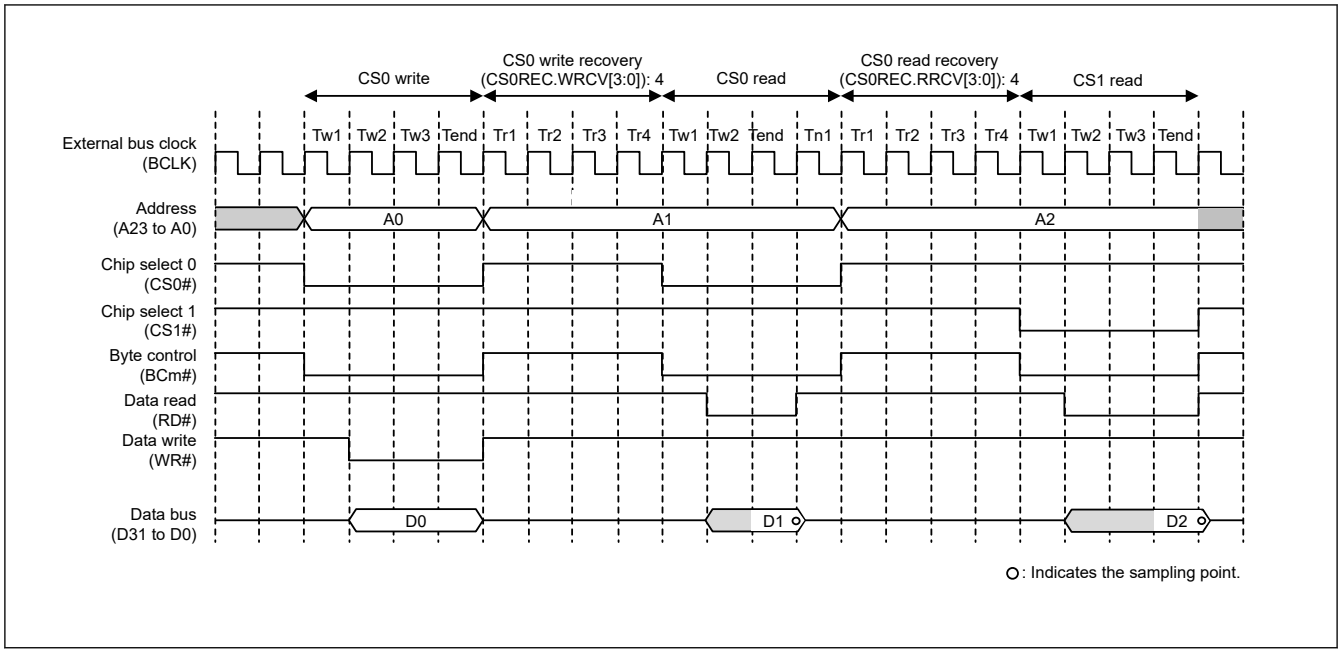


Figure 14.29 Example recovery cycle insertion with separate bus interface (m = 0 to 3)

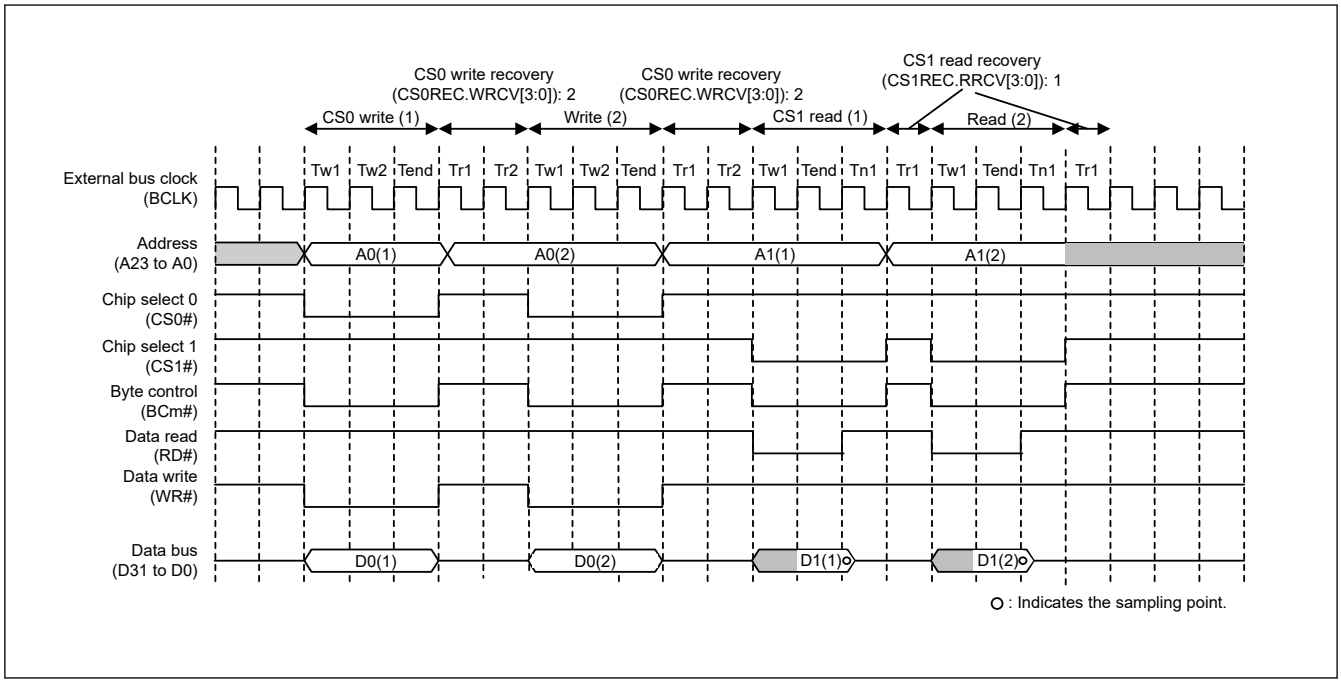


Figure 14.30 Example recovery cycle insertion when bus access is split, with separate bus interface and normal access (m = 0 to 3)

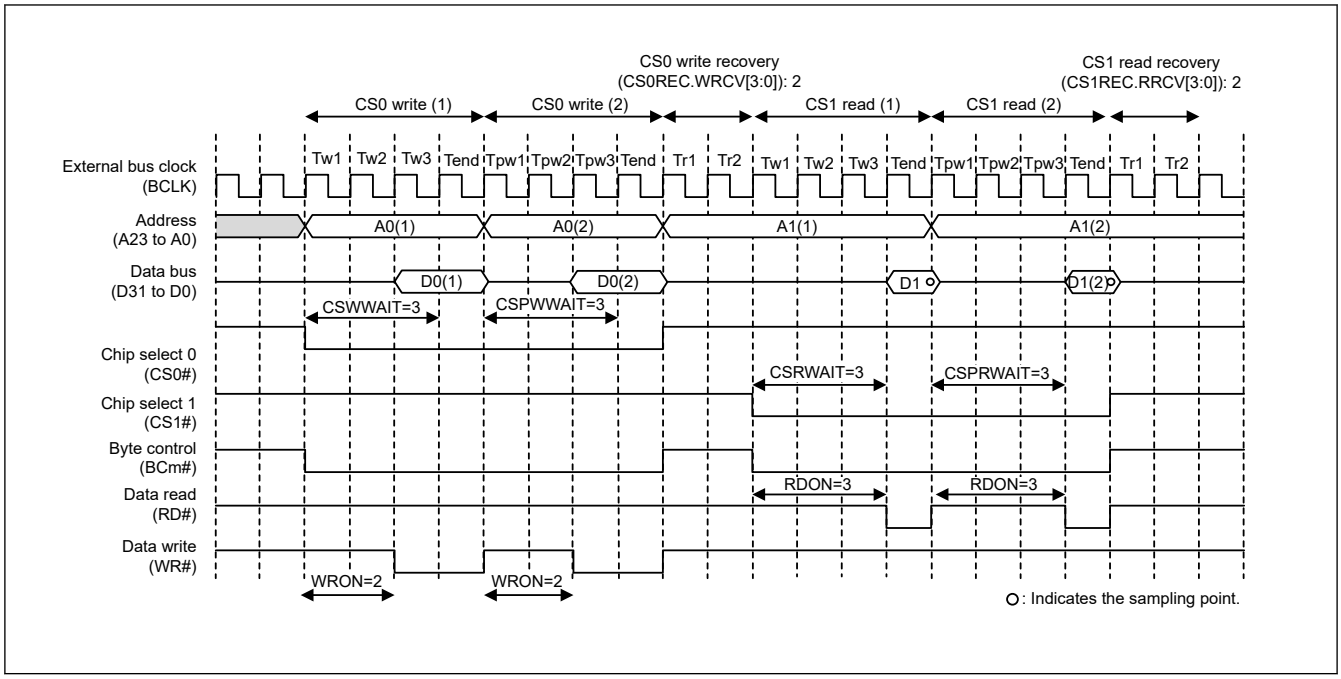


Figure 14.31 Example recovery cycle insertion when bus access is split, with separate bus interface and page access (m = 0 to 3)

Figure 14.32 shows an example operation when BCLK/2 is selected as the frequency division in the EBCLK Pin Output Select bit.

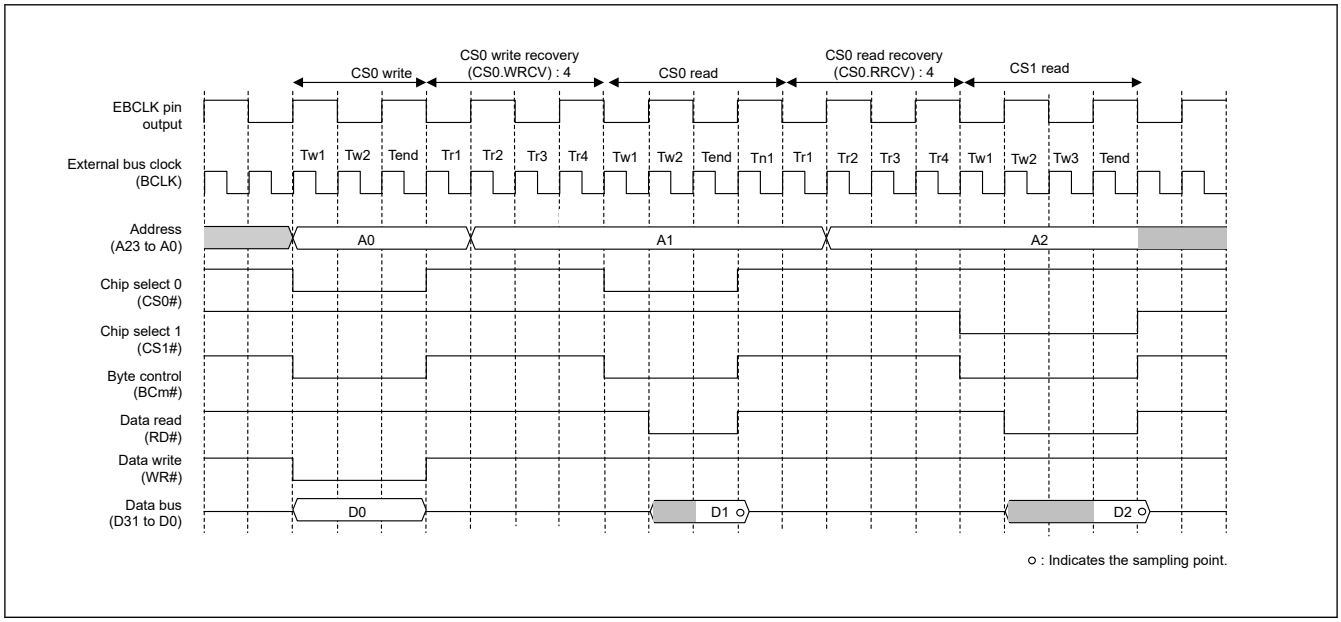


Figure 14.32 Example operation for recovery cycles when BCLK/2 is selected in the EBCLK Pin Output Select bit, with normal access through a separate bus interface (m = 0 to 3)

With the address/data multiplexed I/O interface, recovery cycles are inserted in the same way as that with the separate bus interface. Figure 14.33 and Figure 14.34 show examples of recovery cycle insertion with the address/data multiplexed I/O interface.

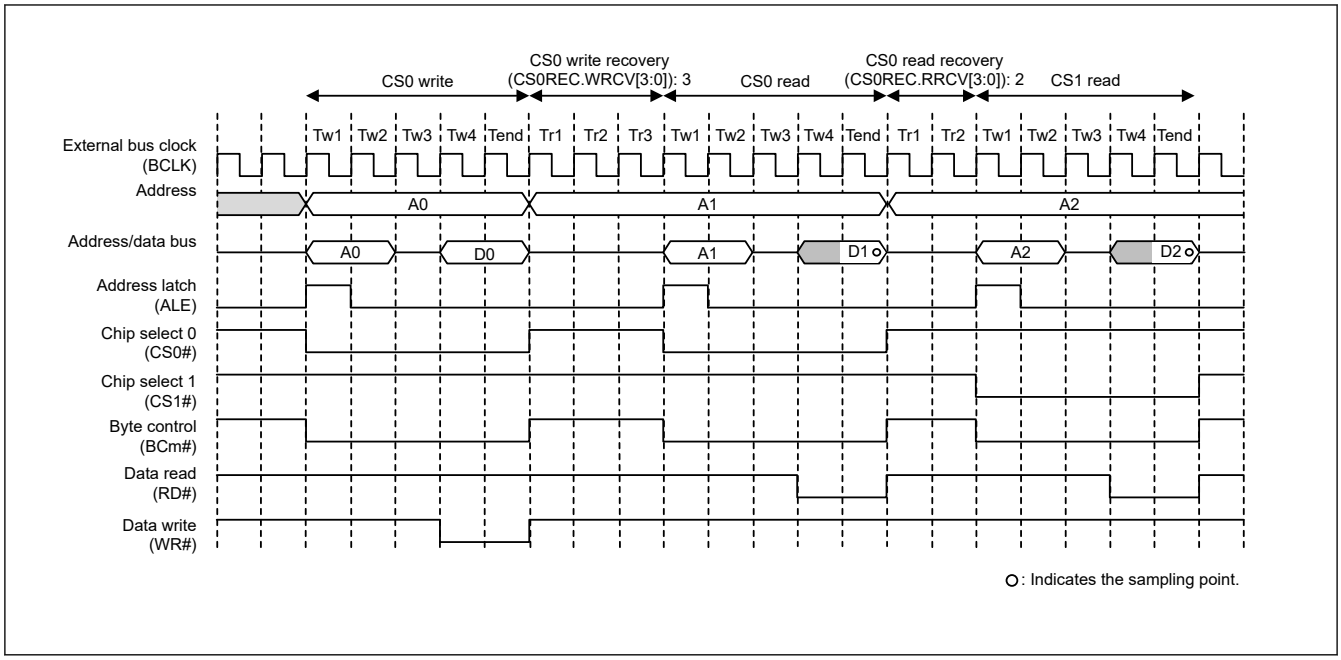


Figure 14.33 Example of recovery cycle insertion with address/data multiplexed I/O interface (m = 0, 1)

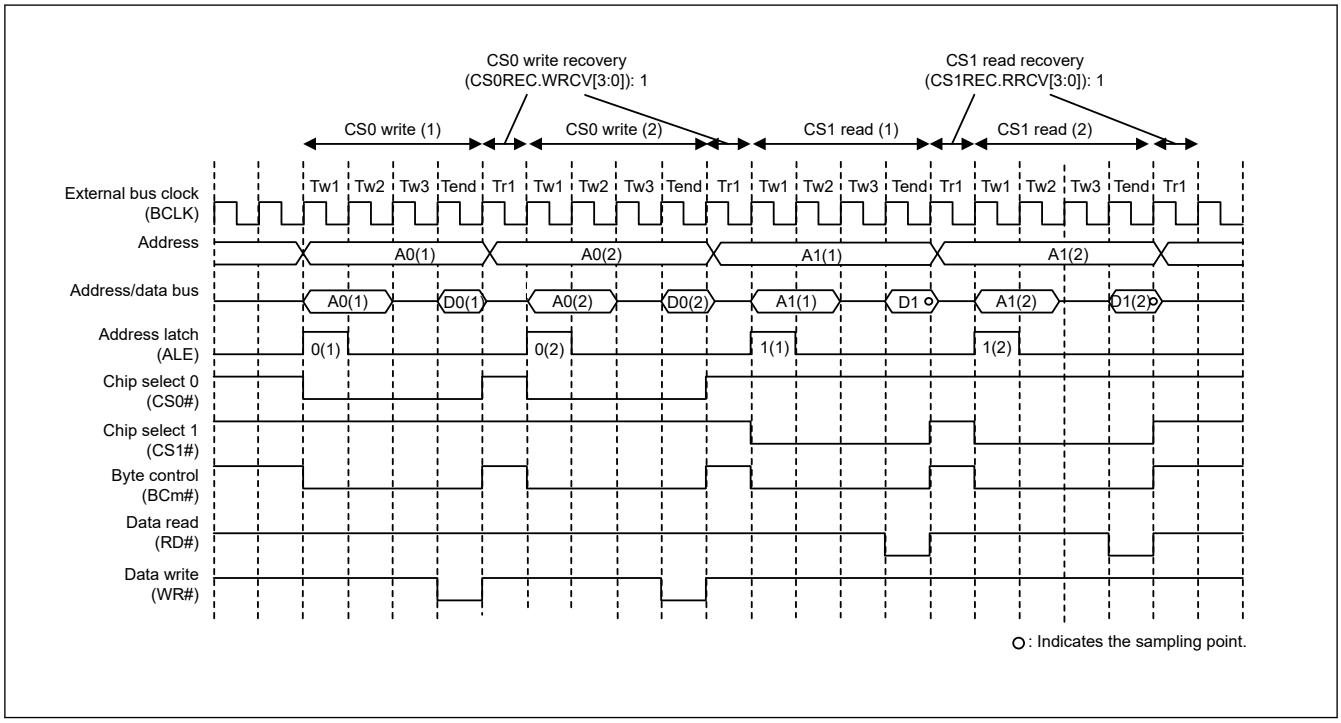


Figure 14.34 Example of recovery cycle insertion when a bus access is split with address/data multiplexed I/O interface (m = 0, 1)

14.5.5 No Access State

When no external address space is accessed, CSn, BCn, WRn, and RDn signals are high, ALE signal is low, and D31 to D00 are in the high-impedance state.

14.5.6 Write Buffer Function (External Bus)

In write access, the main bus is released by writing data to the write buffer before the access is complete. This allows the next round of bus access to start. However, if the next access is to an external address space or to a register of the external bus controller, it is suspended until the external bus operations already in progress are complete.

Figure 14.35 shows an example of operation when the write buffer function is in use. When this function is in use, if the next operation after an external write is an internal access, the internal access is executed in parallel with the external write, for example without waiting for completion of the latter operation.

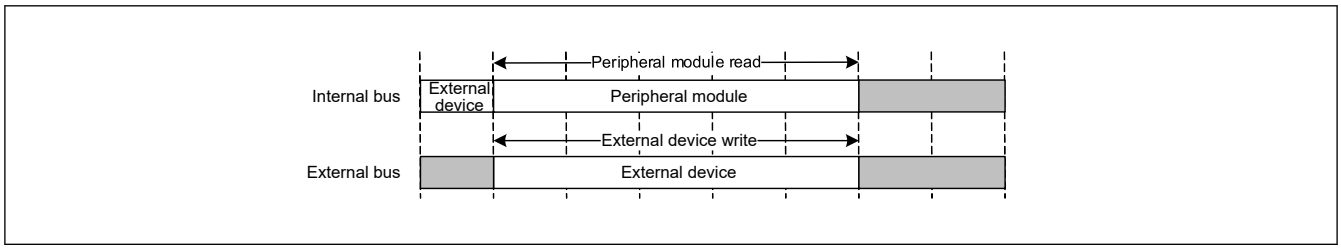


Figure 14.35 Example operation when the write buffer function is in use

14.5.7 Limitations

(1) Constraints on using separate bus interface

Table 14.31 lists the constraints that apply to bits in the CSn Wait Control Register 1 (CSnWCR1) and CSn Wait Control Register 2 (CSnWCR2) when normal and page accesses occur.

Even if the Page Read Access Enable bit or Page Write Access Enable bit in the CSn Mode Register is set to enable (CSnMOD.PRENB = 1 or CSnMOD.PWENB = 1), the first page access or access that does not fall within the scope of a page access is a normal access operation. Because of this, constraints on normal access must be satisfied.

Table 14.31 Constraints on normal access and page access

Limitations at the time of normal access		Limitations at the time of page access	
Reading	**Writing**	**Reading**	**Writing**
CSON[2:0] ≤ CSRWAIT RDON[2:0] ≤ CSRWAIT CSON[2:0] ≤ RDON	1 ≤ WDON[2:0] CSON[2:0] ≤ CSWWAIT WRON[2:0] ≤ CSWWAIT WDON[2:0] ≤CSWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON	CSON[2:0] ≤ CSPRWAIT RDON[2:0] ≤ CSPRWAIT CSON[2:0] ≤ RDON	1 ≤ WDON[2:0] CSON[2:0] ≤ CSPWWAIT WRON[2:0] ≤ CSPWWAIT WDON[2:0] ≤ CSPWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON

Note: When 2 or more external bus access cycles are required for a single transfer request from a bus master, and the recovery cycle insertion condition is satisfied, with page read access enabled (CSnMOD.PRENB = 1) or page write access enabled (CSnMOD.PWENB = 1), recovery cycles are not inserted between bus access cycles and are inserted only after the last bus access cycle of the transfer.

(2) Constraints on using address/data multiplexed bus interface

In the address/data multiplexed I/O space, page accesses are invalid. If a page access setting is specified, the setting is ignored and the normal read or write operation is performed. When the address/data multiplexed I/O interface is set, the BSIZE[1:0] bits in CSnCR should not be set to the 32-bit bus space. If set, the operation cannot be guaranteed.

Table 14.32 Constraints at the time of normal access

Reading	Writing
CSON[2:0] ≤ CSRWAIT RDON[2:0] ≤ CSRWAIT CSON[2:0] ≤ RDON AWAIT[1:0] + 2 ≤ RDON CSON[2:0] ≤ AWAIT	CSON[2:0] ≤ CSWWAIT WRON[2:0] ≤ CSWWAIT WDON[2:0] ≤ CSWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON AWAIT[1:0] + 2 ≤ WRON AWAIT[1:0] + 2 ≤ WDON CSON[2:0] ≤ AWAIT

(3) Constraint on pin multiplexing between the A00 and BC0 functions

Setting the single-write strobe mode is prohibited in the 8-bit bus space.

(4) Constraints when BCLK/2 is selected in the EBCLK Pin Output Select bit

When 1/2 cycle of BCLK is selected in the EBCLK Pin Output Select bit, the external bus access cycle starts on the rising edge of the EBCLK pin output. However, when 2 or more external bus access cycles are generated for a single transfer request from a bus master, the second or subsequent external bus access cycle can start on the falling edge of the EBCLK pin output, depending on the wait cycle settings. Set the registers appropriately for the specifications of connected devices.

(5) Instruction code constraints

You must fix the instruction code to little-endian order.

14.6 SDRAM Area Controller Operation

This section describes how the SDRAM area controller (SDRAMC) is enabled and the SDRAM bus width is set, followed by a description of the SDRAMC operations, including read, write, auto-refresh, self-refresh, initialization sequence, and mode register settings.

14.6.1 Enabling/Disabling SDRAM Access and Setting the SDRAM Bus Width

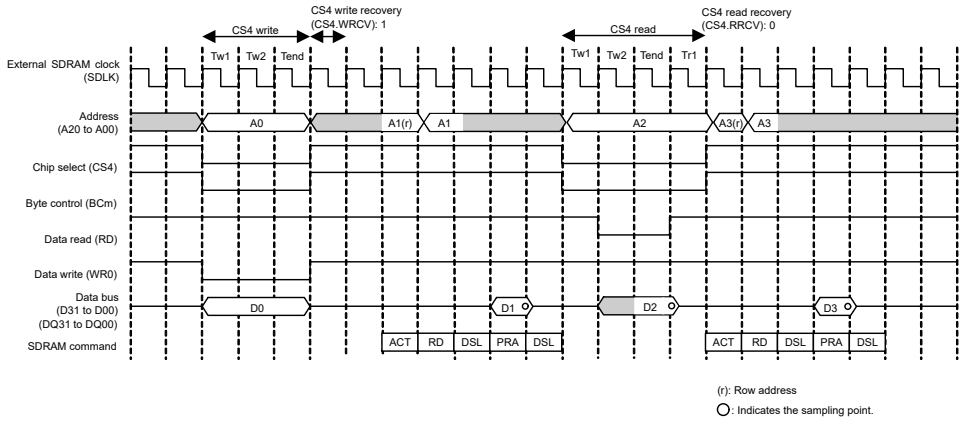
SDRAM access can be enabled or disabled using the SDC Control Register (SDCCR). The SDRAM bus width can also be set using SDCCR. The refresh operation is available even when the operation of the SDRAM address space is disabled, as long as self-refresh or auto-refresh is enabled.

14.6.2 No Access State

When no external address space is accessed, the SDCS, WE, RAS and CAS signals are high.

14.6.3 Insertion of Recovery Cycles

When access to the SDRAM area follows access to the CS area, data recovery cycles are inserted for the CS area controller (CSC). If the number of recovery cycles for the CSC is 0, the ACT command for the next SDRAM access is issued immediately after negation of CSn signal at the earliest. If the number of recovery cycles are not 0, the ACT command is issued 2 cycles after the specified recovery cycle period elapsed after negation of CSn signal at the earliest. Because no data conflicts can occur during access to the SDRAM area, there is no need to set data recovery cycles for the SDRAM (fixed to 0 cycle).



Note: When using SDRAM and CS Area access simultaneously.
- EBCLK cannot be used.
- CS0 to CS3 cannot be used.

Figure 14.36 Example of recovery timing for SDRAM access

14.6.4 Write Buffer Function

In write access, the main bus is released by writing data to the write buffer before access is complete. This allows the next round of bus access to start. However, if the next access is to an external address space or to a register of the external bus controller, it is suspended until the external bus operations already in progress are complete.

14.6.5 SDRAM Commands

To control the SDRAM, the SDRAMC issues a command for each bus cycle. Commands are defined by a combination of the SDCS, RAS, CAS, WE, CKE, and other signals. Table 14.33 lists the commands issued by the SDRAMC.

Table 14.33 SDRAMC commands

Name	Abbreviation	Command	SDCS	RAS	CAS	WE	CKE		BA1	BA0
							n-1	n		
DESL	DSL	Device deselect	H	x	x	x	H	x	x	x
ACTV	ACT	Bank active	L	L	H	H	H	x	V	V
READ	RD	Read	L	H	L	H	H	x	V	V
WRIT	WRI	Write	L	H	L	L	H	x	V	V
PALL	PRA	All bank precharge	L	L	H	L	H	x	x	x
REF	RFA	Auto-refresh	L	L	L	H	H	x	x	x
MRS	MRS	Mode register set	L	L	L	L	H	x	L	L
SELF	RFS	Self-refresh entry	L	L	L	H	H	L	x	x
SELFX	RFX	Self-refresh end	H	x	x	x	L	H	x	x

Note: H = High level, L = Low level, V = Valid, x = Don't care.
n = Command issue cycle, n – 1 = 1 cycle before the command is issued.

14.6.6 Conditions for Setting the SDRAMC Registers

The SDRAMC registers must only be modified when all the conditions shown in Table 14.34 are satisfied.

Table 14.34 Conditions for register modification

Function or operation	Registers	Conditions
Self-refresh	SDSELF*1	• SDRAM access is disabled (SDCCR.EXENB = 0*2) • Auto-refresh operation is enabled (SDRFEN.RFEN = 1).
Auto-refresh	SDRFCR	Self-refresh operation is disabled (SDSELF.SFEN = 0)
	SDRFEN	• SDRAM access is disabled (SDCCR.EXENB = 0*2) • Self-refresh operation is disabled (SDSELF.SFEN = 0).
Initialization sequence	SDIR*1	SDICR is not set yet, and the same conditions as for SDICR modification are satisfied
	SDICR*1	• SDRAM access is disabled (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled (SDRFEN.RFEN = 0) • Self-refresh operation is disabled (SDSELF.SFEN = 0).
Address register	SDADR	• SDRAM access is disabled (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled (SDRFEN.RFEN = 0) • Self-refresh operation is disabled (SDSELF.SFEN = 0).
Timing register	SDTR	• Self-refresh operation is in progress (SDSELF.SFEN = 1) or • SDRAM access is disabled (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled (SDRFEN.RFEN = 0) • Self-refresh operation is disabled (SDSELF.SFEN = 0).
Mode register	SDMOD*1	• SDRAM access is disabled (SDCCR.EXENB = 0*2) • Self-refresh operation is disabled (SDSELF.SFEN = 0).

Note 1. Before modifying this register, confirm that all the status bits in SDSR are 0.
Note 2. After writing 0 to the EXENB bit, confirm that it is cleared to 0.

14.6.7 Self-Refresh

Transition to or recovery from self-refresh mode is controlled with the SDRAM Self-Refresh Control Register (SDSELF). Immediately before the transition to self-refresh mode, an auto-refresh operation is performed. In self-refresh mode, the CKE signal is low. Immediately after recovery from self-refresh mode, the auto-refresh cycle starts.

Figure 14.37 and Figure 14.38 show timing examples of the transition to and recovery from self-refresh mode.

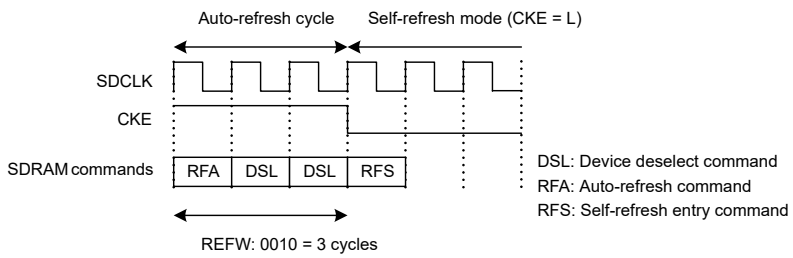


Figure 14.37 Example timing for transition to self-refresh mode when SDRFCR.REFW[3:0] = 0010 (3 cycles)

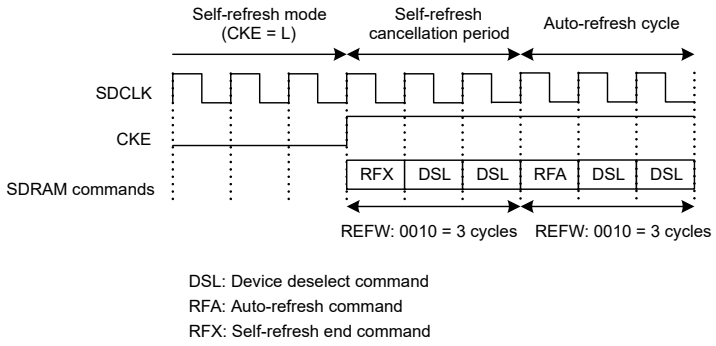


Figure 14.38 Example timing for recovery from self-refresh mode

(1) Self-refresh in Software Standby mode

When invoking self-refresh in Software Standby mode, first follow the procedure shown in section 14.6.12.2. Procedure for transitioning to and recovering from self-refresh mode. Next, set up the transition to Software Standby mode. In this mode, set the Output Port Enable bit (OPE) in the Standby Control Register (SBYCR) to 1 to hold the output state of the address bus and bus control signals.

After canceling Software Standby mode, follow the procedure shown in section 14.6.12.2. Procedure for transitioning to and recovering from self-refresh mode. For details on invoking and canceling Software Standby mode, see section 10, Low Power Modes.

(2) Self-refresh in Deep Software Standby mode

Deep Software Standby mode is invoked from within Software Standby mode. On this transition, the pin states remain unchanged. Therefore, invoking of self-refresh in Deep Software Standby mode can be handled the same as for Software Standby mode with one additional setting. You must also set the I/O Port Keep bit (IOKEEP) in the Deep Software Standby mode Control Register (DPSBYCR) to 1.

Because the SDRAMC is reset internally when Deep Software Standby mode is canceled, the SDRAM control registers must be set again. After canceling Software Standby mode, follow the procedure in this section to cancel self-refresh.

Figure 14.39 shows self-refresh timing in Deep Software Standby mode. For details on invoking and canceling Deep Software Standby mode, see section 10, Low Power Modes.

To cancel self-refresh mode:

1. Set DPSBYCR.IOKEEP to 1 to keep the CKE signal output low in Deep Software Standby mode.
2. Start the clock supply to the SDRAMC.
3. Set the SDRAM control registers (SDCMOD, SDADR, and SDTR) again. These registers were initialized by an internal reset on entering Deep Software Standby mode.
4. Enable an auto-refresh operation by setting SDRFEN.RFEN to 1.
5. Check that all the status bits in SDSR are cleared to 0 and set SDSELF.SFEN to 1 to select self-refresh mode again.
6. Modify the port settings for the SDRAM interface.
7. Set SDCKOCR.SDCKOEN to 1 to start the clock supply to the SDRAM with the SDCLK pin.
8. Check that all the status bits in SDSR are cleared to 0 and set SDSELF.SFEN to 0 to cancel self-refresh mode.

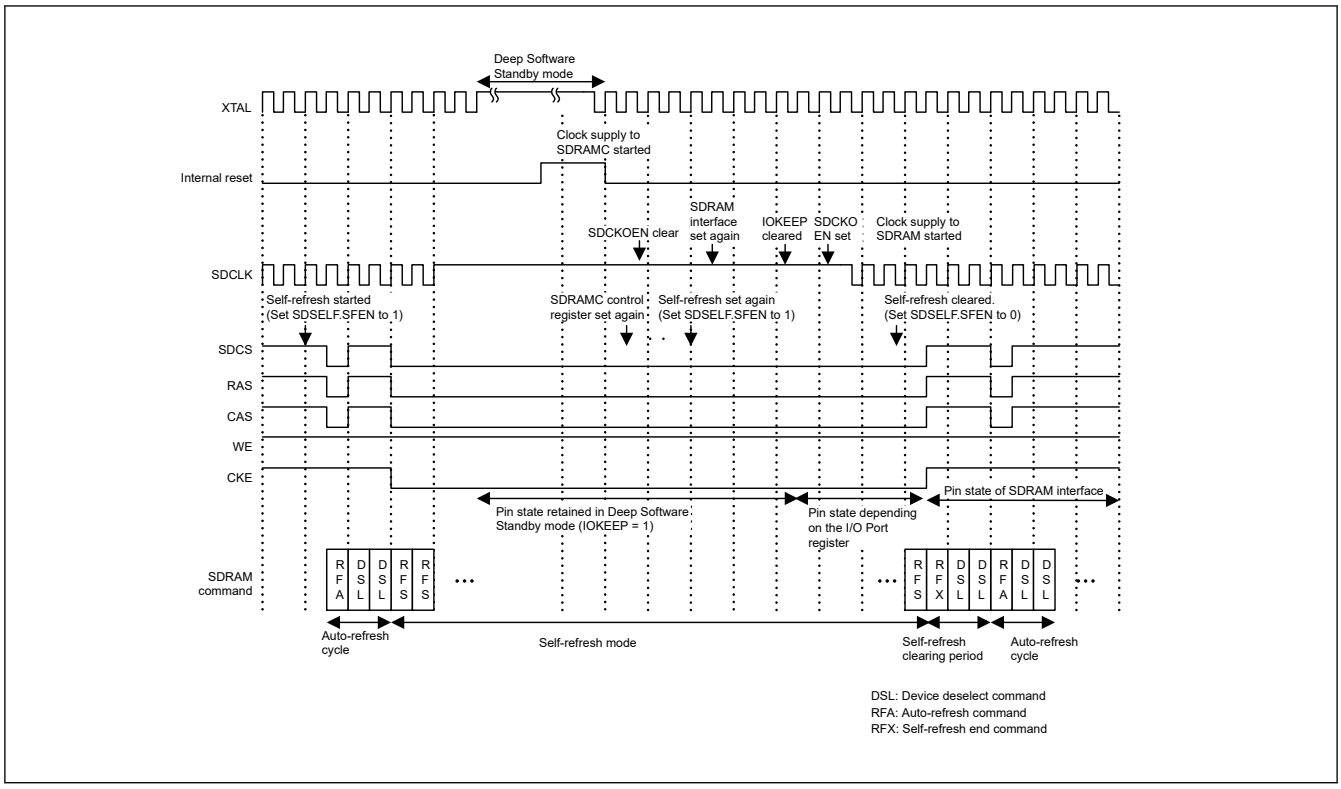


Figure 14.39 Example timing for self-refresh cycle in Deep Software Standby mode

14.6.8 Auto-Refresh

The auto-refresh cycle can be started by setting the Auto-Refresh Operation Enable bit (RFEN) in the SDRAM Auto-Refresh Control Register (SDRFEN) to 1. After the cycle starts, refresh requests are generated at fixed intervals determined by the refresh counter. However, because refresh requests are not accepted during read or write access, the auto-refresh cycle might be suspended.

If an SDRAM access and a refresh request are generated at the same time, the refresh request takes precedence. A CS area access and a refresh request can be made at the same if the SDCS, RAS, CAS, WE, and CKE signals, which are required for issuing the refresh command, are exclusively provided for SDRAM access.

The refresh counter is halted during a self-refresh operation. After recovery from the self-refresh mode, the auto-refresh cycle starts and the counter value is reset, resuming the counter operation.

Figure 14.40 shows a timing example of an auto-refresh cycle.

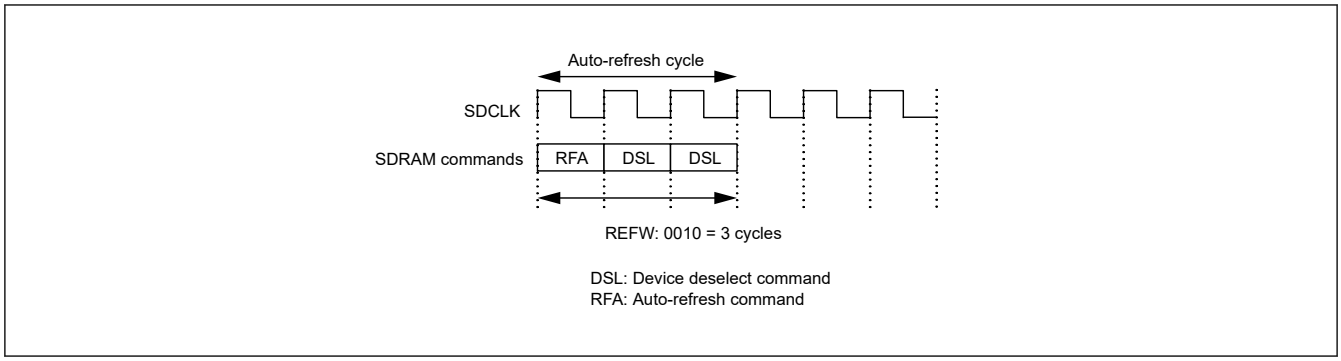


Figure 14.40 Example timing for auto-refresh cycle (1)

Figure 14.41 shows examples of operation when an auto-refresh request is generated during single access.

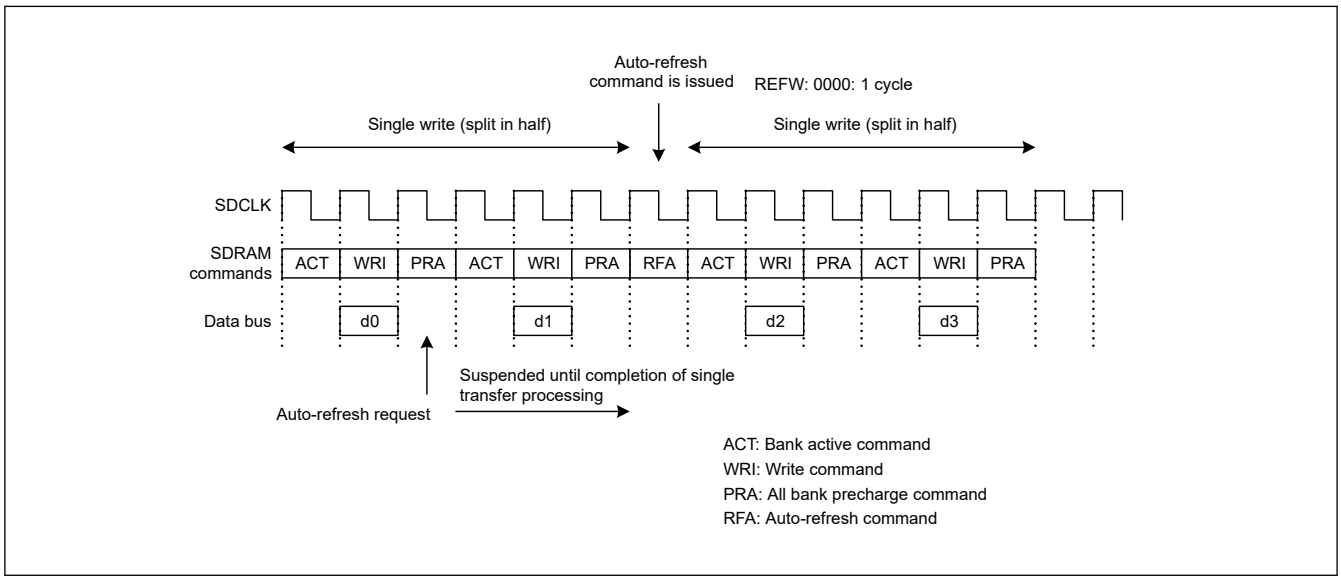


Figure 14.41 Example timing for auto-refresh cycle (2), when the auto-refresh request is made during single access

14.6.9 Initialization Sequencer

The SDRAMC has a sequencer to issue SDRAM initialization commands. After a reset, the initialization sequencer must be activated without fail. Operation is not guaranteed if the SDRAM is not initialized.

The SDRAM initialization sequencer issues an all-bank precharge command followed by auto-refresh commands n times, where n = 1 to 15. The SDRAM initialization sequence timing can be set using the SDRAM Initialization Register (SDIR). The SDRAM initialization sequence can be activated using the SDRAM Initialization Sequence Control Register (SDICR). These registers must be set only when the conditions listed in Table 14.34 are satisfied.

Figure 14.42 shows a timing example of the SDRAM initialization sequence. When the ARFC[3:0] bits in SDIR are set so that auto-refresh operation is performed two or more times, auto-refresh cycles are repeated in the initialization sequence accordingly.

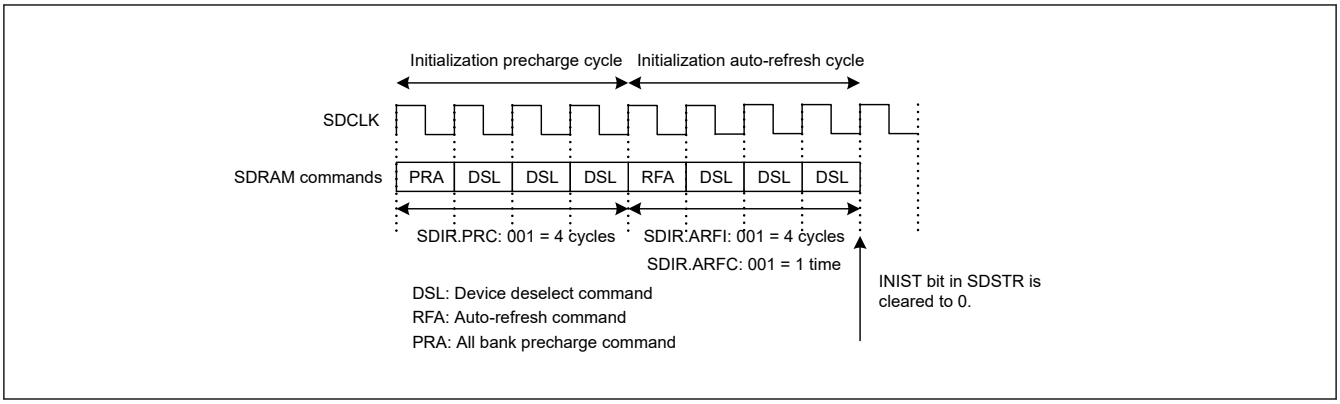


Figure 14.42 Example timing for SDRAM initialization sequence

14.6.10 Read/Write Access

The SDRAMC controls read/write access in the following two modes.

- Single access mode: the row address is output each time data is accessed
- Consecutive access mode: when the same row address is accessed consecutively, only the column address is changed after the row address is output, enabling quick data access.

Consecutive SDRAM access is enabled by setting the continuous access enable bit (BE) in SDRAM access mode register (SDAMOD) to 1. Consecutive SDRAM access targets burst transfers from bus master and transfers divided by the relationship between data size and SDRAM bus width.

Furthermore, setting the SDRAMC column-latency setting bits (CL[2:0]) in SDTR to 1 (CL = 1) in consecutive-access mode is prohibited, and operation is not guaranteed if this setting is made.

When the BE bit in SDAMOD is 0, single access is used in all transfers.

Note: WRAP (Wrapping Burst) of CPU does not result in consecutive SDRAM access.

14.6.10.1 Single Access

Figure 14.43 and Figure 14.44 show timing examples of single read and Figure 14.45 show timing examples of single write, respectively. The specific access timing depends on the SDRAM timing register (SDTR) settings. For details, see section 14.6.12.3. Timing register settings and access timing.

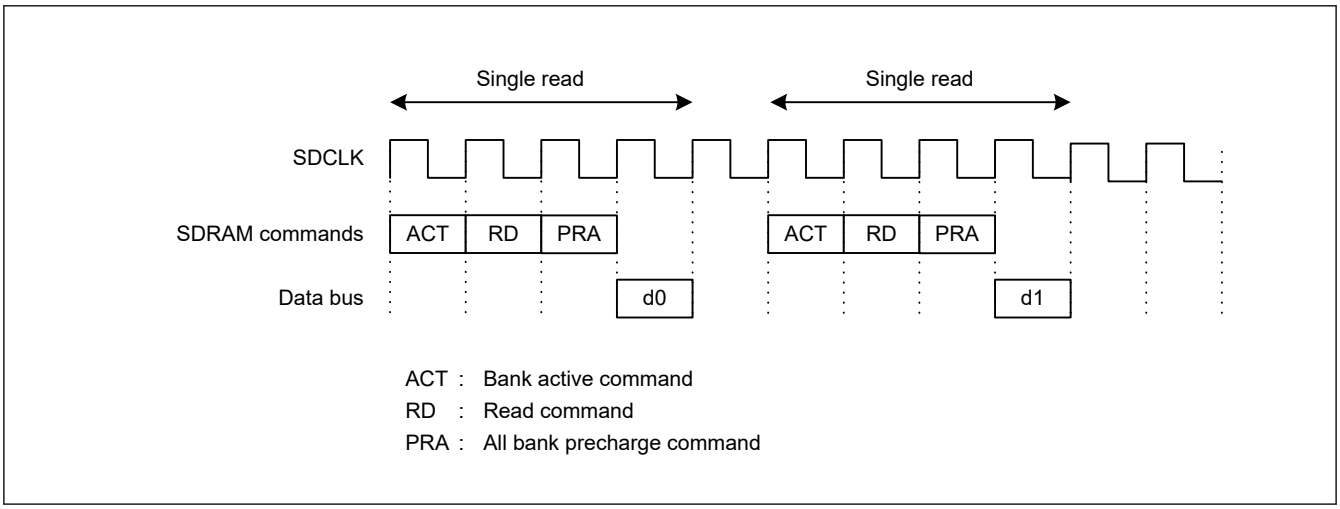


Figure 14.43 Timing Example of Single Read (SDTR.CL[2:0] = 010b: 2 Cycles)

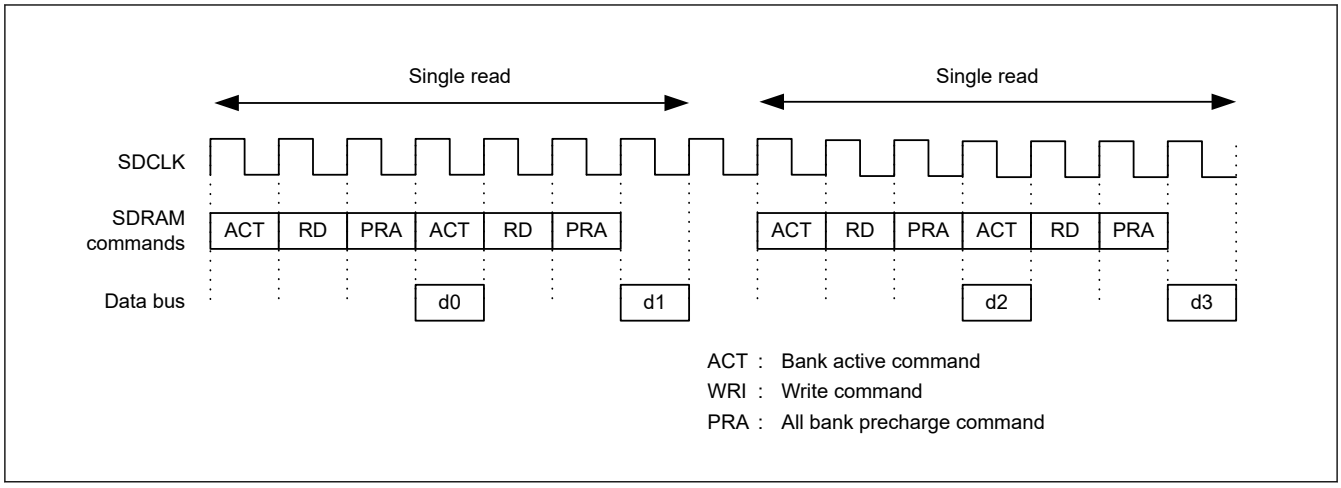


Figure 14.44 Timing Example of Single Read (Burst transfer from bus master with SDAMOD.BE = 0 and SDTR.CL[2:0] = 010b: 2 Cycles)

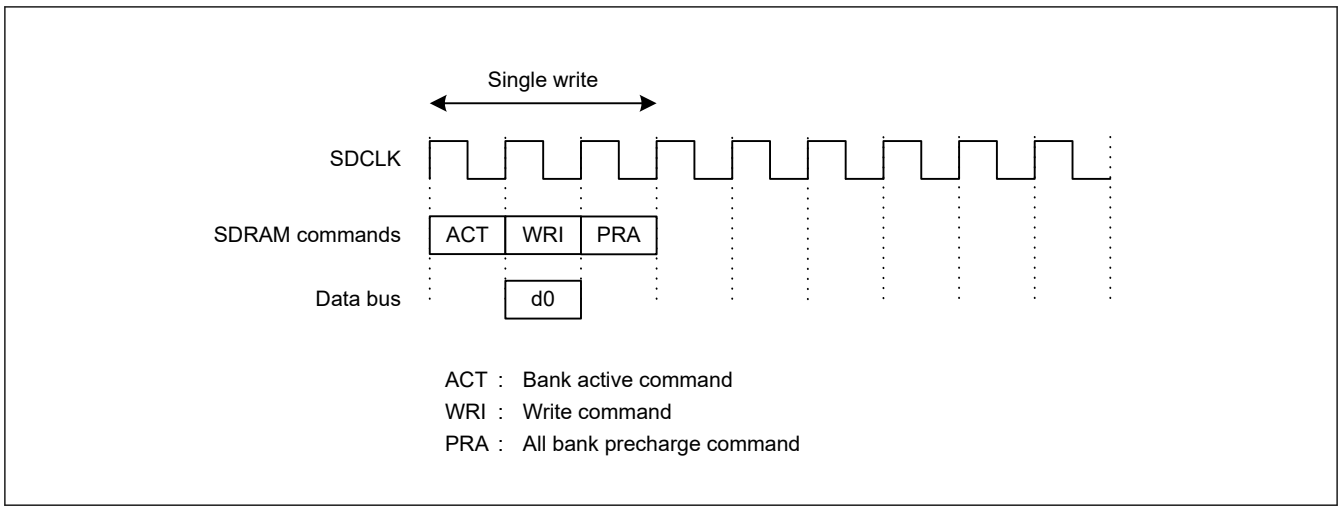


Figure 14.45 Timing Example of Single Write (when the Shortest Timing is Set)

14.6.10.2 Consecutive Access

Figure 14.46 and Figure 14.47 show timing examples of consecutive read and consecutive write for four data, respectively.

The specific access timing depends on the SDRAM timing register (SDTR) settings. For details, see section 14.6.12.3. Timing register settings and access timing.

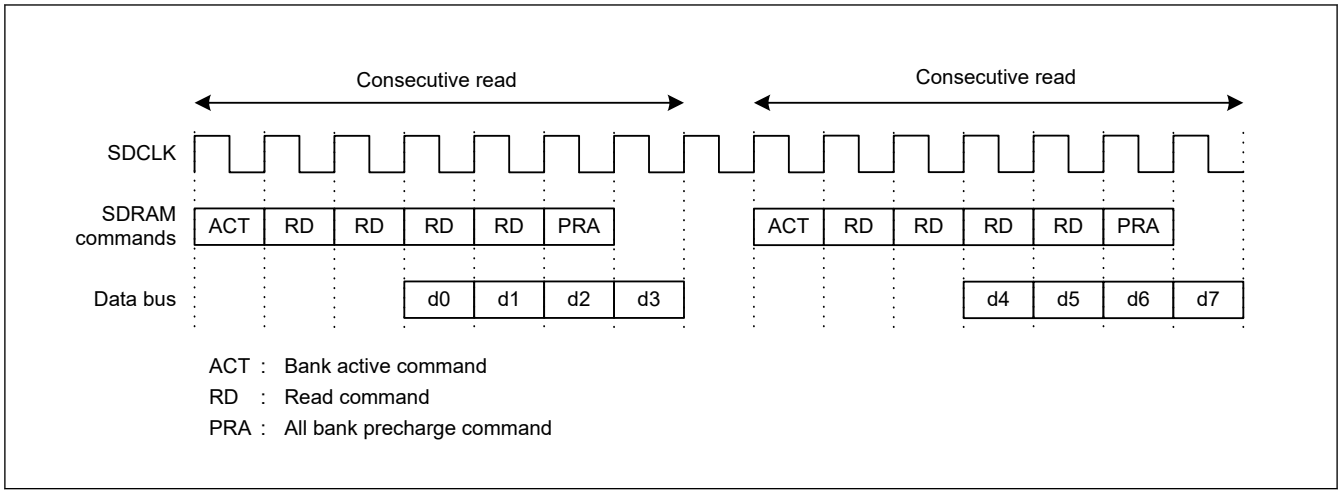


Figure 14.46 Timing Example of Consecutive Read (SDAMOD.BE = 1 and SDTR.CL[2:0] = 010b: 2 Cycles)

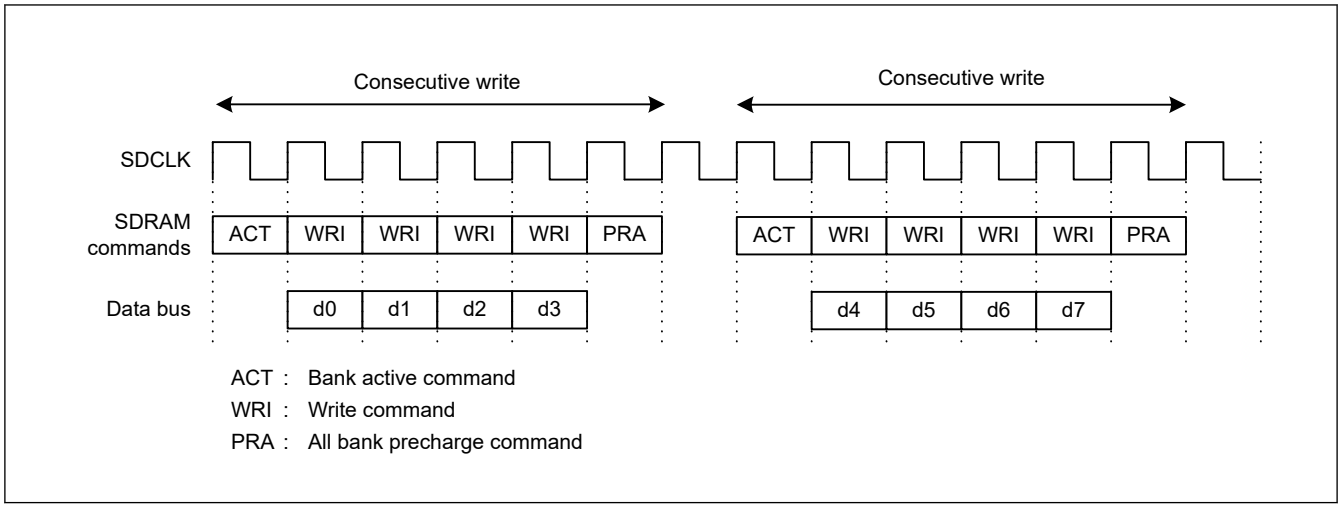


Figure 14.47 Timing Example of Consecutive Write (SDAMOD.BE = 1, when the Earliest Timing is Set)

14.6.11 Setting the Mode Register

Setting the SDRAM Mode Register (SDMOD) allows the mode register set command to be issued to the SDRAM and the value set in the MR[14:0] bits in SDMOD to be output to the lower bits of the address, specifically to A14 to A00 for 8-bit bus width, A15 to A01 for 16-bit bus width, and A16 to A2 for 32-bit bus width. Before setting the mode register, set the SDRAM Bus Width Select bits in the SDC Control Register (SDCCR.BSIZE[1:0]) to determine the data bus width of the SDRAM.

Figure 14.48 shows the mode register setting timing.

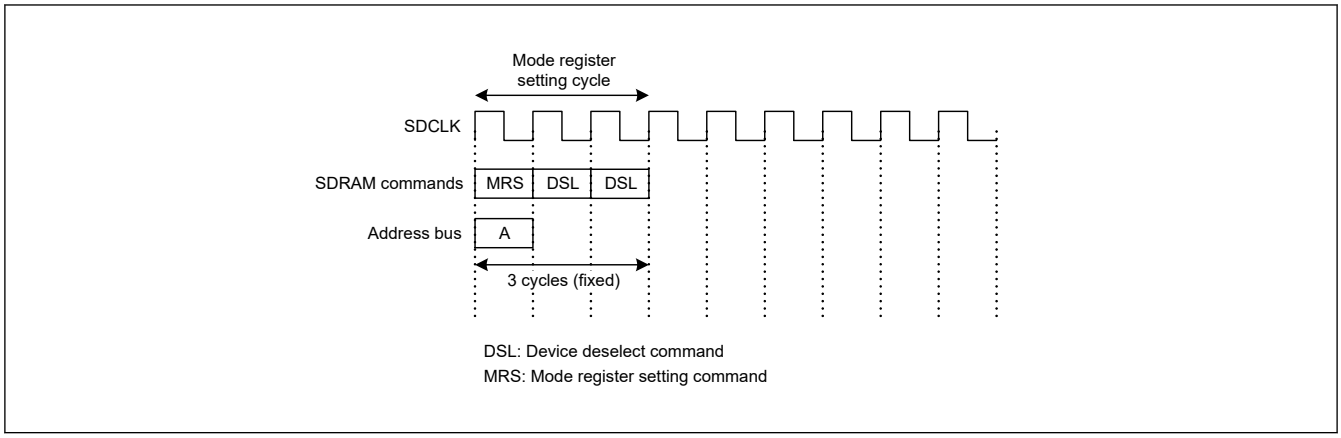


Figure 14.48 Mode register setting timing

14.6.12 SDRAMC Setting Examples

This section describes the following:

- SDRAMC setting procedure
- Timing register setting examples
- Procedure for transitioning to and recovering from self-refresh mode.

14.6.12.1 SDRAMC access procedure

Figure 14.49 shows the SDRAMC setting procedure.

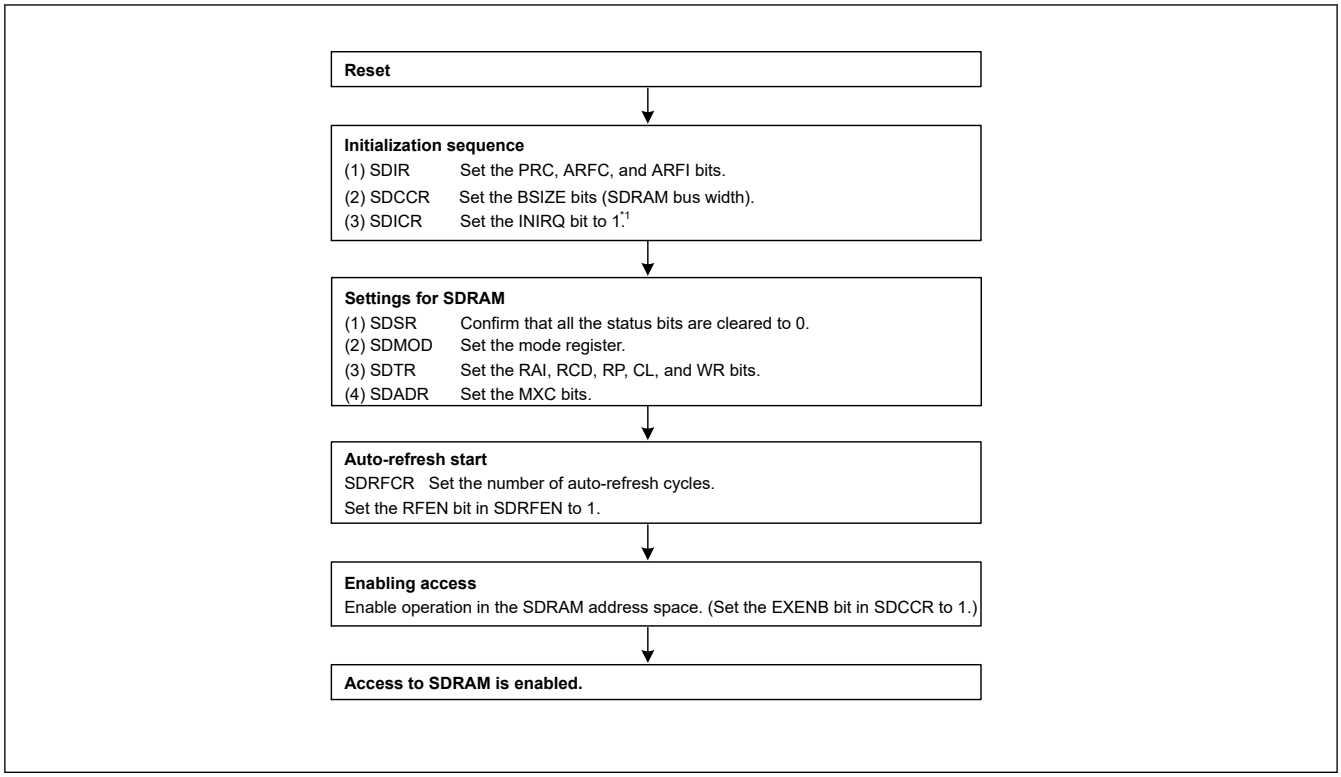


Figure 14.49 SDRAMC setting procedure

14.6.12.2 Procedure for transitioning to and recovering from self-refresh mode

Figure 14.50 shows the procedure for transitioning to and recovering from self-refresh mode.

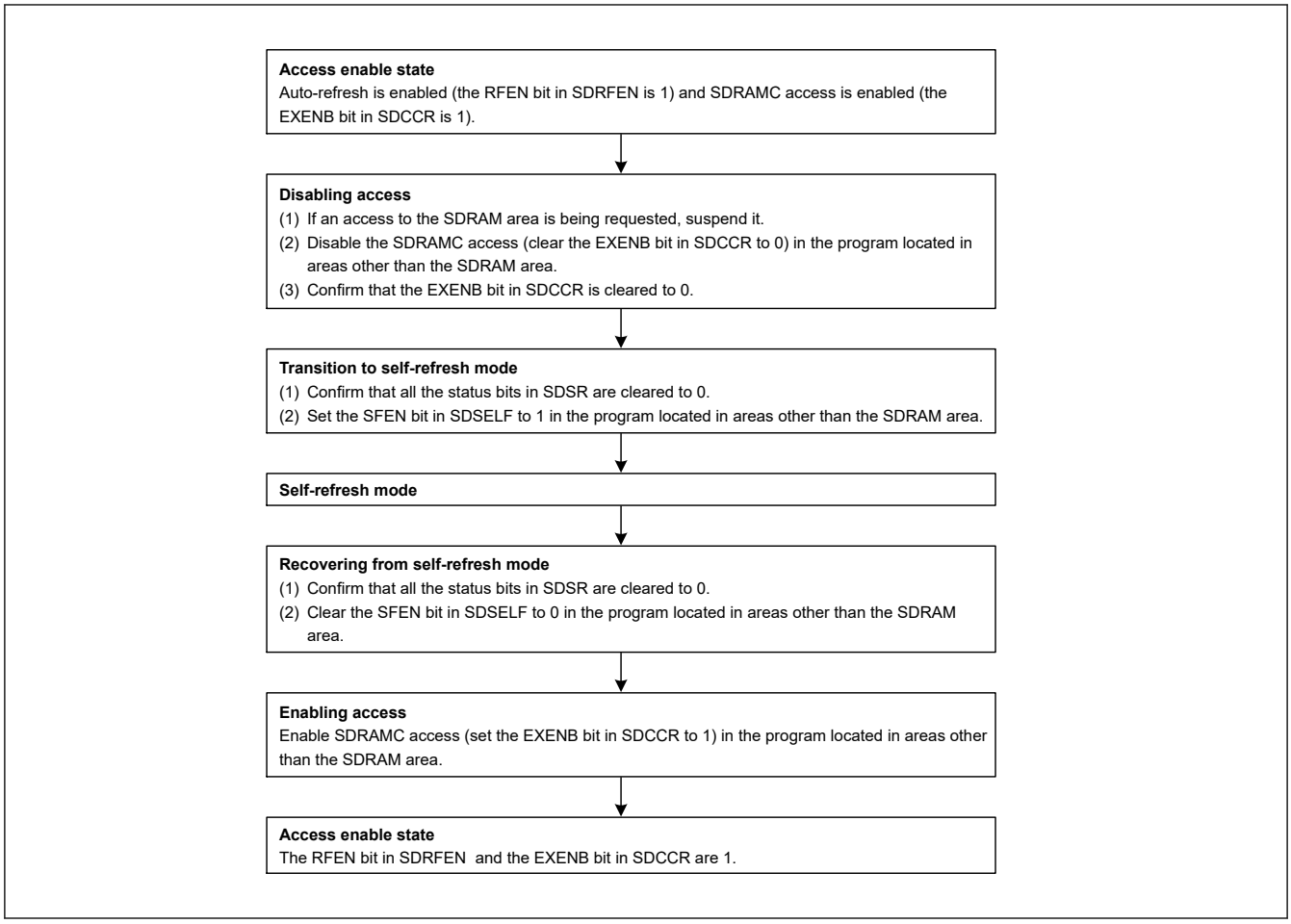


Figure 14.50 Procedure for transitioning to and recovering from self-refresh mode

Note: Self-refresh mode cannot be invoked during SDRAM access. SDRAM access must be disabled during both transition to and recovery from self-refresh mode. Follow the programming instructions shown in Figure 14.51.

Before transitioning to self-refresh mode, disable access to the SDRAM area.

During transition to self-refresh mode, self-refresh operation, and recovery from self-refresh mode, do not allow any operand access or instruction fetch, including prefetch to the SDRAM area, to be generated.

Figure 14.51 shows the procedure for transitioning to and recovering from self-refresh mode in Deep Software Standby mode.

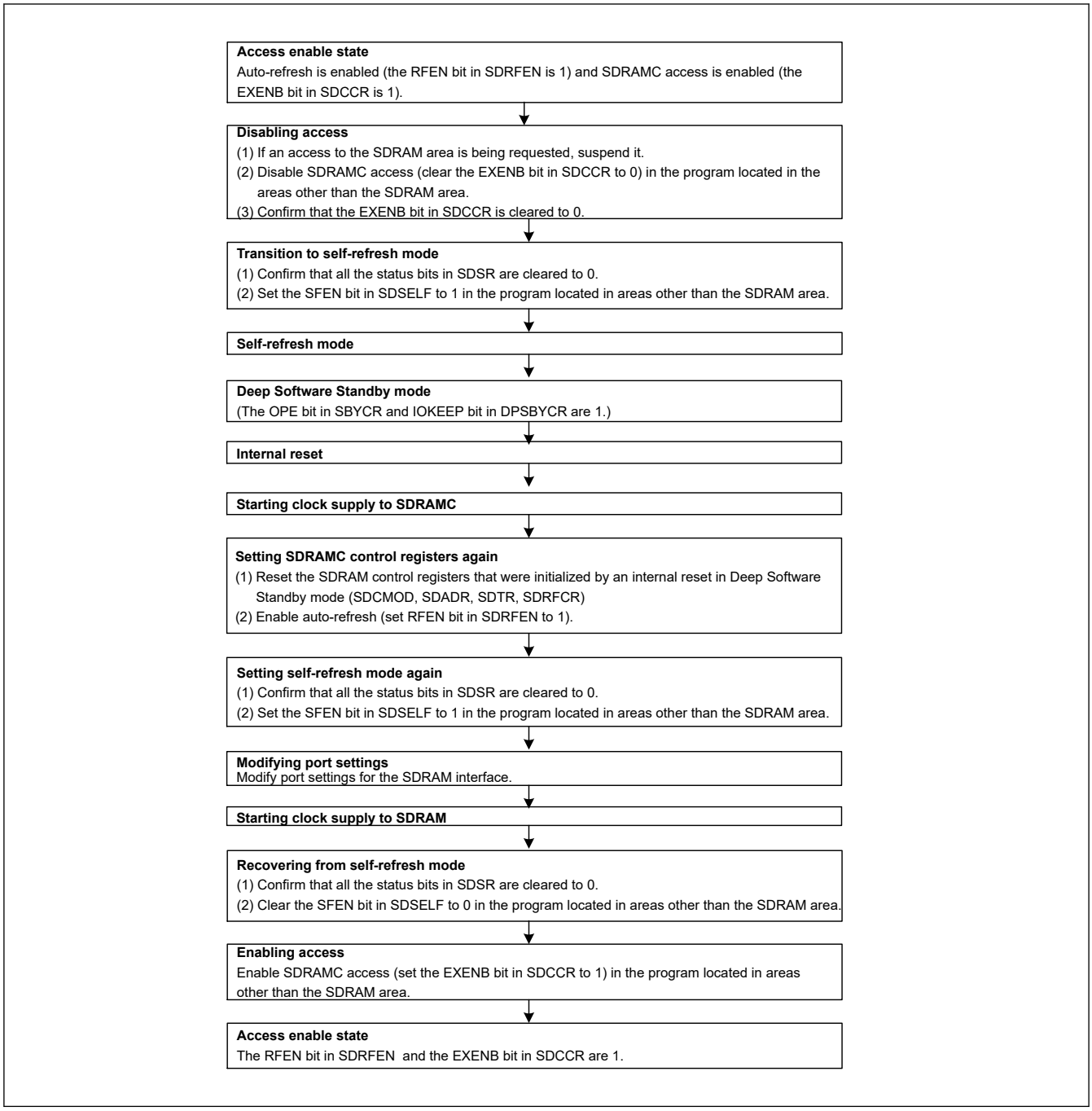


Figure 14.51 Procedure for transitioning to and recovering from self-refresh mode in Deep Software Standby mode

14.6.12.3 Timing register settings and access timing

This section describes the relationship between read and write timing and the settings in the SDRAM Timing Register (SDTR).

(1) Single read timing examples

Figure 14.52 to Figure 14.56 show the relationship between single read timing and the SDTR register settings. Table 14.35 shows the correspondence between the figures and the SDTR register settings.

During read access, the next bus access is enabled at the earliest 2 cycles after the read data becomes valid. However, if two or more accesses occur for one transfer request, the next bus access is enabled at the earliest 1 cycle after the read data becomes valid, as shown in Figure 14.56.

Table 14.35 Correspondence between timing figures and STDR register settings for single read timing

Figure number	RAI[2:0] settings	Number of cycles	RCD[1:0] settings	Number of cycles	RP[2:0] settings	Number of cycles	CL[2:0] settings	Number of cycles
Figure 14.52	010	3	00	1	001	2	010	2
Figure 14.53	000	1	01	2	001	2	010	2
Figure 14.54	000	1	01	2	001	2	011	3
Figure 14.55 Figure 14.56	010	3	00	1	000	1	010	2

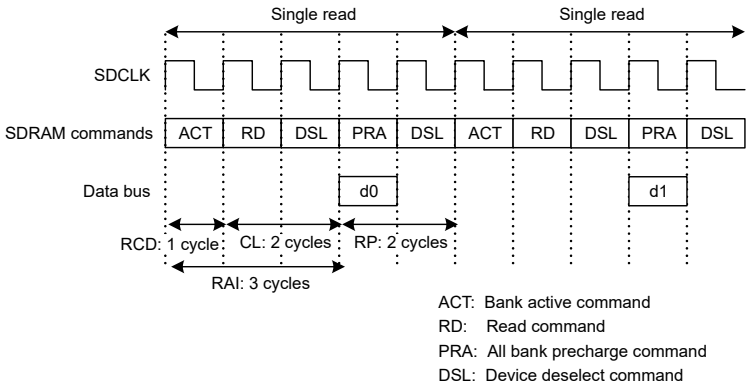
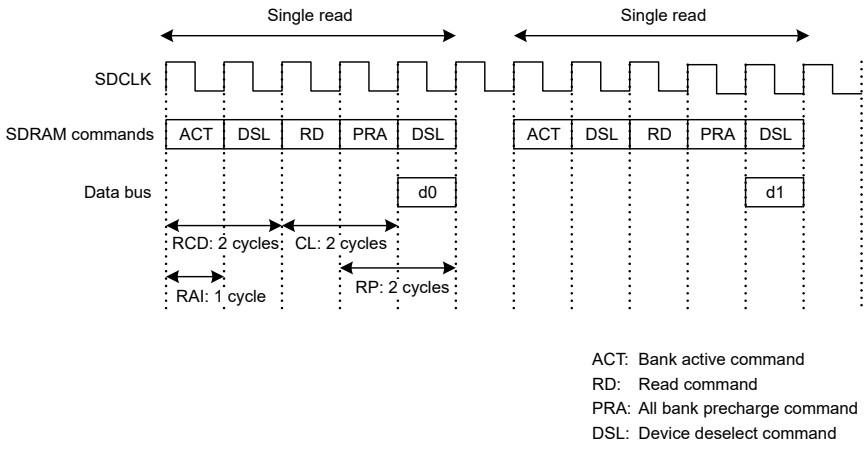


Figure 14.52 Example timing for single read (1)



Note: When the period set in the RAI bits ends earlier than the issue of the RD command, the PRA command is issued in the next cycle of the RD command cycle

Figure 14.53 Example timing for single read (2)

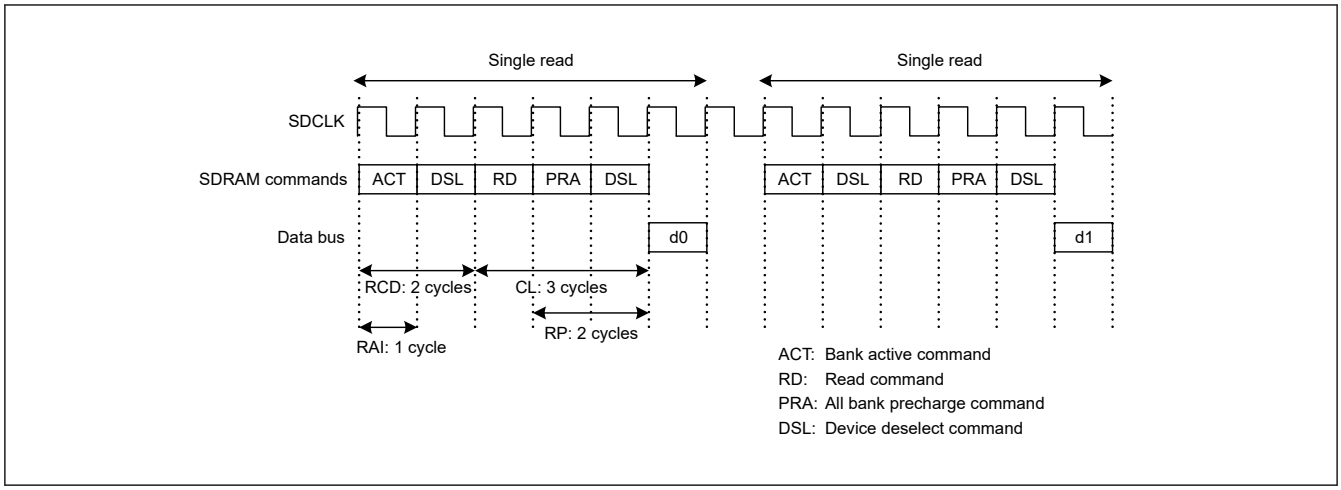


Figure 14.54 Example timing for single read (3)

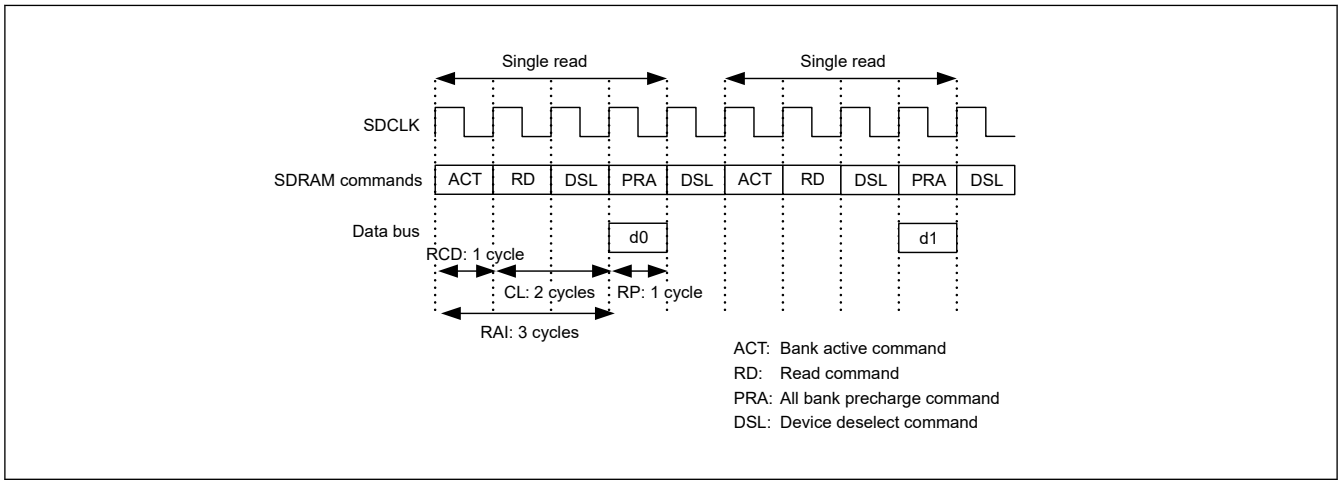


Figure 14.55 Example timing for single read (4)

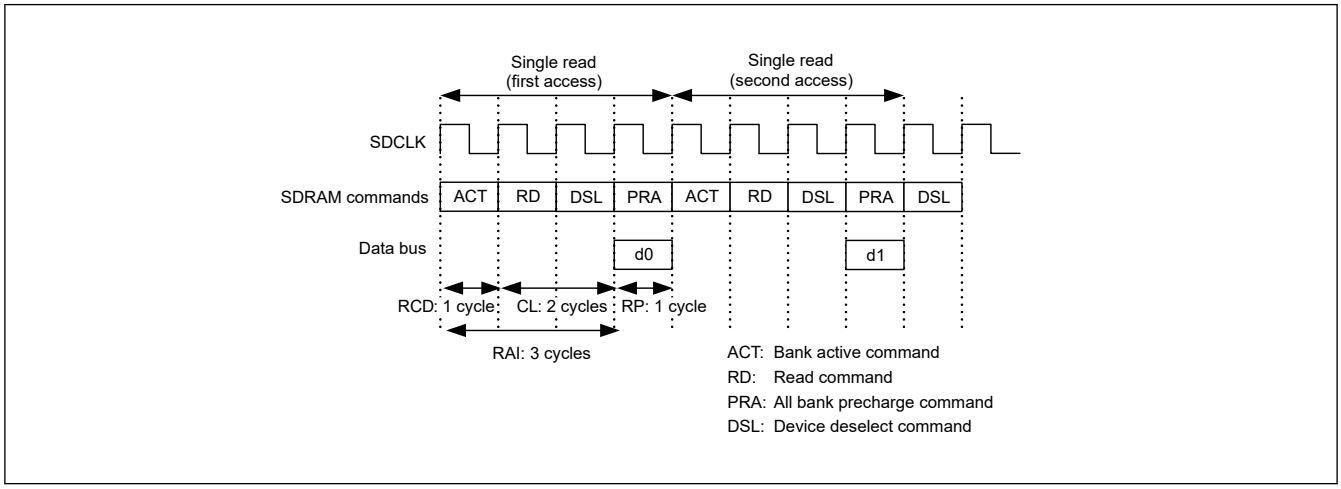


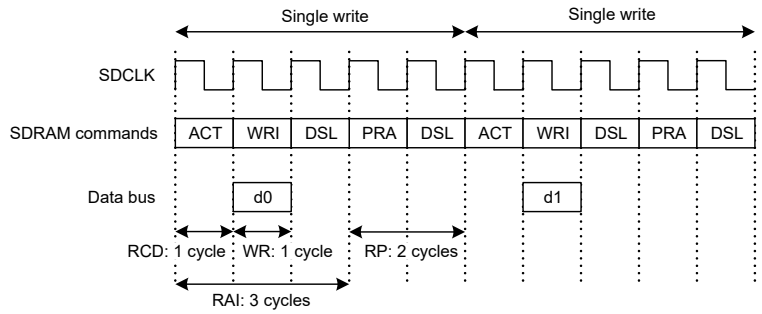
Figure 14.56 Example timing for single read (5), when two bus accesses occur for one transfer request

(2) Single write timing examples

Figure 14.57 to Figure 14.61 show the relationship between the single write timing and the SDTR register settings. Table 14.36 shows the association between the figures and the SDTR register settings. During write access, the next bus access is enabled at the earliest 2 cycles after an all bank precharge command (PRA) is issued. However, if two or more accesses occur for one transfer request, the next bus access is enabled at the earliest 1 cycle after the PRA is issued, as shown in Figure 14.61.

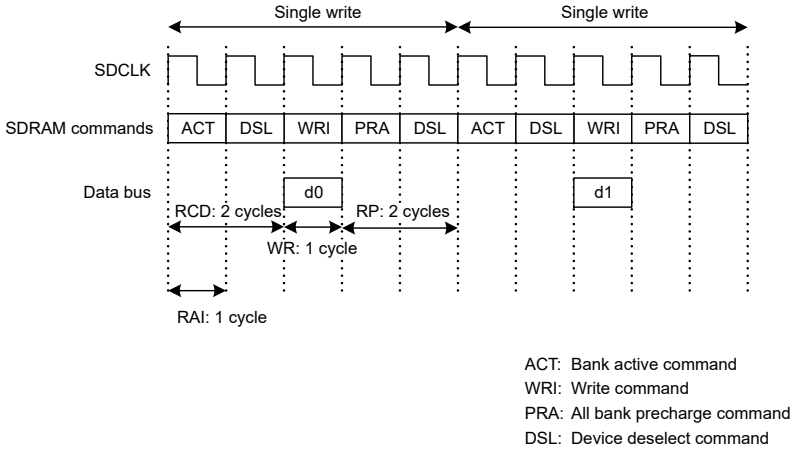
Table 14.36 Association between timing figures and STDR register settings for single write timing

Figure number	RAI[2:0] settings	Number of cycles	RCD[1:0] settings	Number of cycles	RP[2:0] settings	Number of cycles	WR settings	Number of cycles
Figure 14.57	010	3	00	1	001	2	0	1
Figure 14.58	000	1	01	2	001	2	0	1
Figure 14.59	000	1	01	2	001	2	1	2
Figure 14.60 Figure 14.61	010	3	00	0	000	2	0	1



Note: When the period specified in the RAI bits is longer than the period specified in WR bits, starting from the issue of the WRI command, the RAI setting is applied.

Figure 14.57 Example timing for single write (1)



Note: When the period specified in the RAI bits is shorter than the period specified in WR bits, starting from the issue of the WRI command, the WR setting is applied.

Figure 14.58 Example timing for single write (2)

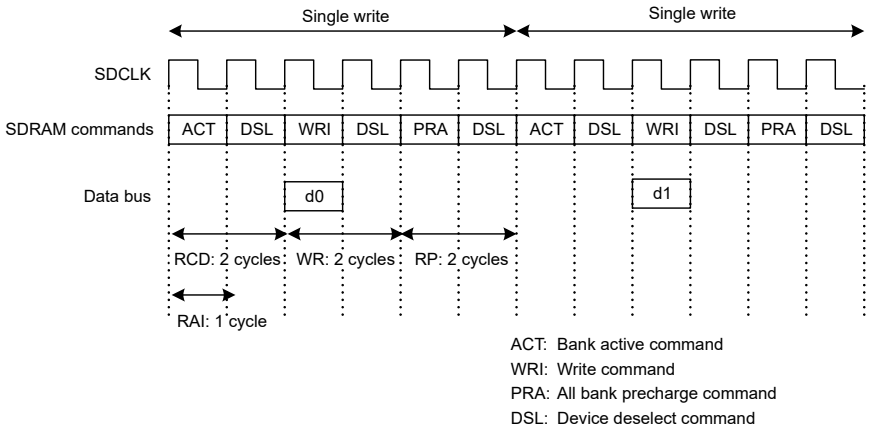
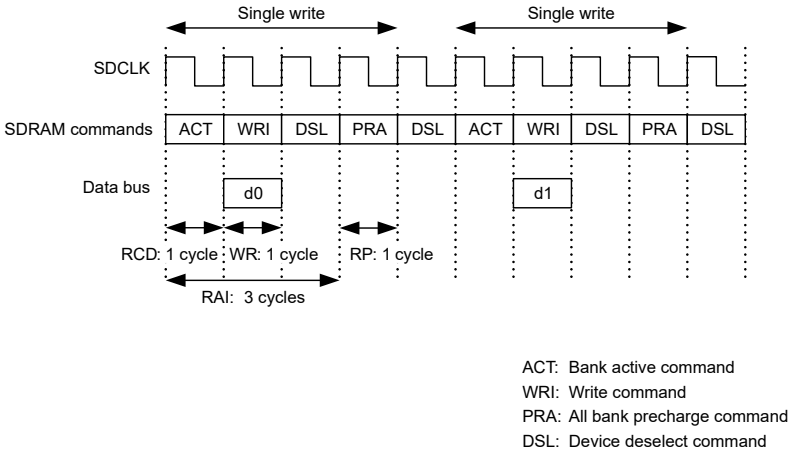
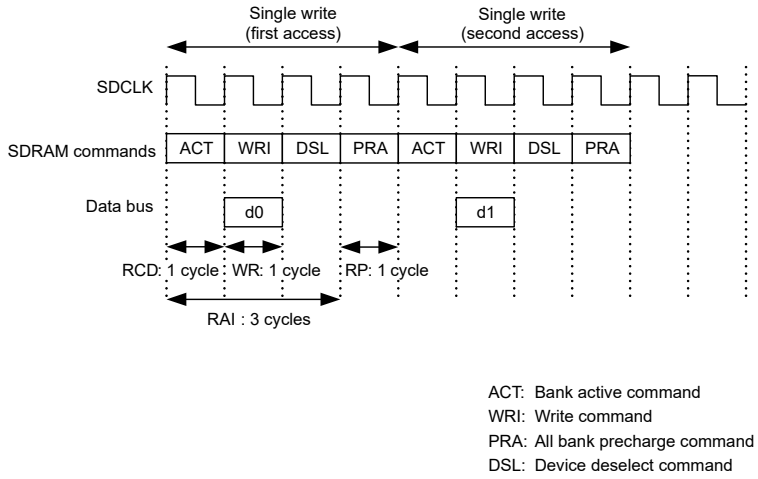


Figure 14.59 Example timing for single write (3)



Note: When the period specified in the RAI bits is longer than the period specified in WR bits, starting from the issue of he WRI command, the RAI setting is applied.

Figure 14.60 Example timing for single write (4)



Note: When the period specified in the RAI bits is longer than the period specified in WR bits, starting from the issue of the WRI command, the RAI setting is applied.

Figure 14.61 Example timing for single write (5), when two bus accesses occur for one transfer request

(3) Consecutive Read Timing Examples

Figure 14.62 to Figure 14.64 show the relationship between the consecutive read timing for four data and the SDTR register settings. Table 14.37 shows the correspondence between the figures and the SDTR register settings.

Table 14.37 Correspondence between Target Figures and STDR Register Settings (Consecutive Read Timing)

Figure No.	RAI[2:0] Settings	Number of Cycle	RCD[1:0] Settings	Number of Cycle	RP[2:0] Settings	Number of Cycle	CL[2:0] Settings	Number of Cycle
Figure 14.62	010	3	00	1	001	2	010	2
Figure 14.63	000	1	01	2	001	2	010	2
Figure 14.64	000	1	01	2	001	2	011	3

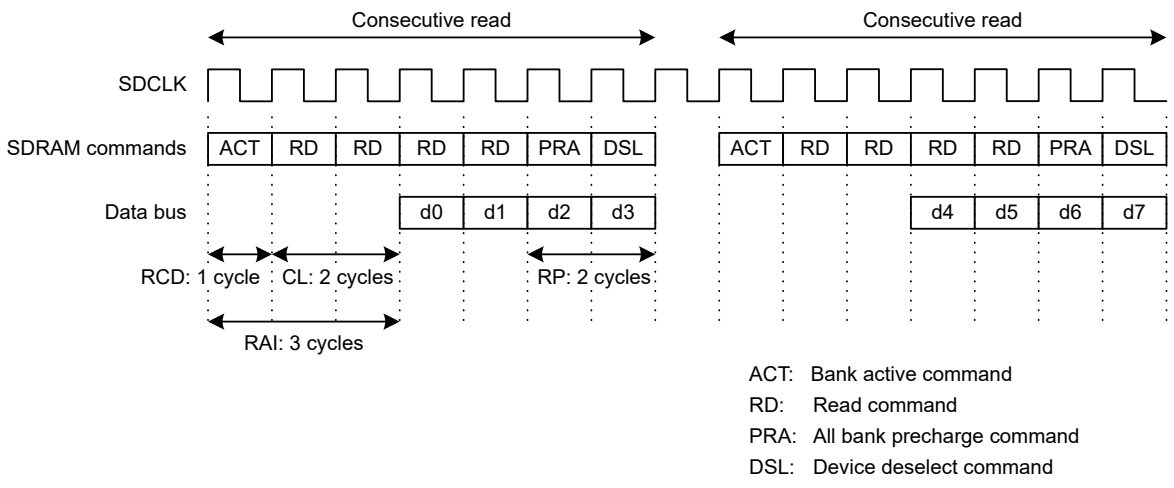


Figure 14.62 Timing example of consecutive read (1)

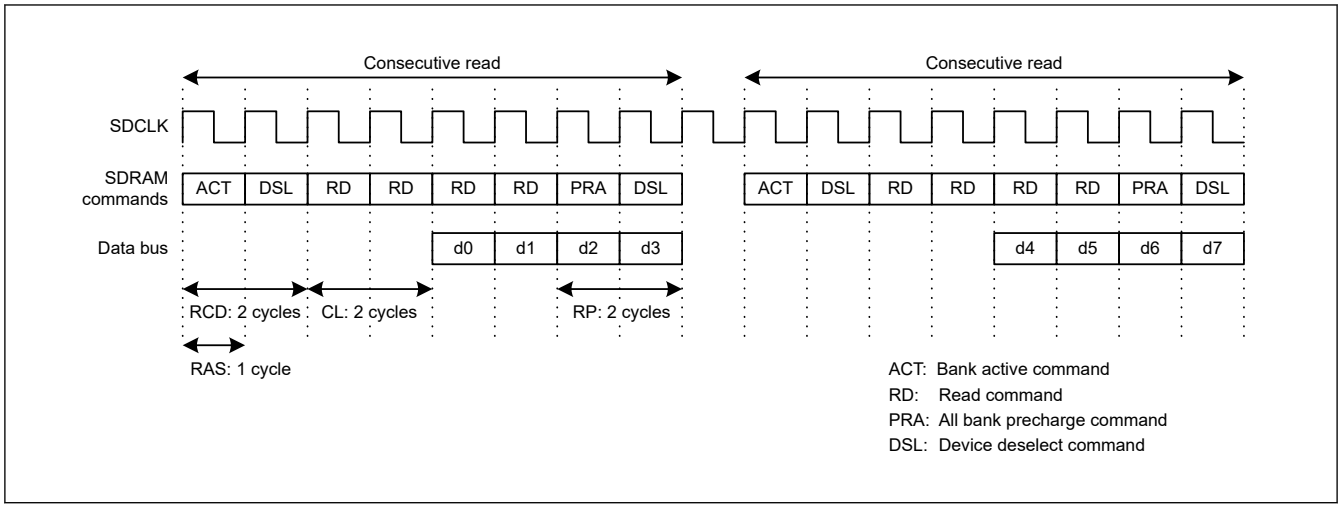


Figure 14.63 Timing example of consecutive read (2)

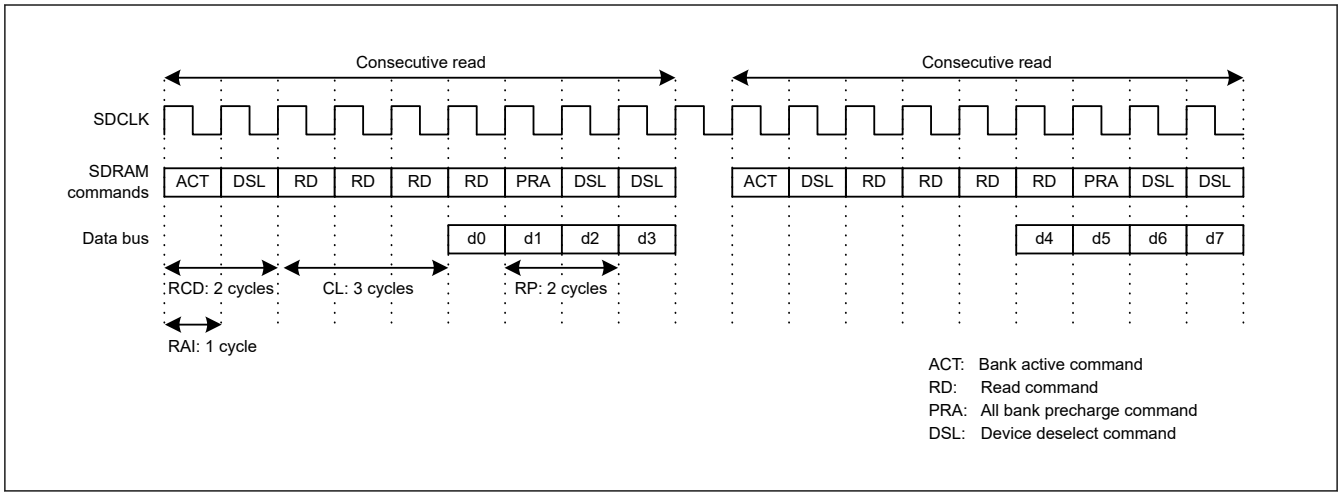


Figure 14.64 Timing example of consecutive read (3)

(4) Consecutive Write Timing Examples

Figure 14.65 to Figure 14.67 show the relationship between the consecutive write timing for four data and the SDTR register settings. Table 14.38 shows the correspondence between the figures and the SDTR register settings.

Table 14.38 Correspondence between Target Figures and STDR Register Settings (Consecutive Write Timing)

Figure No.	RAI[2:0] Settings	Number of Cycles	RCD[1:0] Settings	Number of Cycles	RP[2:0] Settings	Number of Cycles	WR Settings	Number of Cycles
Figure 14.65	010	3	00	1	001	2	0	1
Figure 14.66	000	1	01	2	001	2	0	1
Figure 14.67	000	1	01	2	001	2	1	2

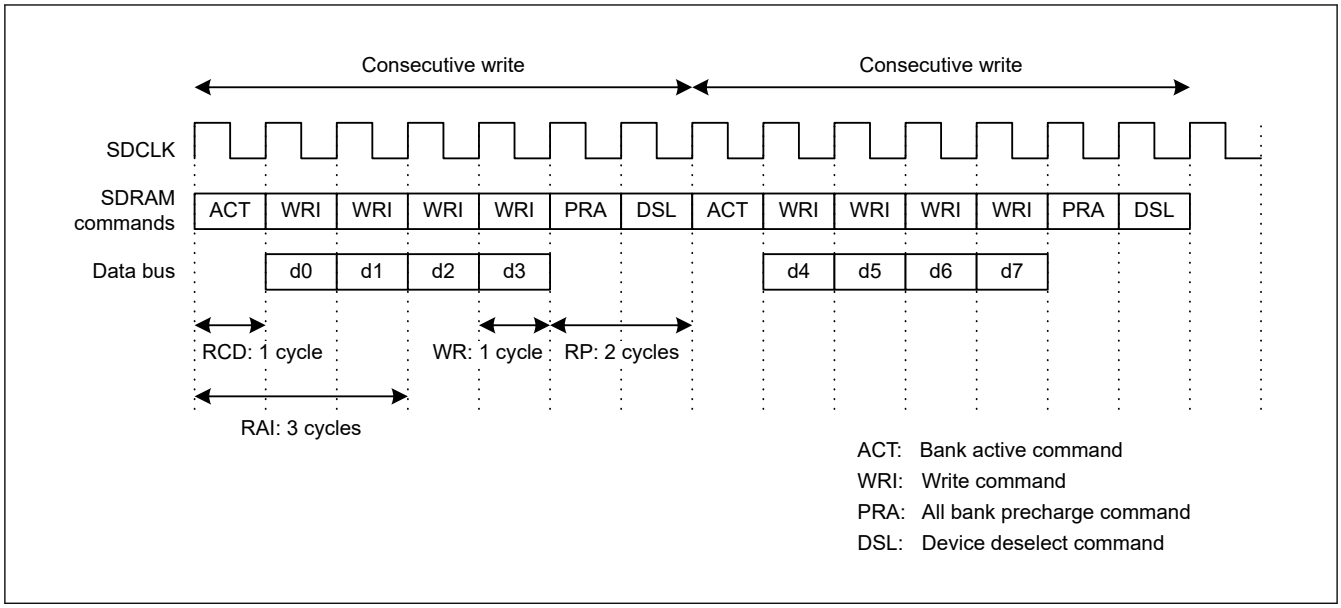


Figure 14.65 Timing example of consecutive write (1)

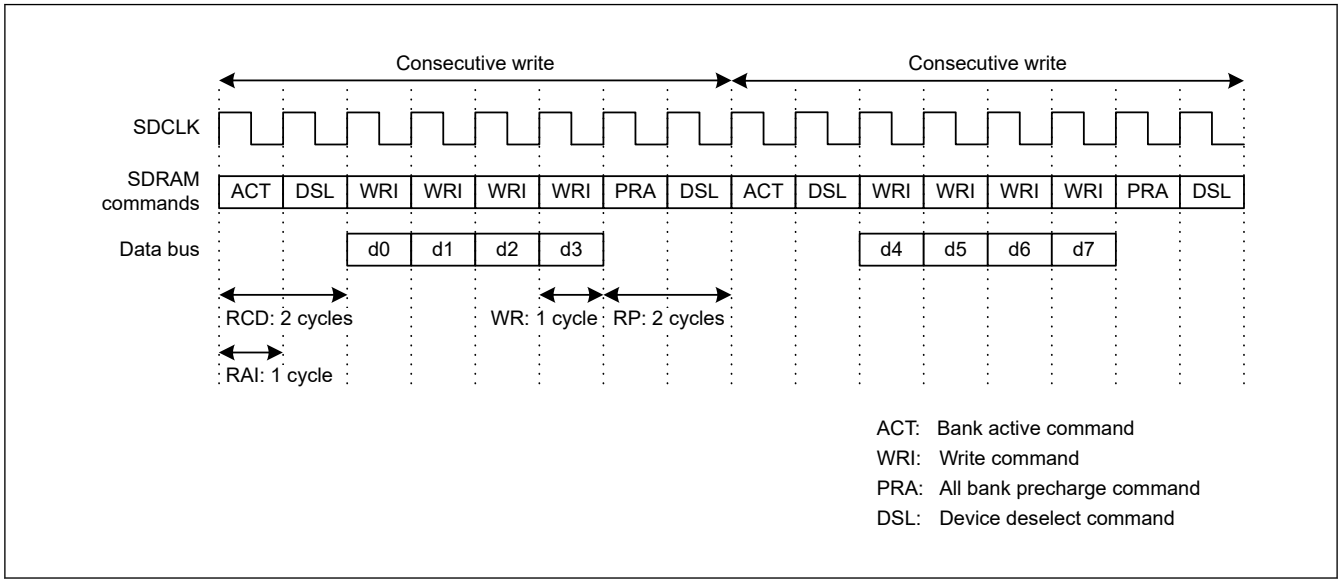


Figure 14.66 Timing example of consecutive write (2)

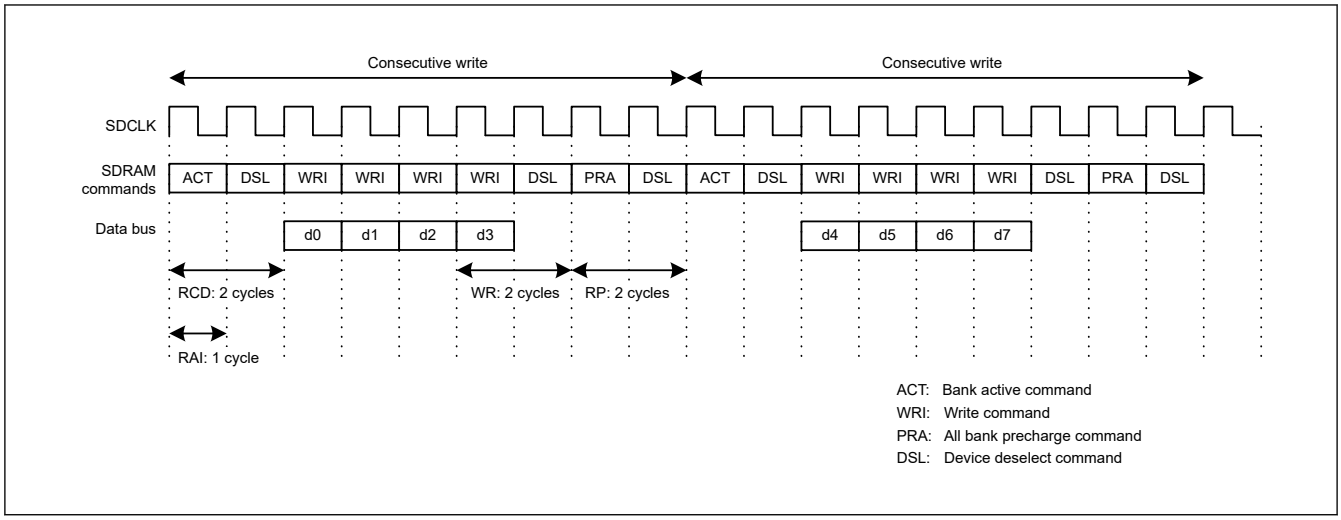


Figure 14.67 Timing example of consecutive write (3)

14.6.13 Address Multiplexing

In the SDRAM space, row and column addresses are multiplexed. The size of the shift in a row address must be specified in the Address Multiplex Select bits (SDADR.MXC[1:0]) in the SDRAM Address Register (SDADR). Additionally, in the SDRAM space, the address precharge-select command (Precharge-sel) is output to the upper bits of column addresses. Table 14.39 shows the relationship between the SDADR.MXC[1:0] settings and the shift amount.

Table 14.39 Address multiplexing (1 of 2)

MXC [1:0]	Shift amount	Data bus width	Address	Address pins external to the MCU																
				A16	A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
00	8 bits	8 bits	Row	A24	A23	A22	A21	A20	A19	A18*1	A17	A16	A15	A14	A13	A12	A11	A10	A09	A08
			Column	A24	A23	A22	A21	A20	A19	P	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		16 bits	Row	A24	A23	A22	A21	A20	A19*1	A18	A17	A16	A15	A14	A13	A12	A11	A10	A09	A08
			Column	A24	A23	A22	A21	A20	P	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		32 bits	Row	A24	A23	A22	A21	A20*1	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A09	A08
			Column	A24	A23	A22	A21	P	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
01	9 bits	8 bits	Row	A25	A24	A23	A22	A21	A20	A19*1	A18	A17	A16	A15	A14	A13	A12	A11	A10	A09
			Column	A25	A24	A23	A22	A21	A20	P	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		16 bits	Row	A25	A24	A23	A22	A21	A20*1	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A09
			Column	A25	A24	A23	A22	A21	P	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		32 bits	Row	A25	A24	A23	A22	A21*1	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A09
			Column	A25	A24	A23	A22	P	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
10	10 bits	8 bits	Row	A26	A25	A24	A23	A22	A21	A20*1	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
			Column	A26	A25	A24	A23	A22	A21	P	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		16 bits	Row	A26	A25	A24	A23	A22	A21*1	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
			Column	A26	A25	A24	A23	A22	P	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		32 bits	Row	A26	A25	A24	A23	A22*1	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
			Column	A26	A25	A24	A23	P	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00

Table 14.39 Address multiplexing (2 of 2)

MXC [1:0]	Shift amount	Data bus width	Address	Address pins external to the MCU															
				A16	A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
11	11 bits	8 bits	Row	—	A26	A25	A24	A23	A22	A21*1	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
			Column	—	A26	A25	A24	A23	A10	P	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		16 bits	Row	—	A26	A25	A24	A23	A22*1	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
			Column	—	A26	A25	A24	A11	P	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		32 bits	Row	—	A26	A25	A24	A23*1	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
			Column	—	A26	A25	A24	P	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00

Note: P: Precharge-select command (Precharge-sel) is output.
Note 1. When the PALL command is issued, Precharge-sel = 1 (high) is output. When the Active command is issued, the associated address is output.

14.6.14 Example SDRAM Connections

14.6.14.1 32-Bit Bus Space

Table 14.40 shows an example for connecting to two 512-Mb SDRAMs with 13-bit row address, 10-bit column address and 16-bit bus.

Table 14.40 SDRAM connection example (512-Mb × 2, with 16-bit bus) (1 of 3)

This MCU (shift amount: 10 bits)					SDRAM (512 Mb, with 16-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
1	RAS			→	RAS#			
2	CAS			→	CAS#			
3	WE			→	WE#			
4	CKE			→	CKE			
5	SDCS			→	CS#			
6	SDCLK			→	CLK			
7	DQM1			→	UDQM			
8	DQM0			→	LDQM			
9	A16	A26	A26	→	BA1 (A14)	BA1	—	—
10	A15	A25	A25	→	BA0 (A13)	BA0	—	—
11	A14	A24	A24	→	A12		A12	—
12	A13	A23	A23	→	A11		A11	—
13	A12	A22	P	→	A10	AP	A10	—
14	A11	A21	A11	→	A9		A9	A9
15	A10	A20	A10	→	A8		A8	A8
16	A09	A19	A9	→	A7		A7	A7
17	A08	A18	A8	→	A6		A6	A6
18	A07	A17	A7	→	A5		A5	A5
19	A06	A16	A6	→	A4		A4	A4
20	A05	A15	A5	→	A3		A3	A3
21	A04	A14	A4	→	A2		A2	A2
22	A03	A13	A3	→	A1		A1	A1
23	A02	A12	A2	→	A0		A0	A0

Table 14.40 SDRAM connection example (512-Mb × 2, with 16-bit bus) (2 of 3)

This MCU (shift amount: 10 bits)					SDRAM (512 Mb, with 16-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
24	DQ15 to DQ08			↔	DQ[15:8]			
25	DQ07 to DQ00			↔	DQ[7:0]			

Table 14.40 SDRAM connection example (512-Mb × 2, with 16-bit bus) (3 of 3)

This MCU (shift amount: 9 bits)					SDRAM (128 Mb, with 16-bit bus)			
No.*1	Pin name	Row	Column		Pin name	BA/AP	Row	Column
(1)	RAS			→	RAS#			
(2)	CAS			→	CAS#			
(3)	WE			→	WE#			
(4)	CKE			→	CKE			
(5)	SDCS			→	CS#			
(6)	SDCLK			→	CLK			
26	DQM3			→	UDQM			
27	DQM2			→	LDQM			
(9)	A16	A26	A26	→	BA1 (A14)	BA1	—	—
(10)	A15	A25	A25	→	BA0 (A13)	BA0	—	—
(11)	A14	A24	A24	→	A12		A12	—
(12)	A13	A23	A23	→	A11		A11	—
(13)	A12	A22	P	→	A10	AP	A10	—
(14)	A11	A21	A11	→	A9		A9	A9
(15)	A10	A20	A10	→	A8		A8	A8
(16)	A09	A19	A9	→	A7		A7	A7
(17)	A08	A18	A8	→	A6		A6	A6
(18)	A07	A17	A7	→	A5		A5	A5
(19)	A06	A16	A6	→	A4		A4	A4
(20)	A05	A15	A5	→	A3		A3	A3
(21)	A04	A14	A4	→	A2		A2	A2
(22)	A03	A13	A3	→	A1		A1	A1
(23)	A02	A12	A2	→	A0		A0	A0
28	DQ31 to DQ24			↔	DQ[15:8]			
29	DQ23 to DQ16			↔	DQ[7:0]			

Note 1. The number (No.) in parentheses indicates that the pin branches out to two SDRAM pins.

Table 14.41 shows an example for connecting to a 256-Mb SDRAM with 12-bit row address, 9-bit column address and 32-bit bus.

Table 14.41 SDRAM connection example (256-Mb × 1, with 32-bit bus) (1 of 2)

This MCU (shift amount: 9 bits)					SDRAM (256 Mb, with 32-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
1	RAS			→	RAS#			
2	CAS			→	CAS#			
3	WE			→	WE#			
4	CKE			→	CKE			

Table 14.41 SDRAM connection example (256-Mb × 1, with 32-bit bus) (2 of 2)

This MCU (shift amount: 9 bits)					SDRAM (256 Mb, with 32-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
5	SDCS			→	CS#			
6	SDCLK			→	CLK			
7	DQM3			→	DQM3			
8	DQM2			→	DQM2			
9	DQM1			→	DQM1			
10	DQM0			→	DQM0			
11	A15	A24	A24	→	BA1 (A13)	BA1	—	—
12	A14	A23	A23	→	BA0 (A12)	BA0	—	—
13	A13	A22	A22	→	A11		A11	—
14	A12	A21	P	→	A10	AP	A10	—
15	A11	A20	A11	→	A9		A9	—
16	A10	A19	A10	→	A8		A8	A8
17	A09	A18	A9	→	A7		A7	A7
18	A08	A17	A8	→	A6		A6	A6
19	A07	A16	A7	→	A5		A5	A5
20	A06	A15	A6	→	A4		A4	A4
21	A05	A14	A5	→	A3		A3	A3
22	A04	A13	A4	→	A2		A2	A2
23	A03	A12	A3	→	A1		A1	A1
24	A02	A11	A2	→	A0		A0	A0
25	DQ31 to DQ24			↔	DQ[31:24]			
26	DQ23 to DQ16			↔	DQ[23:16]			
27	DQ15 to DQ08			↔	DQ[15:8]			
28	DQ07 to DQ00			↔	DQ[7:0]			

Table 14.42 shows an example for connecting to two 128-Mb SDRAMs with 12-bit row address, 9-bit column address and 16-bit bus.

Table 14.42 SDRAM connection example (128-Mb × 2, with 16-bit bus) (1 of 3)

This MCU (shift amount: 9 bits)					SDRAM (128 Mb, with 16-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
1	RAS			→	RAS#			
2	CAS			→	CAS#			
3	WE			→	WE#			
4	CKE			→	CKE			
5	SDCS			→	CS#			
6	SDCLK			→	CLK			
7	DQM1			→	UDQM			
8	DQM0			→	LDQM			
9	A15	A24	A24	→	BA1 (A13)	BA1	—	—
10	A14	A23	A23	→	BA0 (A12)	BA0	—	—
11	A13	A22	A22	→	A11		A11	—

Table 14.42 SDRAM connection example (128-Mb × 2, with 16-bit bus) (2 of 3)

This MCU (shift amount: 9 bits)					SDRAM (128 Mb, with 16-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
12	A12	A21	P	→	A10	AP	A10	—
13	A11	A20	A11	→	A9		A9	—
14	A10	A19	A10	→	A8		A8	A8
15	A09	A18	A9	→	A7		A7	A7
16	A08	A17	A8	→	A6		A6	A6
17	A07	A16	A7	→	A5		A5	A5
18	A06	A15	A6	→	A4		A4	A4
19	A05	A14	A5	→	A3		A3	A3
20	A04	A13	A4	→	A2		A2	A2
21	A03	A12	A3	→	A1		A1	A1
22	A02	A11	A2	→	A0		A0	A0
23	DQ15 to DQ08			↔	DQ[15:8]			
24	DQ07 to DQ00			↔	DQ[7:0]			

Table 14.42 SDRAM connection example (128-Mb × 2, with 16-bit bus) (3 of 3)

This MCU (shift amount: 9 bits)					SDRAM (128 Mb, with 16-bit bus)			
No.*1	Pin name	Row	Column		Pin name	BA/AP	Row	Column
(1)	RAS			→	RAS#			
(2)	CAS			→	CAS#			
(3)	WE			→	WE#			
(4)	CKE			→	CKE			
(5)	SDCS			→	CS#			
(6)	SDCLK			→	CLK			
25	DQM3			→	UDQM			
26	DQM2			→	LDQM			
(9)	A15	A24	A24	→	BA1 (A13)	BA1	—	—
(10)	A14	A23	A23	→	BA0 (A12)	BA0	—	—
(11)	A13	A22	A22	→	A11		A11	—
(12)	A12	A21	P	→	A10	AP	A10	—
(13)	A11	A20	A11	→	A9		A9	—
(14)	A10	A19	A10	→	A8		A8	A8
(15)	A09	A18	A9	→	A7		A7	A7
(16)	A08	A17	A8	→	A6		A6	A6
(17)	A07	A16	A7	→	A5		A5	A5
(18)	A06	A15	A6	→	A4		A4	A4
(19)	A05	A14	A5	→	A3		A3	A3
(20)	A04	A13	A4	→	A2		A2	A2
(21)	A03	A12	A3	→	A1		A1	A1
(22)	A02	A11	A2	→	A0		A0	A0
27	DQ31 to DQ24			↔	DQ[15:8]			
28	DQ23 to DQ16			↔	DQ[7:0]			

Note 1. The number (No.) in parentheses indicates that the pin branches out to two SDRAM pins.

14.6.14.2 16-bit bus space

Table 14.43 shows an example connection to two 512-Mb SDRAMs with a 13-bit row address, 11-bit column address, and 8-bit bus.

Table 14.43 SDRAM connection example with 512-Mb × 2 and 8-bit bus (1 of 3)

This MCU (shift amount: 11 bits)					SDRAM (512 Mb, with 8-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
1	RAS			→	RAS#			
2	CAS			→	CAS#			
3	WE			→	WE#			
4	CKE			→	CKE			
5	SDCS			→	CS#			
6	SDCLK			→	CLK			
7	DQM0			→	LDQM			
8	A15	A26	A26	→	BA1 (A14)	BA1	—	—
9	A14	A25	A25	→	BA0 (A13)	BA0	—	—
10	A13	A24	A24	→	A12		A12	—
11	A12	A23	A11	→	A11		A11	A11
12	A11	A22	P	→	A10	AP	A10	—
13	A10	A21	A10	→	A9		A9	A9
14	A09	A20	A9	→	A8		A8	A8
15	A08	A19	A8	→	A7		A7	A7
16	A07	A18	A7	→	A6		A6	A6
17	A06	A17	A6	→	A5		A5	A5
18	A05	A16	A5	→	A4		A4	A4
19	A04	A15	A4	→	A3		A3	A3
20	A03	A14	A3	→	A2		A2	A2
21	A02	A13	A2	→	A1		A1	A1
22	A01	A12	A1	→	A0		A0	A0
23	DQ07 to DQ00			↔	DQ[7:0]			

Table 14.43 SDRAM connection example with 512-Mb × 2 and 8-bit bus (2 of 3)

This MCU (shift amount: 11 bits)					SDRAM (512 Mb, with 8-bit bus)			
No.*1	Pin name	Row	Column		Pin name	BA/AP	Row	Column
(1)	RAS			→	RAS#			
(2)	CAS			→	CAS#			
(3)	WE			→	WE#			
(4)	CKE			→	CKE			
(5)	SDCS			→	CS#			
(6)	SDCLK			→	CLK			
24	DQM1			→	LDQM			
(8)	A15	A26	A26	→	BA1 (A14)	BA1	—	—
(9)	A14	A25	A25	→	BA0 (A13)	BA0	—	—

Table 14.43 SDRAM connection example with 512-Mb × 2 and 8-bit bus (3 of 3)

This MCU (shift amount: 11 bits)					SDRAM (512 Mb, with 8-bit bus)			
No.*1	Pin name	Row	Column		Pin name	BA/AP	Row	Column
(10)	A13	A24	A24	→	A12		A12	—
(11)	A12	A23	A11	→	A11		A11	A11
(12)	A11	A22	P	→	A10	AP	A10	—
(13)	A10	A21	A10	→	A9		A9	A9
(14)	A09	A20	A9	→	A8		A8	A8
(15)	A08	A19	A8	→	A7		A7	A7
(16)	A07	A18	A7	→	A6		A6	A6
(17)	A06	A17	A6	→	A5		A5	A5
(18)	A05	A16	A5	→	A4		A4	A4
(19)	A04	A15	A4	→	A3		A3	A3
(20)	A03	A14	A3	→	A2		A2	A2
(21)	A02	A13	A2	→	A1		A1	A1
(22)	A01	A12	A1	→	A0		A0	A0
25	DQ15 to DQ08			↔	DQ[7:0]			

Note 1. The number (No.) in parentheses indicates that the pin branches out to two SDRAM pins.

Table 14.44 shows an example connection to a 512-Mb SDRAMs with a 13-bit row address, 10-bit column address, and 16-bit bus.

Table 14.44 SDRAM connection example with 512-Mb × 1 and 16-bit bus (1 of 2)

This MCU (shift amount: 10 bits)					SDRAM (512 Mb, with 16-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
1	RAS			→	RAS#			
2	CAS			→	CAS#			
3	WE			→	WE#			
4	CKE			→	CKE			
5	SDCS			→	CS#			
6	SDCLK			→	CLK			
7	DQM1			→	UDQM			
8	DQM0			→	LDQM			
9	A15	A25	A25	→	BA1 (A14)	BA1	—	—
10	A14	A24	A24	→	BA0 (A13)	BA0	—	—
11	A13	A23	A23	→	A12		A12	—
12	A12	A22	A22	→	A11		A11	—
13	A11	A21	P	→	A10	AP	A10	—
14	A10	A20	A10	→	A9		A9	A9
15	A09	A19	A9	→	A8		A8	A8
16	A08	A18	A8	→	A7		A7	A7
17	A07	A17	A7	→	A6		A6	A6
18	A06	A16	A6	→	A5		A5	A5
19	A05	A15	A5	→	A4		A4	A4
20	A04	A14	A4	→	A3		A3	A3

Table 14.44 SDRAM connection example with 512-Mb × 1 and 16-bit bus (2 of 2)

This MCU (shift amount: 10 bits)					SDRAM (512 Mb, with 16-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
21	A03	A13	A3	→	A2		A2	A2
22	A02	A12	A2	→	A1		A1	A1
23	A01	A11	A1	→	A0		A0	A0
24	DQ15 to DQ08			↔	DQ[15:8]			
25	DQ07 to DQ00			↔	DQ[7:0]			

Table 14.45 shows an example connection to a 256-Mb SDRAMs with a 13-bit row address, 9-bit column address, and 16-bit bus.

Table 14.45 SDRAM connection example with 256-Mb × 1 and 16-bit bus

This MCU (shift amount: 9 bits)					SDRAM (256 Mb, with 16-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
1	RAS			→	RAS#			
2	CAS			→	CAS#			
3	WE			→	WE#			
4	CKE			→	CKE			
5	SDCS			→	CS#			
6	SDCLK			→	CLK			
7	DQM1			→	UDQM			
8	DQM0			→	LDQM			
9	A15	A24	A24	→	BA1 (A14)	BA1	—	—
10	A14	A23	A23	→	BA0 (A13)	BA0	—	—
11	A13	A22	A22	→	A12		A12	—
12	A12	A21	A21	→	A11		A11	—
13	A11	A20	P	→	A10	AP	A10	—
14	A10	A19	A10	→	A9		A9	—
15	A09	A18	A9	→	A8		A8	A8
16	A08	A17	A8	→	A7		A7	A7
17	A07	A16	A7	→	A6		A6	A6
18	A06	A15	A6	→	A5		A5	A5
19	A05	A14	A5	→	A4		A4	A4
20	A04	A13	A4	→	A3		A3	A3
21	A03	A12	A3	→	A2		A2	A2
22	A02	A11	A2	→	A1		A1	A1
23	A01	A10	A1	→	A0		A0	A0
24	DQ15 to DQ08			↔	DQ[15:8]			
25	DQ07 to DQ00			↔	DQ[7:0]			

14.6.15 Constraints

(1) Constraints on using separate bus interface

In Software Standby and Deep Software Standby modes, auto-refresh operation is not available because the clock supply to the SDRAMC is stopped. To retain the data in the SDRAM when the SDRAM is externally connected, use the self-refresh function. For the procedure for transitioning to and recovering from self-refresh mode, see section 14.6.7. Self-Refresh.

(2) Setting the SDRAM Timing Register

Set the RAI[2:0] bits in the SDRAM Timing Register (SDTR) to a value less than or equal to the sum of the row column latency (SDTR.RCD[1:0]) and column latency (SDTR.CL[2:0]) settings. Operation is not guaranteed if this condition is not satisfied.

(3) Instruction code constraint

It should be fixed the instruction code to little-endian order.

(4) Consecutive-Access Mode

For consecutive access mode, the setting CL = 1 is prohibited, and operation is not guaranteed if this setting is made.

14.7 Bus Error Monitoring Section

This monitoring system monitors each individual area, and whenever it detects an error, it returns the error to the requesting master using the AHB-Lite error response protocol.

14.7.1 Bus Error Types

The following types of errors can occur on each bus:

- Master Security Attribution Unit error
- Bus master MPU error.
- Illegal address access.
- Slave TrustZone Filter error
- Bus error transmitted from each slave module.

Table 14.49 lists the address ranges where access leads to illegal address access errors. The reserved area in the slave does not trigger an illegal address access error. For more information on the bus master MPU, see section 15, Memory Protection Unit (MPU).

14.7.2 Operations When a Bus Error Occurs

In the following bufferable write access, the write response can be given from the midpoint in the following cases.

1. CEU accesses ECBI.
2. CPU M-AXI accesses ECBI, FLBI, ROMBI and STBYSBI.
3. CPU P-AHB accesses an accessible area.
4. CPU P-AHB accesses illegal address.
5. DMA accesses ECBI, EOBI, FLBI, PABI, PBBI, S0BI, S1BI and PSBI.

If an access violation occurs due to these bufferable write accesses, an error response may not be returned to the bus master.

When a bufferable write access error that does not return an error response is detected, access is blocked and 1 is set in the bit corresponding to the error occurrence location of MBWERRSTAT or SBWERRSTAT. Furthermore, NMI request or Reset request is generated according to the setting of the BUSOAD.BWERROAD. Since MBWERRSTAT or SBWERRSTAT are held until reset other than Bus and Memory Error Reset or cleared by MBWERRCLR or SBWERRCLR, they can be confirmed in the Interrupt handler. Error address information and Error R/W information are not retained with a bufferable write access. If error information is required, set the Bus Master to non-bufferable access.

A bufferable write access error can occur in the following cases. In this case, BUSnERRSTAT is not set and NMI request or Reset request is not generated according to the setting of the BUSOAD.ILERROAD or BUSOAD.SLERROAD.

1. CEU accesses ECBI and a slave error or slave TrustZone Filter error occurs.
2. CPU M-AXI accesses ECBI, FLBI, ROMBI and STBYSBI and a slave error or slave TrustZone Filter error occurs.
3. CPU P-AHB accesses an accessible area and a slave error or slave TrustZone Filter error occurs.
4. CPU P-AHB accesses illegal address and a illegal address access error occurs.

5. DMA accesses ECBI, EOBI, FLBI, PABI, PBBI, S0BI, S1BI and PSBI and a slave error or slave TrustZone Filter error occurs.

Table 14.46 Bus error notice method for bufferable write access

	DAP	CPU	DMAC/DTC	EDMAC	DRW0/1	MIPI	GLCDC0/1, CEU
BUSOAD.BWERROAD = 0	BUSERR.NMI	BUSERR.NMI	BUSERR.NMI	BUSERR.NMI	BUSERR.NMI	BUSERR.NMI	BUSERR.NMI
BUSOAD.BWERROAD = 1	Reset	Reset	Reset	Reset	Reset	Reset	Reset

When a bus error occurs for an access other than the bufferable write, error access is blocked, and error operations determine by OAD setting and the bus master are performed. Table 14.47 shows OAD setting corresponding to the each error. Table 14.48 shows how to notify when a bus error occurs.

When OAD setting is NMI.

If CPU is the bus master, it only returns an error response to bus master. CPU generates BusFault when it receives error response. If master other than CPU is the bus master, error response is returned to the master and NMI is generated.

When OAD setting is Reset.

Error response is returned to the bus master and reset request is generated.

Table 14.47 OAD setting corresponding to the each error

Error type	OAD bit
MSAU Error	MSAOAD.OAD
MPU Error	MMPUOAD.OAD
Illegal access Error	BUSOAD.ILERROAD
Slave Bus Error	BUSOAD.SLERROAD
Slave TrustZone Filter error	BUSOAD.SLERROAD

Table 14.48 Bus error notice method for Non-bufferable write access

OAD setting	DAP*2	CPU*1	DMAC/DTC	EDMAC	DRW0/1	MIPI	GLCDC0/1, CEU
NMI	Error response only	CPU.BusFault	DMA0_TRANSERR*1 BUSERR.NMI	ETHER_EINT*1 BUSERR.NMI	DRW_IRQ*1 BUSERR.NMI	DSI_SEQ0*1 / DSI_SEQ1*1 / DSI_RCV*1 BUSERR.NMI	BUSERR.NMI
Reset	Error response only	Reset	Reset	Reset	Reset	Reset	Reset

Note 1. DMAC/DTC, EDMAC, DRW and MIPI can issue interrupt when it receives error response. This NMI is output to ICU and transmitted from ICU to CPU.

Note 2. Only when the bus master is DAP, only an error response is returned regardless of the BUSOAD.ILERROAD or the BUSOAD.SLERROAD settings, and NMI and RESET requests are not generated.

Figure 14.68 shows the OAD setting flow after reset. During this register setting, stop the Bus Master.

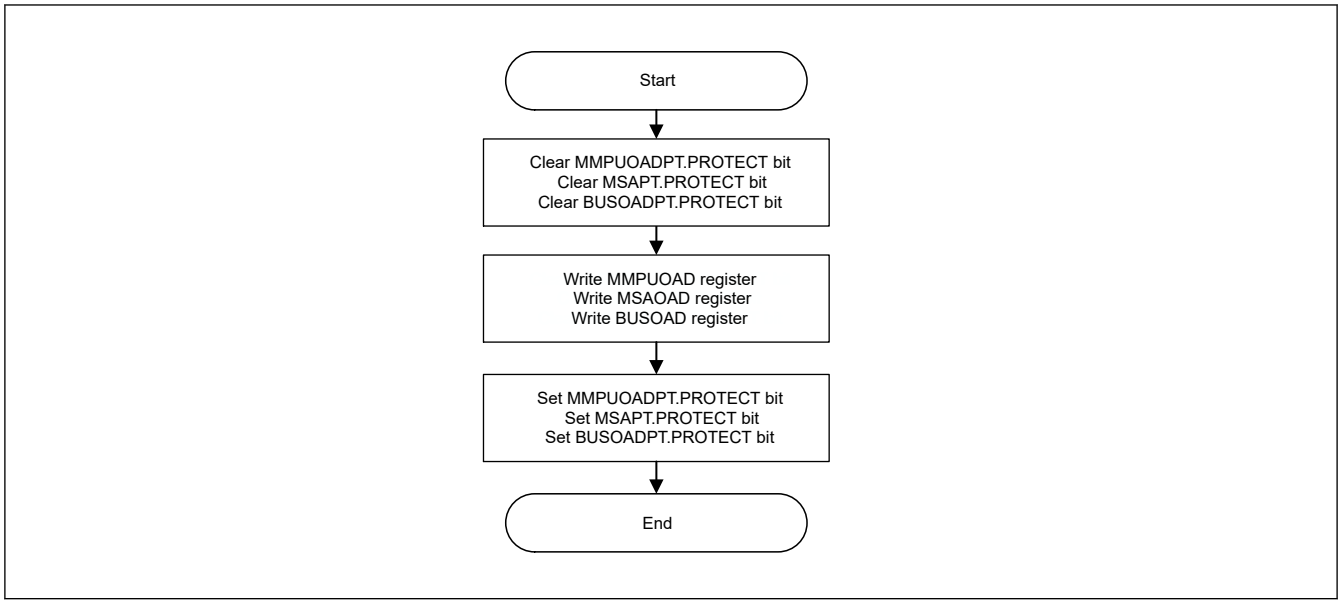


Figure 14.68 OAD setting flow after reset

For details, see section 15.3.1.41. MMPUOADPT : MMPU Operation After Detection Protect Register and section 15.3.1.40. MMPUOAD : MMPU Operation After Detection Register.

For details, see section 43.9.15. MSAPT : Master Security Attribution Protect Register and section 43.9.14. MSAOAD : Master Security Attribution Operation After Detection Register.

(1) Master Security Attribution Unit Error

The bus master of other than CPU and DAP has MSAU to block access to Secure alias address from Non-secure masters. The CPU does not have MSAU since it has IDAU and SAU. When MSAU Error is detected, an Error response is returned to the Master. At the same time, store the address when MSAU Error occurs in BMSAnERRADD (n = 4 to 9), store the RW information when MSAU Error occurs in BMSAnERRRW (n = 4 to 9), set 1 to MSERRSTAT bit of BUSnERRSTAT (n = 1 to 12). Furthermore, NMI request or Reset request is generated according to the setting of the MSAOAD.OAD. Since BMSAnERRADD (n = 4 to 9), BMSAnERRRW (n = 4 to 9), BUSnERRSTAT (n = 1 to 12) are held until reset other than Bus error reset and Common memory error reset, or cleared by BUSnERRCLR (n = 1 to 12), they can be confirmed in the NMI handler or after reset. NMI request is generated only at the first MSAU Error after reset or clearing BUSnERRSTAT (n = 1 to 12) .MSERRSTAT bit by BUSnERRCLR (n = 1 to 12).

(2) Bus Master MPU Error

The bus master of other than CPU and DAP has Master-MPU for access control of the set address area.The CPU does not have Master-MPU since it has Arm MPU. When Master-MPU Error is detected, an Error response is returned to the Master. At the same time, store the address when Master-MPU Error occurs in BUSnERRADD (n = 4 to 9), store the RW information when Master-MPU Error occurs in BUSnERRRW (n = 4 to 9), set 1 to MMERRSTAT bit of BUSnERRSTAT (n = 1 to 12). Furthermore, NMI request or Reset request is generated according to the setting of the MMPUOAD.OAD. Since BUSnERRADD (n = 4 to 9), BUSnERRRW (n = 4 to 9), BUSnERRSTAT (n = 1 to 12) are held until reset other than Bus error reset and Common memory error reset, or cleared by BUSnERRCLR (n = 1 to 12), they can be confirmed in the NMI handler or after reset. NMI request is generated only at the first Master-MPU Error after reset or clearing BUSnERRSTAT (n = 1 to 12) .MMERRSTAT bit by BUSnERRCLR (n = 1 to 12).

(3) Illegal Access Error

According to section 14.7.3. Conditions Leading to Illegal Address Access Errors, Illegal Access Error occurs. When Illegal Access Error is detected, an Error response is returned to the Master. When a master other than DAP caused Illegal address access error, store the Illegal Access Error occurrence address in BUSnERRADD (n = 4 to 9), store the Illegal Access Error occurrence RW information in BUSnERRRW (n = 4 to 9), set 1 to ILERRSTAT bit of BUSnERRSTAT (n = 1 to 12). Furthermore, NMI request or Reset request is generated according to the setting of the BUSOAD.ILERROAD. Since BUSnERRADD (n = 4 to 9), BUSnERRRW (n = 4 to 9), BUSnERRSTAT (n = 1 to 12) are held until reset other than Bus error reset and Common memory error reset, or cleared by BUSnERRCLR (n = 1 to 12), they can be confirmed in

the BusFault Handler or the NMI handler. NMI request is generated only at the first Illegal Error after reset or clearing BUSnERRSTAT (n = 1 to 12).ILERRSTAT bit by BUSnERRCLR (n = 1 to 12).

(4) Slave TrustZone Filter Error

When Slave TrustZone Filter Error is detected, an Error response is returned to the Master. When a master other than DAP caused Slave TrustZone Filter error, store the Slave TrustZone Filter error occurrence address in BMSAnERRADD (n = 4 to 9), store the Slave TrustZone Filter error occurrence RW information in BMSAnERRRW (n = 4 to 9), set 1 to SLERRSTAT bit of BUSnERRSTAT (n = 1 to 12). Furthermore, NMI request or Reset request is generated according to the setting of the BUSOAD.SLERROAD. Since BMSAnERRADD (n = 4 to 9), BMSAnERRRW (n = 4 to 9) and BUSnERRSTAT (n = 1 to 12) are held until reset other than Bus error reset and Common memory error reset, or cleared by BUSnERRCLR (n = 1 to 12), they can be confirmed in the NMI handler or after reset. NMI request is generated only at the first Slave TrustZone Filter Error after reset or clearing BUSnERRSTAT (n = 1 to 12).SLERRSTAT bit by BUSnERRCLR (n = 1 to 12).

(5) Slave Bus Error

Slave Bus Error occurs in the slave. When Slave Bus Error is detected, an Error response is returned to the Master. When a master other than DAP caused Slave Bus error, store the Slave Bus error occurrence address in BUSnERRADD (n = 4 to 9), store the Slave Bus error Occurrence RW information in BUSnERRRW (n = 4 to 9), set 1 to SLERRSTAT bit of BUSnERRSTAT (n = 1 to 12). Furthermore, NMI request or Reset request is generated according to the setting of the BUSOAD.SLERROAD. Since BUSnERRADD (n = 4 to 9), BUSnERRRW (n = 4 to 9), BUSnERRSTAT (n = 1 to 12) are held until reset other than Bus error reset and Common memory error reset, or cleared by BUSnERRCLR (n = 1 to 12), they can be confirmed in the BusFault Handler or NMI handler. NMI request is generated only at the first Slave-Bus Error after reset or clearing BUSnERRSTAT (n = 1 to 12).SLERRSTAT bit by BUSnERRCLR (n = 1 to 12).

14.7.3 Conditions Leading to Illegal Address Access Errors

Illegal address access error by bus access occurs in the address space shown in Table 14.49.

Table 14.49 Conditions leading to illegal address access errors (1 of 3)

Address		SLAVE bus	MASTER bus					
Start	End		CPU MAXI	CPU PAHB	DMAC/DTC	EDMAC	GLCDC0/GLCDC1/ MIPI/DRW0/ DRW1	CEU
0x0000_0000	0x00FF_FFFF	CPU SAHBI	E	N/A	—	E	E	E
0x0100_0000	0x01FF_FFFF	Reserved	E	N/A	E	E	E	E
0x0200_0000	0x03FF_FFFF	FHBI	—	N/A	—	E	—	E
0x0400_0000	0x0FFF_FFFF	Reserved	E	N/A	E	E	E	E
0x1000_0000	0x10FF_FFFF	CPU SAHBI	E	N/A	—	E	E	E
0x1100_0000	0x11FF_FFFF	Reserved	E	N/A	E	E	E	E
0x1200_0000	0x13FF_FFFF	FHBI	—	N/A	—	E	—	E
0x1400_0000	0x1FFF_FFFF	Reserved	E	N/A	E	E	E	E
0x2000_0000	0x20FF_FFFF	CPU SAHBI	E	N/A	—	E	E	E
0x2100_0000	0x21FF_FFFF	Reserved	E	N/A	E	E	E	E
0x2200_0000	0x2205_FFFF	S0BI	—	N/A	—	—	—	—
0x2206_0000	0x220D_FFFF	S1BI	—	N/A	—	—	—	—
0x220E_0000	0x25FF_FFFF	Reserved	E	N/A	E	E	E	E
0x2600_0000	0x2600_03FF	STBYSBI	—	N/A	—	—	E	E
0x2600_0400	0x26FF_FFFF	Reserved	E	N/A	E	E	E	E
0x2700_0000	0x2703_FFFF	FLBI	—	N/A	—	E	E	E
0x2704_0000	0x2FFF_FFFF	Reserved	E	N/A	E	E	E	E
0x3000_0000	0x30FF_FFFF	CPU SAHBI	E	N/A	—	E	E	E
0x3100_0000	0x31FF_FFFF	Reserved	E	N/A	E	E	E	E
0x3200_0000	0x3205_FFFF	S0BI	—	N/A	—	—	—	—

Table 14.49 Conditions leading to illegal address access errors (2 of 3)

Address		SLAVE bus	MASTER bus					
Start	End		CPU MAXI	CPU PAHB	DMAC/DTC	EDMAC	GLCDC0/GLCDC1/ MIPI/DRW0/ DRW1	CEU
0x3206_0000	0x320D_FFFF	S1BI	—	N/A	—	—	—	—
0x320E_0000	0x35FF_FFFF	Reserved	E	N/A	E	E	E	E
0x3600_0000	0x3600_03FF	STBYSBI	—	N/A	—	—	E	E
0x3600_0400	0x36FF_FFFF	Reserved	E	N/A	E	E	E	E
0x3700_0000	0x3703_FFFF	FLBI	—	N/A	—	E	E	E
0x3704_0000	0x3FFF_FFFF	Reserved	E	N/A	E	E	E	E
0x4000_0000	0x4000_0FFF	PSBI	N/A	—	—	E	E	E
0x4000_1000	0x4000_1FFF	Reserved	N/A	E*1	E	E	E	E
0x4000_2000	0x4000_4FFF	PSBI	N/A	—	—	E	E	E
0x4000_5000	0x4000_5FFF	Reserved	N/A	E*1	E	E	E	E
0x4000_6000	0x4000_6FFF	PSBI	N/A	—	—	E	E	E
0x4000_7000	0x4000_7FFF	Reserved	N/A	E*1	E	E	E	E
0x4000_8000	0x4000_8FFF	PSBI	N/A	—	—	E	E	E
0x4000_9000	0x4000_9FFF	Reserved	N/A	E*1	E	E	E	E
0x4000_A000	0x4000_AFFF	PSBI	N/A	—	—	E	E	E
0x4000_B000	0x4000_BFFF	Reserved	N/A	E*1	E	E	E	E
0x4000_C000	0x4000_CFFF	PSBI	N/A	—	E	E	E	E
0x4000_D000	0x4000_EFFF	Reserved	N/A	E*1	E	E	E	E
0x4000_F000	0x4001_CFFF	PSBI	N/A	—	—	E	E	E
0x4001_D000	0x4001_DFFF	Reserved	N/A	E*1	E	E	E	E
0x4001_E000	0x4001_EFFF	PSBI	N/A	—	—	E	E	E
0x4001_F000	0x400F_FFFF	Reserved	N/A	E*1	E	E	E	E
0x4010_0000	0x4011_FFFF	FLBI	N/A	—	—	E	E	E
0x4012_0000	0x401F_FFFF	Reserved	N/A	E*1	E	E	E	E
0x4020_0000	0x402F_FFFF	PBBI	N/A	—	—	E	E	E
0x4030_0000	0x403F_FFFF	PABI	N/A	—	—	E	E	E
0x4040_0000	0x404F_FFFF	PIBI	N/A	—	—	E	E	E
0x4050_0000	0x4FFF_FFFF	Reserved	N/A	E*1	E	E	E	E
0x5000_0000	0x5000_0FFF	PSBI	N/A	—	—	E	E	E
0x5000_1000	0x5000_1FFF	Reserved	N/A	E*1	E	E	E	E
0x5000_2000	0x5000_4FFF	PSBI	N/A	—	—	E	E	E
0x5000_5000	0x5000_5FFF	Reserved	N/A	E*1	E	E	E	E
0x5000_6000	0x5000_6FFF	PSBI	N/A	—	—	E	E	E
0x5000_7000	0x5000_7FFF	Reserved	N/A	E*1	E	E	E	E
0x5000_8000	0x5000_8FFF	PSBI	N/A	—	—	E	E	E
0x5000_9000	0x5000_9FFF	Reserved	N/A	E*1	E	E	E	E
0x5000_A000	0x5000_AFFF	PSBI	N/A	—	—	E	E	E
0x5000_B000	0x5000_BFFF	Reserved	N/A	E*1	E	E	E	E
0x5000_C000	0x5000_CFFF	PSBI	N/A	—	E	E	E	E

Table 14.49 Conditions leading to illegal address access errors (3 of 3)

Address		SLAVE bus	MASTER bus					
Start	End		CPU MAXI	CPU PAHB	DMAC/DTC	EDMAC	GLCDC0/GLCDC1/ MIPI/DRW0/ DRW1	CEU
0x5000_D000	0x5000_EFFF	Reserved	N/A	E*1	E	E	E	E
0x5000_F000	0x5000_FFFF	PSBI	N/A	—	—	E	E	E
0x5001_0000	0x5001_CFFF	PSBI	N/A	—	—	E	E	E
0x5001_D000	0x5001_DFFF	Reserved	N/A	E*1	E	E	E	E
0x5001_E000	0x5001_EFFF	PSBI	N/A	—	—	E	E	E
0x5001_F000	0x500F_FFFF	Reserved	N/A	E*1	E	E	E	E
0x5010_0000	0x5011_FFFF	FLBI	N/A	—	—	E	E	E
0x5012_0000	0x501F_FFFF	Reserved	N/A	E*1	E	E	E	E
0x5020_0000	0x502F_FFFF	PBBI	N/A	—	—	E	E	E
0x5030_0000	0x503F_FFFF	PABI	N/A	—	—	E	E	E
0x5040_0000	0x504F_FFFF	PIBI	N/A	—	—	E	E	E
0x5050_0000	0x5FFF_FFFF	Reserved	N/A	E*1	E	E	E	E
0x6000_0000	0x6FFF_FFFF	ECBI	—	N/A	—	—	—	—
0x7000_0000	0x7FFF_FFFF	Reserved	E	N/A	E	E	E	E
0x8000_0000	0x9FFF_FFFF	EOBI	—	N/A	—	—	—	—
0xA000_0000	0xFFFF_FFFF	Reserved	E	N/A	E	E	E	E

Note: “E” : A bus error is occurred.
“N/A“ : Transfer does not occur.
“—“ : A bus error has not occurred. Even if there is reserved area, a bus error has not occurred.
Do not access the reserved area in FLB, S0B and S1B. If accessed, a slave TZF error may occur.
Note 1. If CPU issues bufferable write accesses, a bufferable write access error occur instead of an illegal address access error.

14.7.4 Timeout

As for some peripheral module, a timeout error can occur in the module-stop function.

When there is no response from the slave for a certain time period, a timeout error is detected.

Timeout error is returned to the request master by Error response of the AHB-Lite protocol.

14.8 References

1. ARM Limited, *Arm v8-M Architecture Reference Manual*(ARM DDI0553B.g)
2. ARM Limited, *Arm® Cortex®-M85 Processor Integration and Implementation Manual Revision:r0p2* (ARM 101925_0002_05_en)
3. ARM Limited, *Arm® Cortex®-M85 Processor Technical Reference Manual Revision:r0p2* (ARM 101924_0002_05_en)
4. ARM Limited, *Arm® Cortex®-M85 Processor User Guide Reference Material Revision:r0p2* (ARM 101927_0002_05_en)
5. ARM Limited, *ARM AMBA 5 AHB Protocol Specification AHB5, AHB-Lite* (ARM IHI 0033B.b)
6. ARM Limited, *ARM AMBA AXI and ACE Protocol Specification AXI3, AXI4, and AXI4-Lite*, ACE and ACE-Lite (ARM IHI 0022H.c)
7. ARM Limited, *ARM AMBA APB Protocol Specification Version: 2.0* (ARM IHI 0024D)

14.9 Usage Notes

14.9.1 BC0 pin when in Software Standby Mode (SSTBY) and SBYCR.OPE=1

When entering Software Standby Mode with SBYCR.OPE = 1, the BC0 pin outputs the value of the least significant bit of the address signal most recently accessed. If you want to selectively output this value as a high-level or low-level, perform one of the following operations.

- If you want to output high-level, perform a dummy read of an odd address with an 8-bit data size before entering Software Standby Mode. If you want to output low-level, perform a dummy read of an even address before entering Software Standby Mode.
- Before entering Software Standby Mode, set BC0 pin to a general I/O port and output high-level or low-level. Set it toBC0 pin again after returning from Software Standby Mode.

If you do not specify a value for the BC0 pin in Software Standby Mode, these operations are not necessary.

15. Memory Protection Unit (MPU)

15.1 Overview

All bus masters have Memory Protection Units (MPUs).

Table 15.1 lists the MPU specifications, and Table 15.2 shows the behavior on detection of each MPU error.

Table 15.1 MPU specifications

Classification	Module/Function	Specifications
Memory protection	Arm MPU	Memory protection function for the CPU: • CPU: Secure MPU 8 regions and Non-secure MPU 8 regions
	Bus master MPU	Memory protection function for each bus master except for the CPU: • DMAC (DMAC/DTC): 8 regions • EDMAC (Ether-DMAC): 4 regions • GLCDC (GLCDC0/GLCDC1): 2 regions • DRW (DRW0/DRW1): 3 regions • MIPI DSI:1 region • CEU: 2 regions

Table 15.2 Behavior on MPU Error detection

MPU type	Access permissions setting	Boundary address setting minimum unit	Error response for the MPU error notification	Bus access at error detection	Hold the information of error access
Arm MPU	Read access Write access Execution	32 bytes	Supported*1	• Incorrectly write access • Incorrectly read access	Hold in CPU
Bus master MPU	Read access Write access Privileged access (DMAC/DTC only)	DMAC: 32 bytes EDMAC: 32 bytes GLCDC: 1 KB DRW: 1 KB MIPI DSI: 4 KB CEU: 4 KB	Supported	• Write access ignored • Read access is read as 0	Hold

Note 1. A privileged DAP request through the unprivileged debug extension mechanism is demoted to an unprivileged access and is subject to MPU checks. Both privileged and unprivileged requests are subject to MPU checks.

For information on error access for the Arm MPU, see section 15.4. References. For information on error access for other MPUs, see section 15.3.1. Register Descriptions and section 14.7. Bus Error Monitoring Section.

15.2 Arm MPU

The Arm MPU provides full support for:

- CPU: 8 secure regions and 8 non-secure regions
- Access permissions
- Export of memory attributes to the system

Arm MPU mismatches and permission violations invoke the programmable-priority MemManage fault (Hard Fault) handler. For details, see section 15.4. References.

15.3 Bus master MPU

The bus master MPU monitors the addresses accessed by the bus masters in the entire address space (0x0000_0000 to 0xFFFF_FFFF). The access control information can be set up to:

- 8 regions in DMAC/DTC
- 4 regions in EDMAC
- 2 regions in GLCDC
- 3 regions in DRW

- 1 region in MIPI DSI
- 2 regions in CEU

The bus master MPU monitors access to each region based on these settings.

If accesses violate the access permissions that are configured in bus master MPU, the bus master MPU returns an error response. For information on error access, see section 15.3.1. Register Descriptions and section 14.7.2. Operations When a Bus Error Occurs.

Table 15.3 lists the specifications of the bus master MPU.

Table 15.3 Bus master MPU specifications

Parameter	Description
Master groups	(Group name: Corresponding bus master) • DMAC: DMAC/DTC • EDMAC: Ether-DMAC • GLCDC: GLCDC0/GLCDC1 • DRW: DRW0/DRW1 • MIPI DSI: MIPI DSI • CEU: CEU
Protected regions	0x0000_0000 to 0xFFFF_FFFF
Number of regions	• DMAC: 8 regions • EDMAC: 4 regions • GLCDC: 2 regions • DRW: 3 regions • MIPI DSI: 1 region • CEU: 2 regions
Address specification for individual regions	• Region start and end addresses configurable
Enable or disable setting for memory protection in individual regions	• Settings enabled or disabled for the associated region
Access-control settings for individual regions	• Permission to read and write • Permission to unprivileged access
Operation on error detection	• Reset or error response
Register protection	• Register can be protected from illegal writes

15.3.1 Register Descriptions

Register writes processing should be performed after stopping the corresponding master group bus access.

SA (Secure Attribution) is set in each register and the SA is used to determine whether the target register is a secure register or a non-secure register.

15.3.1.1 MMPUSARA : Master Memory Protection Unit Security Attribution Register A

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x130

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	MMPUASAn[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	MMPUASAn[7:0]	MMPUA Security Attribution n (n = 0 to 7) 0: Secure 1: Non-secure	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1
P-TYPE-1

MMPUASAn[7:0] bits (MMPUA Security Attribution n (n = 0 to 7))

Security attributes of bus master MPU Region Setting register. The target registers are as follows:

- MMPUSDMACn (n = 0 to 7)
- MMPUEDMACn (n = 0 to 7)
- MMPUACDMACn (n = 0 to 7)

15.3.1.2 MMPUSARB : Master Memory Protection Unit Security Attribution Register B

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x134

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MMPUBSA8	—	—	—	—	—	—	—	MMPUBSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MMPUBSA0	MMPUB Security Attribution 0 0: Secure 1: Non-secure	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	MMPUBSA8	MMPUB Security Attribution 8 0: Secure 1: Non-secure	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1
P-TYPE-1

MMPUBSA0 bit (MMPUB Security Attribution 0)

Security attributes of DMAC MPU Enable Setting register. The target registers are as follows:

- MMPUENDMAC
- MMPUENPTDMAC

MMPUBSA8 bit (MMPUB Security Attribution 8)

Security attributes of MPU Operation After Detection Setting register. The target registers are as follows:

- MMPUOAD
- MMPUOADPT

15.3.1.3 MMPUSDMACn : MMPU Start Address Register for DMAC (n = 0 to 7)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0204 + 0x010 × n

Bit position:	31																									0
Bit field:																										
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Region start address register Address where the region starts, for use in region determination. The starting address of the MPU area must be set in the range of 0x0000_0000 to 0xFFFF_FFE0. Writing is ignored for bit 0 to bit 4. These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3
P-TYPE-2

The MMPUSDMACn (n = 0 to 7) register specifies the start address where the region starts.

This register requires word access. Byte access and halfword access is prohibited. When byte access and halfword access is executed, operation is not guaranteed.

Regions set by MMPUSDMACn (n = 0 to 7), MMPUEDMACn (n = 0 to 7) and MMPUACDMACn (n = 0 to 7) registers, can be set for a secure access or a non-secure access with the MMPUSARA register. If the corresponding MMPUSARA.MMPUASAn (n = 0 to 7) bit is set to 1, it is only possible to use that region with a non-secure access. On the other hand, if the corresponding MMPUSARA.MMPUASAn (n = 0 to 7) bit is set to 0, it is only possible to use that region with a secure access.

15.3.1.4 MMPUSEDMACn : MMPU Start Address Register for EDMAC (n = 0 to 3)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0604+ 0x010 × n

Bit position:	31																									0
Bit field:																										
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Region start address register for EDMAC Address where the region starts, for use in region determination. The starting address of the MPU area must be set in the range of 0x0000_0000 to 0xFFFF_FFE0. Writing is ignored for bit 0 to bit 4. These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3
P-TYPE-2

The MMPUSEDMACn (n = 0 to 3) register specifies the start address where the region starts.

This register requires word access. Byte access and halfword access is prohibited. When byte access and halfword access is executed, operation is not guaranteed.

15.3.1.5 MMPUSGLCDCn : MMPU Start Address Register for GLCDC (n = 0, 1)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0804+ 0x010 × n

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Region start address register for GLCDC Address where the region starts, for use in region determination. The starting address of the MPU area must be set in the range of 0x0000_0000 to 0xFFFF_FC00. Writing is ignored for bit 0 to bit 9. These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3
P-TYPE-2

The MMPUSGLCDCn (n = 0, 1) register specifies the start address where the region starts.

This register requires word access. Byte access and halfword access is prohibited. When byte access and halfword access is executed, operation is not guaranteed.

15.3.1.6 MMPUSDRWn : MMPU Start Address Register for DRW (n = 0 to 2)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0A04+ 0x010 × n

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Region start address register for DRW Address where the region starts, for use in region determination. The starting address of the MPU area must be set in the range of 0x0000_0000 to 0xFFFF_FC00. Writing is ignored for bit 0 to bit 9. These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3
P-TYPE-2

The MMPUSDRWn (n = 0 to 2) register specifies the start address where the region starts.

This register requires word access. Byte access and halfword access is prohibited. When byte access and halfword access is executed, operation is not guaranteed.

15.3.1.7 MMPUSMIPI : MMPU Start Address Register for MIPI DSI

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0C04

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Region start address register for MIPI DSI Address where the region starts, for use in region determination. The starting address of the MPU area must be set in the range of 0x0000_0000 to 0xFFFF_F000. Writing is ignored for bit 0 to bit 11. These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3
P-TYPE-2

The MMPUSMIPI register specifies the start address where the region starts.

This register requires word access. Byte access and halfword access is prohibited. When byte access and halfword access is executed, operation is not guaranteed.

15.3.1.8 MMPUSCEUn : MMPU Start Address Register for CEU (n = 0 to 1)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0E04+ 0x010 × n

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Region start address register for CEU Address where the region starts, for use in region determination. The starting address of the MPU area must be set in the range of 0x0000_0000 to 0xFFFF_F000. Writing is ignored for bit 0 to bit 11. These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3
P-TYPE-2

The MMPUSCEUn (n = 0 to 1) register specifies the start address where the region starts.

This register requires word access. Byte access and halfword access is prohibited. When byte access and halfword access is executed, operation is not guaranteed.

15.3.1.9 MMPUEDMACn : MMPU End Address Register for DMAC (n = 0 to 7)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0208+ 0x010 × n

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	Region end address register Address where the region end, for use in region determination. The ending address of the MPU area must be set in the range of 0x0000_001F to 0xFFFF_FFFF. Writing is ignored for bit 0 to bit 4. These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-3
P-TYPE-2

The MMPUEDMACn (n = 0 to 7) register specifies the end address where the region ends.

This register requires word access. Byte access and halfword access is prohibited. When byte access and halfword access is executed, operation is not guaranteed.

15.3.1.10 MMPUEEDMACn : MMPU End Address Register for EDMAC (n = 0 to 3)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0608+ 0x010 × n

Bit position:	31																								0
Bit field:																									
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	Region end address register for EDMAC Address where the region ends, for use in region determination. The ending address of the MPU area must be set in the range of 0x0000_001F to 0xFFFF_FFFF. Writing is ignored for bit 0 to bit 4. These bits are read as 1.The write value should be 1.	R/W

Note: S-TYPE-3
P-TYPE-2

The MMPUEEDMACn (n = 0 to 3) register specifies the end address where the region ends.

This register requires word access. Byte access and halfword access is prohibited. When byte access and halfword access is executed, operation is not guaranteed.

15.3.1.11 MMPUEGLCDCn : MMPU End Address Register for GLCDC (n = 0 to 1)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0808+ 0x010 × n

Bit position:	31																								0
Bit field:																									
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	Region end address register for GLCDC Address where the region ends, for use in region determination. The ending address of the MPU area must be set in the range of 0x0000_03FF to 0xFFFF_FFFF. Writing is ignored for bit 0 to bit 9. These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-3
P-TYPE-2

The MMPUEGLCDCn (n = 0 to 1) register specifies the end address where the region ends.

This register requires word access. Byte access and halfword access is prohibited. When byte access and halfword access is executed, operation is not guaranteed.

15.3.1.12 MMPUEDRWn : MMPU End Address Register for DRW (n = 0 to 2)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0A08+ 0x010 × n

Bit position:	31																								0
Bit field:																									
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	Region end address register for DRW Address where the region ends, for use in region determination. The ending address of the MPU area must be set in the range of 0x0000_03FF to 0xFFFF_FFFF. Writing is ignored for bit 0 to bit 9. These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-3
P-TYPE-2

The MMPUEDRWn (n = 0 to 2) register specifies the end address where the region ends.

This register requires word access. Byte access and halfword access is prohibited. When byte access and halfword access is executed, operation is not guaranteed.

15.3.1.13 MMPUEMIPI : MMPU End Address Register for MIPI DSI

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0C08

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	Region end address register for MIPI DSI Address where the region ends, for use in region determination. The ending address of the MPU area must be set in the range of 0x0000_0FFF to 0xFFFF_FFFF. Writing is ignored for bit 0 to bit 11. These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-3
P-TYPE-2

The MMPUEMIPI register specify the end address where the region ends.

This register requires word access. Byte access and halfword access is prohibited. When byte access and halfword access is executed, operation is not guaranteed.

15.3.1.14 MMPUECEUn : MMPU End Address Register for CEU (n = 0 to 1)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0E08+ 0x010 × n

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	Region end address register for CEU Address where the region ends, for use in region determination. The ending address of the MPU area must be set in the range of 0x0000_0FFF to 0xFFFF_FFFF. Writing is ignored for bit 0 to bit 11. These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-3
P-TYPE-2

The MMPUECEUn (n = 0 to 1) register specifies the end address where the region ends.

This register requires word access. Byte access and halfword access is prohibited. When byte access and halfword access is executed, operation is not guaranteed.

15.3.1.15 MMPUACDMACn : MMPU Access Control Register for DMAC (n = 0 to 7)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0200+ 0x010 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PP	WP	RP	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	Region Enable 0: DMAC region n unit is disabled. 1: DMAC region n unit is enabled.	R/W
1	RP	Read protection 0: Read permission 1: Read protection	R/W
2	WP	Write protection 0: Write permission 1: Write protection	R/W
3	PP	Privilege protection 0: Unprivileged access permission 1: Unprivileged access protection	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3
P-TYPE-2

ENABLE bit (Region Enable)

The ENABLE bit enables or disables the DMAC region n (n = 0 to 7) unit.

When the ENABLE bit is set to 1, the RP bit, the WP bit and the PP bit are enabled for permit or protect access to the region that is set in MMPUSDMACn (n = 0 to 7) and MMPUEDMACn (n = 0 to 7).

When the ENABLE bit is set to 0, the access to DMAC region n (n = 0 to 7) is not specified.

RP bit (Read protection)

The RP bit enables or disables read protection for DMAC region n (n = 0 to 7).

When the ENABLE bit is set to 1, the RP bit is available.

WP bit (Write protection)

The WP bit enables or disables write protection for DMAC region n (n = 0 to 7).

When the ENABLE bit is set to 1, the WP bit is available.

PP bit (Privilege protection)

The PP bit enables or disables unprivileged access protection for DMAC region n (n = 0 to 7).

When the ENABLE bit is set to 1, the PP bit is available.

15.3.1.16 MMPUACEDMACn : MMPU Access Control Register for EDMAC (n = 0 to 3)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0600+ 0x010 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	WP	RP	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	Region Enable 0: EDMAC region n unit is disabled. 1: EDMAC region n unit is enabled.	R/W
1	RP	Read protection 0: Read permission 1: Read protection	R/W
2	WP	Write protection 0: Write permission 1: Write protection	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3
P-TYPE-2

ENABLE bit (Region Enable)

The ENABLE bit enables or disables the EDMAC region n (n = 0 to 3) unit.

When the ENABLE bit is set to 1, the RP bit and the WP bit are enabled for permit or protect access to the region that is set in MMPUSEDMACn (n = 0 to 3) and MMPUEEDMACn (n = 0 to 3).

When the ENABLE bit is set to 0, the access to EDMAC region n (n = 0 to 3) is not specified.

RP bit (Read protection)

The RP bit enables or disables read protection for EDMAC region n (n = 0 to 3).

When the ENABLE bit is set to 1, the RP bit is available.

WP bit (Write protection)

The WP bit enables or disables write protection for EDMAC region n (n = 0 to 3).

When the ENABLE bit is set to 1, the WP bit is available.

15.3.1.17 MMPUACGLCDCn : MMPU Access Control Register for GLCDC (n = 0, 1)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0800 + 0x010 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	WP	RP	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	Region Enable 0: GLCDC region n unit is disabled. 1: GLCDC region n unit is enabled.	R/W

Bit	Symbol	Function	R/W
1	RP	Read protection 0: Read permission 1: Read protection	R/W
2	WP	Write protection 0: Write permission 1: Write protection	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3
P-TYPE-2

ENABLE bit (Region Enable)

The ENABLE bit enables or disables the GLCDC region n (n = 0, 1) unit.

When the ENABLE bit is set to 1, the RP bit and the WP bit are enabled for permit or protect access to the region that is set in MMPUSGLCDCn (n = 0, 1) and MMPUEGLCDCn (n = 0, 1).

When the ENABLE bit is set to 0, the access to GLCDC region n (n = 0, 1) is not specified.

RP bit (Read protection)

The RP bit enables or disables read protection for GLCDC region n (n = 0, 1).

When the ENABLE bit is set to 1, the RP bit is available.

WP bit (Write protection)

The WP bit enables or disables write protection for GLCDC region n (n = 0, 1).

When the ENABLE bit is set to 1, the WP bit is available.

15.3.1.18 MMPUACDRWn : MMPU Access Control Register for DRW (n = 0 to 2)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0A00+ 0x010 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	WP	RP	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	Region Enable 0: DRW region n unit is disabled. 1: DRW region n unit is enabled.	R/W
1	RP	Read protection 0: Read permission 1: Read protection	R/W
2	WP	Write protection 0: Write permission 1: Write protection	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3
P-TYPE-2

ENABLE bit (Region Enable)

The ENABLE bit enables or disables the DRW region n (n = 0 to 2) unit.

When the ENABLE bit is set to 1, the RP bit and the WP bit are enabled for permit or protect access to the region that is set in MMPUSDRWn (n = 0 to 2) and MMPUEDRWn (n = 0 to 2).

When the ENABLE bit is set to 0, the access to DRW region n (n = 0 to 2) is not specified.

RP bit (Read protection)

The RP bit enables or disables read protection for DRW region n (n = 0 to 2).

When the ENABLE bit is set to 1, the RP bit is available.

WP bit (Write protection)

The WP bit enables or disables write protection for DRW region n (n = 0 to 2).

When the ENABLE bit is set to 1, the WP bit is available.

15.3.1.19 MMPUACMIPI : MMPU Access Control Register for MIPI DSI

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0C00

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	WP	RP	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	Region Enable 0: MIPI DSI region unit is disabled. 1: MIPI DSI region unit is enabled.	R/W
1	RP	Read protection 0: Read permission 1: Read protection	R/W
2	WP	Write protection 0: Write permission 1: Write protection	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3
P-TYPE-2

ENABLE bit (Region Enable)

The ENABLE bit enables or disables the MIPI DSI region unit.

When the ENABLE bit is set to 1, the RP bit and the WP bit are enabled for permit or protect access to the region that is set in MMPUSMIPI and MMPUEMIPI.

When the ENABLE bit is set to 0, the access to MIPI DSI region is not specified.

RP bit (Read protection)

The RP bit enables or disables read protection for MIPI DSI region.

When the ENABLE bit is set to 1, the RP bit is available.

WP bit (Write protection)

The WP bit enables or disables write protection for MIPI DSI region.

When the ENABLE bit is set to 1, the WP bit is available.

15.3.1.20 MMPUACCEUn : MMPU Access Control Register for CEU (n = 0, 1)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0E00+ 0x010 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	WP	RP	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	Region Enable 0: CEU region n unit is disabled. 1: CEU region n unit is enabled.	R/W
1	RP	Read protection 0: Read permission 1: Read protection	R/W
2	WP	Write protection 0: Write permission 1: Write protection	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3
P-TYPE-2

ENABLE bit (Region Enable)

The ENABLE bit enables or disables the CEU region n (n = 0, 1) unit.

When the ENABLE bit is set to 1, the RP bit and the WP bit are enabled for permit or protect access to the region that is set in MMPUSCEUn (n = 0, 1) and MMPUECEUn (n = 0, 1).

When the ENABLE bit is set to 0, the access to CEU region n (n = 0, 1) is not specified.

RP bit (Read protection)

The RP bit enables or disables read protection for CEU region n (n = 0, 1).

When the ENABLE bit is set to 1, the RP bit is available.

WP bit (Write protection)

The WP bit enables or disables write protection for CEU region n (n = 0, 1).

When the ENABLE bit is set to 1, the WP bit is available.

15.3.1.21 MMPUENDMAC : MMPU Enable Register for DMAC

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0100

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	Bus master MPU of DMAC Enable 0: Bus master MPU of DMAC is disabled. 1: Bus master MPU of DMAC is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing of the ENABLE bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access. Byte-write access is prohibited.
When byte-write access is executed, operation is not guaranteed.

ENABLE bit (Bus master MPU of DMAC Enable)

The ENABLE bit enables or disables the bus master MPU function for DMAC.

When the ENABLE bits is set to 1, MMPUACDMACn (n = 0 to 7) is available.

When the ENABLE bits is set to 0, MMPUACDMACn (n = 0 to 7) is unavailable, including permission for all regions.

The bus master MPU function of each master group sets the ENABLE bit.

When the ENABLE bit is set, write 0xA5 in KEY[7:0] at the same time.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing of the ENABLE.

When writing the ENABLE bit, write 0xA5 in KEY[7:0] at the same time.

When a value other than 0xA5 is written in KEY[7:0] bits, the ENABLE bit is not updated.The KEY[7:0] bits are always read as 0x00.

15.3.1.22 MMPUENEDMAC : MMPU Enable Register for EDMAC

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0500

Bit position:	15							8								0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	Bus master MPU of EDMAC Enable 0: Bus master MPU of EDMAC is disabled. 1: Bus master MPU of EDMAC is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the ENABLE bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access.
Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

ENABLE bit (Bus master MPU of EDMAC Enable)

The ENABLE bit enables or disables the bus master MPU function of each master group.

When the ENABLE bit is set to 1, MMPUACEDMACn (n = 0 to 3) is available.

When the ENABLE bit is set to 0, MMPUACEDMACn (n = 0 to 3) is unavailable, and permission for all regions.

The bus master MPU function of each master group sets the ENABLE bit.

When the ENABLE bit is set, write 0xA5 to the KEY[7:0] bits at the same time.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing to the ENABLE bit.

When writing the ENABLE bit, write 0xA5 to the KEY[7:0] bits at the same time.

When a value other than 0xA5 is written to the KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.1.23 MMPUENGLCDC : MMPU Enable Register for GLCDC

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0700

Bit position:	15							8								0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	Bus master MPU of GLCDC Enable 0: Bus master MPU of GLCDC is disabled. 1: Bus master MPU of GLCDC is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the ENABLE bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access.
Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

ENABLE bit (Bus master MPU of GLCDC Enable)

The ENABLE bit enables or disables the bus master MPU function of each master group.

When the ENABLE bit is set to 1, MMPUACGLCDCn (n = 0, 1) is available.

When the ENABLE bit is set to 0, MMPUACGLCDCn (n = 0, 1) is unavailable, and permission for all regions.

The bus master MPU function of each master group sets the ENABLE bit.

When the ENABLE bit is set, write 0xA5 to the KEY[7:0] bits at the same time.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing to the ENABLE bit.

When writing the ENABLE bit, write 0xA5 to the KEY[7:0] bits at the same time.

When a value other than 0xA5 is written to the KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.1.24 MMPUENDRW : MMPU Enable Register for DRW

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0900

Bit position:	15							8								0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	Bus master MPU of DRW Enable 0: Bus master MPU of DRW is disabled. 1: Bus master MPU of DRW is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the ENABLE bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access.
Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

ENABLE bit (Bus master MPU of DRW Enable)

The ENABLE bit enables or disables the bus master MPU function of each master group.

When the ENABLE bit is set to 1, MMPUACDRWn (n = 0 to 2) is available.

When the ENABLE bit is set to 0, MMPUACDRWn (n = 0 to 2) is unavailable, and permission for all regions.

The bus master MPU function of each master group sets the ENABLE bit.

When the ENABLE bit is set, write 0xA5 to the KEY[7:0] bits at the same time.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing to the ENABLE bit.

When writing the ENABLE bit, write 0xA5 to the KEY[7:0] bits at the same time.

When a value other than 0xA5 is written to the KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.1.25 MMPUENMIPI : MMPU Enable Register for MIPI DSI

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0B00

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	Bus master MPU of MIPI DSI Enable 0: Bus master MPU of MIPI DSI is disabled. 1: Bus master MPU of MIPI DSI is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing of the ENABLE bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access.
Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

ENABLE bit (Bus master MPU of MIPI DSI Enable)

The ENABLE bit enables or disables the bus master MPU function of each master group.

When the ENABLE bits is set to 1, MMPUACMIPI is available.

When the ENABLE bits is set to 0, MMPUACMIPI is unavailable, and permission for all regions.

The bus master MPU function of each master group sets the ENABLE bit.

When the ENABLE bit is set, write 0xA5 in KEY[7:0] at the same time.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing of the ENABLE.

When writing the ENABLE bit, write 0xA5 in KEY[7:0] at the same time.

When a value other than 0xA5 is written in KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.1.26 MMPUENCEU : MMPU Enable Register for CEU

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0D00

Bit position:	15							8								0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	Bus master MPU of CEU Enable 0: Bus master MPU of CEU is disabled. 1: Bus master MPU of CEU is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the ENABLE bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access.
Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

ENABLE bit (Bus master MPU of CEU Enable)

The ENABLE bit enables or disables the bus master MPU function of each master group.

When the ENABLE bit is set to 1, MMPUACCEUn (n = 0, 1) is available.

When the ENABLE bit is set to 0, MMPUACCEUn (n = 0, 1) is unavailable, and permission for all regions.

The bus master MPU function of each master group sets the ENABLE bit.

When the ENABLE bit is set, write 0xA5 to the KEY[7:0] bits at the same time.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing to the ENABLE bit.

When writing the ENABLE bit, write 0xA5 to the KEY[7:0] bits at the same time.

When a value other than 0xA5 is written to the KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.1.27 MMPUENPTDMAC : MMPU Enable Protect Register for DMAC

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0104

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MMPUENDMAC register write is possible. 1: MMPUENDMAC register write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing of the PROTECT bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access.
Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit controls protection of MMPUENDMAC register.

When the PROTECT bit is set, write 0xA5 in KEY[7:0] at the same time.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing of the PROTECT.

When writing the PROTECT bit, write 0xA5 in KEY[7:0] at the same time.

When a value other than 0xA5 is written in KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.1.28 MMPUENPTEDMAC : MMPU Enable Protect Register for EDMAC

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0504

Bit position:	15	8	0
Bit field:	KEY[7:0]	— — — — — — —	PROTECT
Value after reset:	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MMPUENEDMAC register write is possible. 1: MMPUENEDMAC register write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access.

Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit controls protection of MMPUENEDMAC register.

When the PROTECT bit is set, write 0xA5 to the KEY[7:0] bits at the same time.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing to the PROTECT bit.

When writing the PROTECT bit, write 0xA5 to the KEY[7:0] bits at the same time.

When a value other than 0xA5 is written to the KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.1.29 MMPUENPTGLCDC : MMPU Enable Protect Register for GLCDC

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0704

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MMPUENGLCDC register write is possible. 1: MMPUENGLCDC register write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access.

Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit controls protection of MMPUENGLCDC register.

When the PROTECT bit is set, write 0xA5 to the KEY[7:0] bits at the same time.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing to the PROTECT bit.

When writing the PROTECT bit, write 0xA5 to the KEY[7:0] bits at the same time.

When a value other than 0xA5 is written to the KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.1.30 MMPUENPDRW : MMPU Enable Protect Register for DRW

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0904

Bit position:	15							8								0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MMPUENDRW register write is possible. 1: MMPUENDRW register write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access.
Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit controls protection of MMPUENDRW register.

When the PROTECT bit is set, write 0xA5 to the KEY[7:0] bits at the same time.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing to the PROTECT bit.

When writing the PROTECT bit, write 0xA5 to the KEY[7:0] bits at the same time.

When a value other than 0xA5 is written to the KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.1.31 MMPUENPTMIPI : MMPU Enable Protect Register for MIPI DSI

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0B04

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MMPUENMIPI register write is possible. 1: MMPUENMIPI register write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing of the PROTECT bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access.

Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit controls protection of MMPUENMIPI register.

When the PROTECT bit is set, write 0xA5 in KEY[7:0] at the same time.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing of the PROTECT.

When writing the PROTECT bit, write 0xA5 in KEY[7:0] at the same time.

When a value other than 0xA5 is written in KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.1.32 MMPUENPTCEU : MMPU Enable Protect Register for CEU

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0D04

Bit position:	15							8								0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MMPUENCEU register write is possible. 1: MMPUENCEU register write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access.

Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit controls protection of MMPUENCEU register.

When the PROTECT bit is set, write 0xA5 to the KEY[7:0] bits at the same time.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing to the PROTECT bit.

When writing the PROTECT bit, write 0xA5 to the KEY[7:0] bits at the same time.

When a value other than 0xA5 is written to the KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.1.33 MMPURPTDMAC : MMPU Regions Protect Register for DMAC Non-secure

Base address: RMPU_NS = 0x5000_0000

Offset address: 0x0108

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Bus master MPU register for DMAC write is possible. 1: Bus master MPU register for DMAC write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing of the PROTECT bit.	W

Note: S-TYPE-7
P-TYPE-2

Note: It is necessary to write by halfword access.
Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writes to the associated registers to be protected.

MMPURPTDMAC.PROTECT controls the following registers:

- MMPUSDMACn (n = 0 to 7) of Non-secure
- MMPUEDMACn (n = 0 to 7) of Non-secure
- MMPUACDMACn (n = 0 to 7) of Non-secure

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits, using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When a value other than 0xA5 is written in KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.1.34 MMPURPTDMAC_SEC : MMPU Regions Protect register for DMAC Secure

Base address: RMPU = 0x4000_0000

Offset address: 0x010C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Bus master MPU register for DMAC Secure write is possible. 1: Bus master MPU register for DMAC Secure write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing of the PROTECT bit.	W

Note: S-TYPE-6
P-TYPE-2

Note: It is necessary to write by halfword access.
Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writes to the associated registers to be protected.

MMPURPTDMAC_SEC.PROTECT controls the following registers:

- MMPUSDMACn (n = 0 to 7) of Secure
- MMPUEDMACn (n = 0 to 7) of Secure
- MMPUACDMACn (n = 0 to 7) of Secure

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits, using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When a value other than 0xA5 is written in KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.1.35 MMPURPTEDMAC : MMPU Regions Protect Register for EDMAC

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0508

Bit position:	15							8								0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Bus master MPU register for EDMAC write is possible. 1: Bus master MPU register for EDMAC write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access.
Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writing to the associated registers to be protected.

MMPURPTEDMAC.PROTECT controls the following registers:

- MMPUSEDMACn (n = 0 to 3)
- MMPUEEDMACn (n = 0 to 3)
- MMPUACEDMACn (n = 0 to 3)

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When a value other than 0xA5 is written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.1.36 MMPURPTGLCDC : MMPU Regions Protect Register for GLCDC

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0708

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Bus master MPU register for GLCDC write is possible. 1: Bus master MPU register for GLCDC write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access.
Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writing to the associated registers to be protected.

MMPURPTGLCDC.PROTECT controls the following registers:

- MMPUSGLCDCn (n = 0, 1)
- MMPUEGLCDCn (n = 0, 1)
- MMPUACGLCDCn (n = 0, 1)

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When a value other than 0xA5 is written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.1.37 MMPURPTDRW : MMPU Regions Protect Register for DRW

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0908

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Bus master MPU register for DRW write is possible. 1: Bus master MPU register for DRW write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access.
Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writing to the associated registers to be protected.

MMPURPDRW.PROTECT controls the following registers:

- MMPUSDRWn (n = 0 to 2)
- MMPUEDRWn (n = 0 to 2)
- MMPUACDRWn (n = 0 to 2)

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When a value other than 0xA5 is written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.1.38 MMPURPTMIPI : MMPU Regions Protect Register for MIPI DSI

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0B08

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Bus master MPU register for MIPI DSI write is possible. 1: Bus master MPU register for MIPI DSI write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing of the PROTECT bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access.
Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writes to the associated registers to be protected.

MMPURPTMIPI.PROTECT controls the following registers:

- MMPUSMIPI
- MMPUEMIPI
- MMPUACMIPI

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits, using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When a value other than 0xA5 is written in KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.1.39 MMPURPTCEU : MMPU Regions Protect Register for CEU

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0D08

Bit position:	15							8								0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Bus master MPU register for CEU write is possible. 1: Bus master MPU register for CEU write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access.
Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writing to the associated registers to be protected.

MMPURPTCEU.PROTECT controls the following registers:

- MMPUSCEUn (n = 0, 1)
- MMPUECEUn (n = 0, 1)
- MMPUACCEUn (n = 0, 1)

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When a value other than 0xA5 is written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.1.40 MMPUOAD : MMPU Operation After Detection Register

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0000

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OAD	Operation after detection 0: NMI 1: Reset	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing of the OAD bit.	R/W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access.
Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

OAD bit (Operation after detection)

The OAD bit specify operation when the access violation is detected.

When OAD bit is 0, error response is returned and NMI is generated.

When OAD bit is 1, reset request is generated.

When writing to the OAD bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the OAD bit. When writing to the OAD bit, write 0xA5 simultaneously to the KEY[7:0] bits. When other values are written, the OAD bit is not updated.

The KEY[7:0] bits always read as 0x00.

15.3.1.41 MMPUOADPT : MMPU Operation After Detection Protect Register

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0004

Bit position:	15							8								0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MMPUOAD register writing is possible. 1: MMPUOAD register writing is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access.
Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writing to the associated registers to be protected.

MMPUOAD.PROTECT controls the following registers:

- MMPUOAD

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When other values are written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.2 Operation

15.3.2.1 Memory protection

The bus master MPU monitors memory access using control settings made individually for the access control regions. If accesses violate the access permissions that are configured in bus master MPU, the bus master MPU generates a memory protection error.

Bus master MPU has master groups of DMAC, EDMAC, GLCDC, DRW, MIPI DSI and CEU. The memory protection function checks the address of the bus for a unified master group, and blocks illegal access of the master group to the protected region by bus master MPU.

The region setting registers of the bus master MPU for DMAC can be set for secure master and non-secure master using the MMPUSARA register. The MPU region setting with MMPUSARA.MMPUASAn = 0 applies only to access from the secure master, and the MPU region setting with MMPUSARA.MMPUASAn = 1 applies only to access from the non-secure master. Figure 15.1 shows an example of access authorization for the secure master and non-secure master when both secure and non-secure MPU settings are configured.

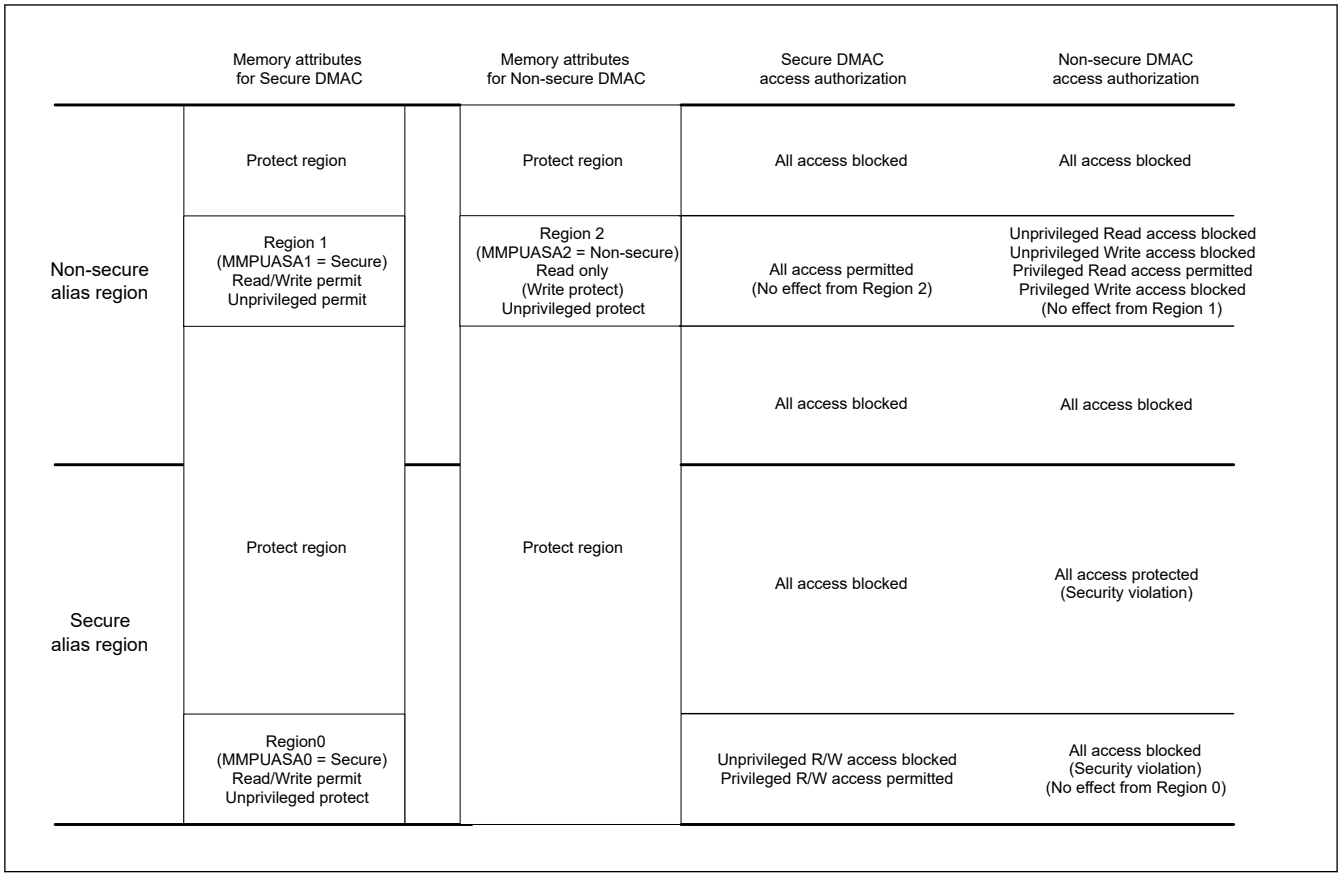


Figure 15.1 Example of access authority when both secure and non-secure region of MPUs are set

Bus master MPU permit of all regions after reset. If MMPUENXXXX.ENABLE is 1 and there is no MPU region setting for the corresponding bus master, all regions are protected. (XXXX = Master Group name)

Figure 15.2 shows the access permission or protection by the overlapping bus master MPU regions.

Access control for the overlapping regions is as follows:

- The region is handled as a protected region when output of one or more region units is a protected region
- The region is handled as a protected region when output of all region units is outside of the regions
- Other cases are handled as permitted regions.

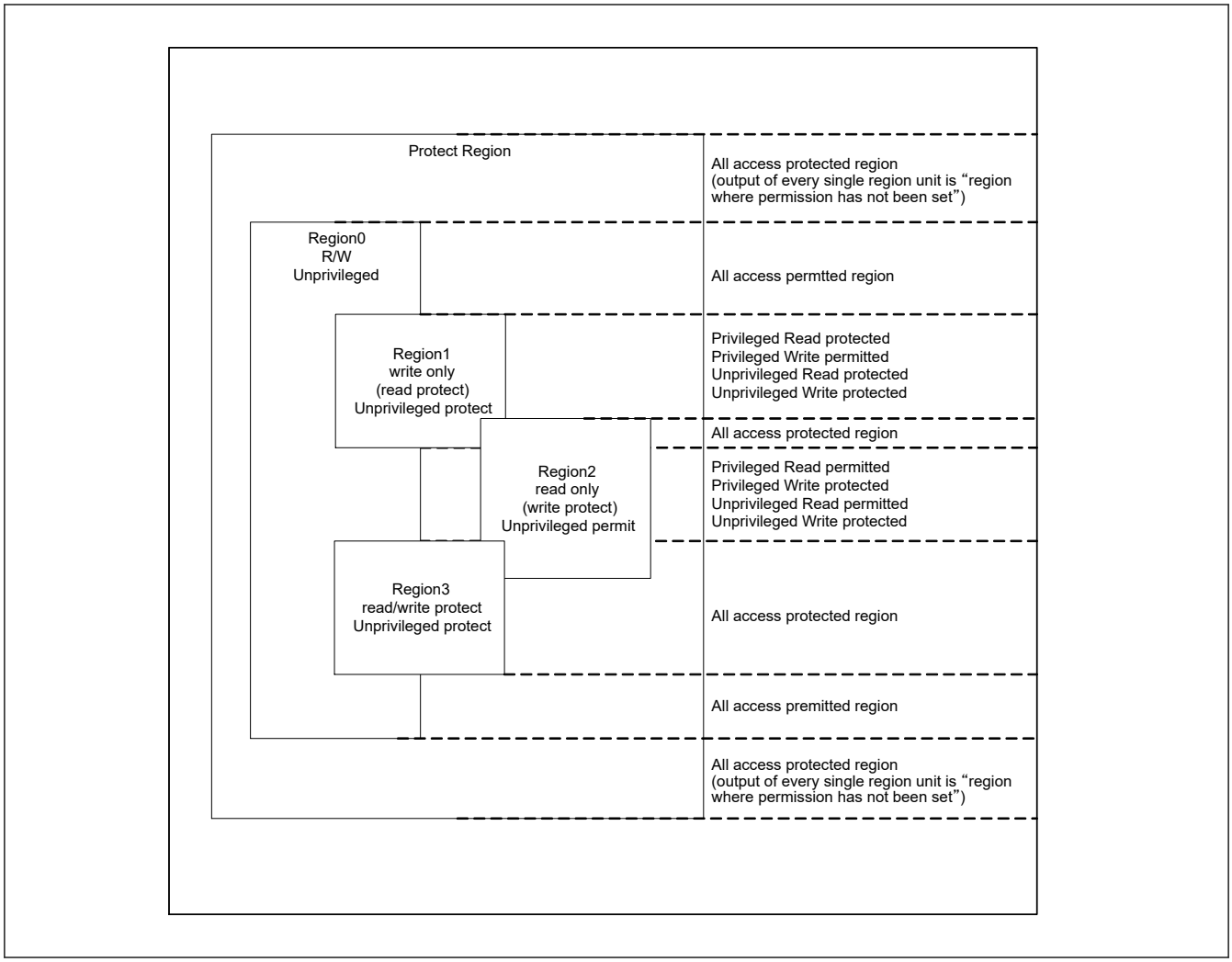


Figure 15.2 Access permission or protection by overlap of the bus master MPU regions

Figure 15.3 shows the register setting flow after reset. During this register setting, stop the bus master except the CPU.

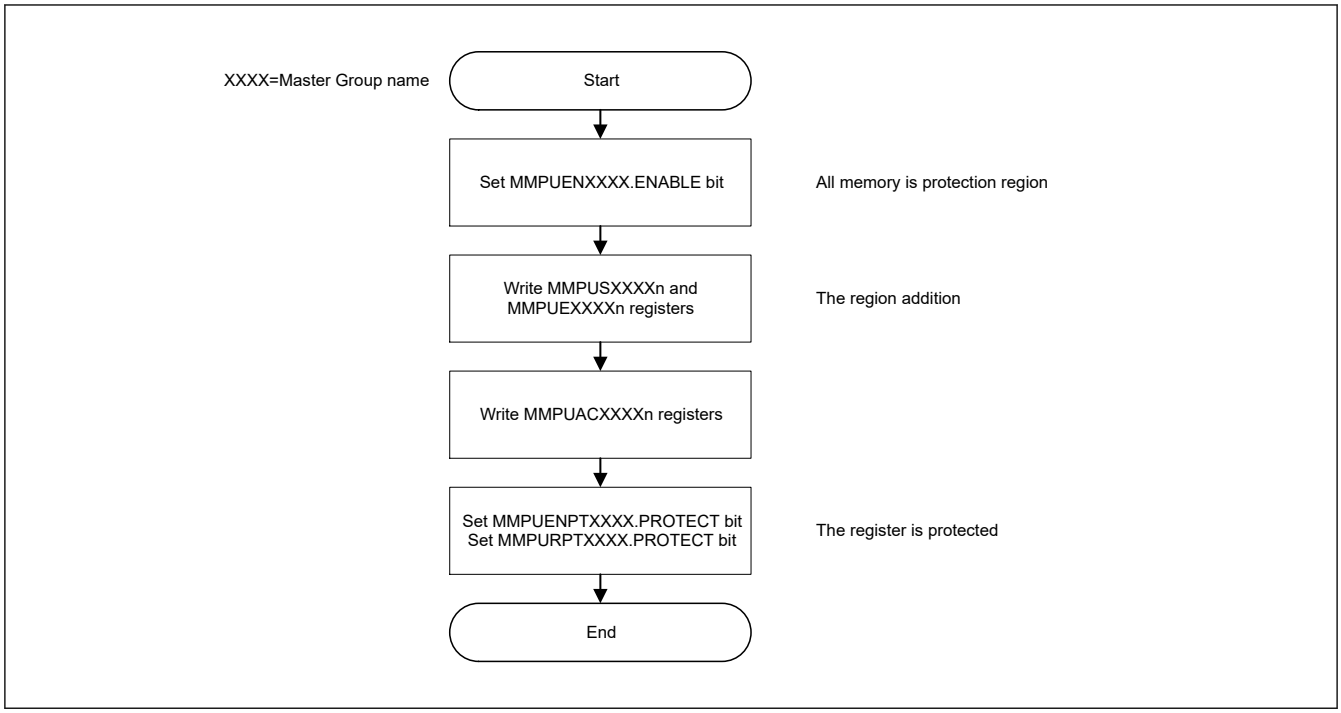


Figure 15.3 Register setting flow after reset

Figure 15.4 shows the register setting flow for adding regions. During this register setting, stop the master except the CPU.

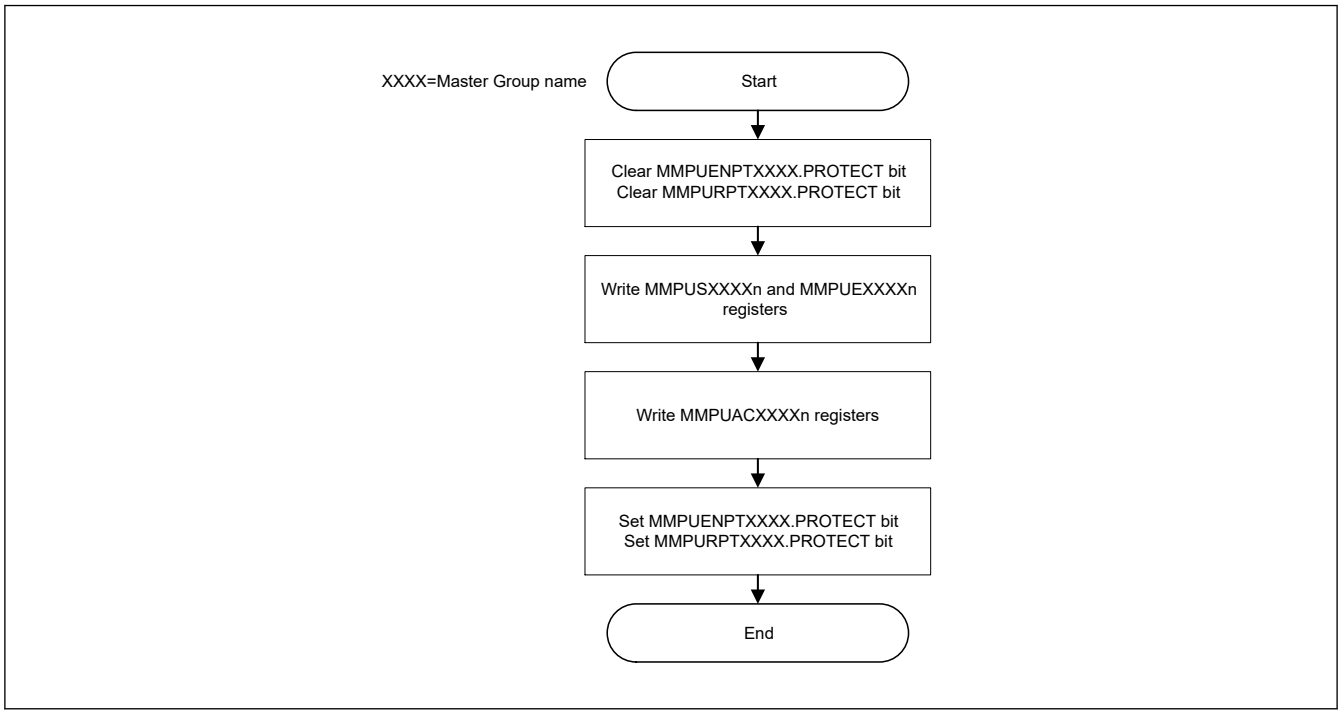


Figure 15.4 Register setting flow for region addition

15.3.2.2 Protecting the registers

Registers related to the bus master MPU can be protected with the PROTECT bit as shown in Table 15.4.

Table 15.4 PROTECT bit and protect target registers (1 of 2)

PROTECT bit	Protect target registers
MMPUENPTDMAC.PROTECT	MMPUENDMAC

Table 15.4 PROTECT bit and protect target registers (2 of 2)

PROTECT bit	Protect target registers
MMPUENPTEDMAC.PROTECT	MMPUENEDMAC
MMPUENPTGLCDC.PROTECT	MMPUENGLCDC
MMPUENPDRW.PROTECT	MMPUENDRW
MMPUENPTMIPI.PROTECT	MMPUENMIPI
MMPUENPTCEU.PROTECT	MMPUENCEU
MMPURPTDMAC.PROTECT	The following registers set to Non-secure by MMPUSARA.MMPUASAn (n = 0 to 7). MMPUSDMACn (n = 0 to 7) MMPUEDMACn (n = 0 to 7) MMPUACDMACn (n = 0 to 7)
MMPURPTDMAC_SEC.PROTECT	The following registers set to Secure by MMPUSARA.MMPUASAn (n = 0 to 7). MMPUSDMACn (n = 0 to 7) MMPUEDMACn (n = 0 to 7) MMPUACDMACn (n = 0 to 7)
MMPURPTEDMAC.PROTECT	MMPUSEDMACn (n = 0 to 3) MMPUEEDMACn (n = 0 to 3) MMPUACEDMACn (n = 0 to 3)
MMPURPTGLCDC.PROTECT	MMPUSGLCDCn (n = 0 to 1) MMPUEGLCDCn (n = 0 to 1) MMPUACGLCDCn (n = 0 to 1)
MMPURPDRW.PROTECT	MMPUSDRWn (n = 0 to 2) MMPUEDRWn (n = 0 to 2) MMPUACDRWn (n = 0 to 2)
MMPURPTMIPI.PROTECT	MMPUSMIPI MMPUEMIPI MMPUACMIPI
MMPURPTCEU.PROTECT	MMPUSCEUn (n = 0 to 1) MMPUECEUn (n = 0 to 1) MMPUACCEUn (n = 0 to 1)
MMPUOADPT.PROTECT	MMPUOAD.OAD

15.3.2.3 Memory Protection error

If access to a protected region is detected, the bus master MPU generates an error. Set the MMPUOAD bit to select whether the error is reported as a non-maskable interrupt or a reset.

See section 14, Buses for details of bus master MPU error.

The non-maskable interrupt status is indicated in ICU.NMISR.BUSST. For details, see section 13, Interrupt Controller Unit (ICU). The reset status is indicated in SYSC.RSTSR1.BUSRF. For details, see section 5, Resets.

15.4 References

1. *ARM® v8-M Architecture Reference Manual* (ARM DDI0553B.g)
2. *ARM® Cortex®-M85 Processor Technical Reference Manual*, ARM® Cortex®-M33 Processor Technical Reference Manual (ARM 101924_0002_05_en, ARM 100230_0004_00_en)
3. *ARM® Cortex®-M85 Processor User Guide Reference Material, ARM® Cortex®-M33 Processor User Guide Reference Material* (ARM 101927_0002_05_en, ARM 100234_0002_00_en)
4. *ARM® Cortex®-M85 Processor Integration and Implementation Manual, ARM® Cortex®-M33 Processor Integration and Implementation Manual* (ARM 101925_0002_05_en, ARM 100323_0002_00_en)

16. DMA Controller (DMAC)

16.1 Overview

The 8-channel direct memory access controller (DMAC) that can transfer data without intervention from the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address.

Table 16.1 lists the DMAC specifications, and Figure 16.1 shows a block diagram of the DMAC.

Table 16.1 DMAC specifications (1 of 2)

Item		Description
Number of channels		8 channels (DMACn (n = 0 to 7))
Transfer space		4 GB (0x00000000 to 0xFFFFFFFF excluding reserved areas)
Maximum transfer volume		64 M data (Maximum number of transfers in block transfer mode: 1,024 data × 65,536 blocks)
DMAC activation source		Selectable for each channel: ● Software trigger ● Interrupt requests from peripheral modules or trigger from external interrupt input pins.*1
Channel priority		Channel 0 > Channel 1 > Channel 2 > Channel 3… > Channel 7 (Channel 0: Highest)
Transfer data	Single data	Bit length: 8, 16, 32 bits
	Block size	Number of data: 1 to 1,024
Transfer mode	Normal transfer mode	● One data transfer by one DMA transfer request ● Free running function (setting in which total number of data transfers is not specified) settable
	Repeat transfer mode	● One data transfer by one DMA transfer request ● Program returns to the transfer start address on completion of the repeat size of data transfer specified for the transfer source or destination. ● Maximum settable repeat size: 1,024 ● Selectable free running function
	Repeat-block transfer mode	● One block data transfer by one DMA transfer request ● Maximum settable block size: 1,024 ● Block transfer can be repeated ● Maximum settable repeat size: 64K ● Selectable free running function
	Block transfer mode	● One block data transfer by one DMA transfer request ● Maximum settable block size: 1,024 data ● Selectable free running function
Selective functions	Extended repeat area function	● Function in which data can be transferred by repeating the address values in the specified range with the upper bit values in the transfer address register fixed ● Area of 2 bytes to 128 Mbytes separately settable as extended repeat area for transfer source and destination
Processing on DMAC transfer error		● When the DMAC transfer error occurs, it is stop the transfer that caused the error channel ● Request to clear the register for activation request of DMAC error channel to ICU
Interrupt (DMAC0n_INT)	Transfer end interrupt	Generated on completion of transferring data volume specified by the transfer counter.
	Transfer escape end interrupt	● Generated when the repeat size of data transfer is completed. ● Generated when the source address extended repeat area overflows. ● Generated when the destination address extended repeat area overflows.
Interrupt (DMA0_TRANSERR)	Error response detection interrupt	● Generated when the DMAC transfer error occurs
Event link activation (DMAC0n_INT)		An event link request is generated after each data transfer (for block transfer, after each block is transferred).
Module-stop function		Module-stop state can be set to reduce power consumption.

Table 16.1 DMAC specifications (2 of 2)

Item	Description
TrustZone Filter	Security and Privilege can be set for each channels

Note: Security attribution Register of DMAC channel is described in ICU.ICUSARC
Note 1. For details on DMAC activation sources, see Table 13.4 in section 13, Interrupt Controller Unit (ICU).

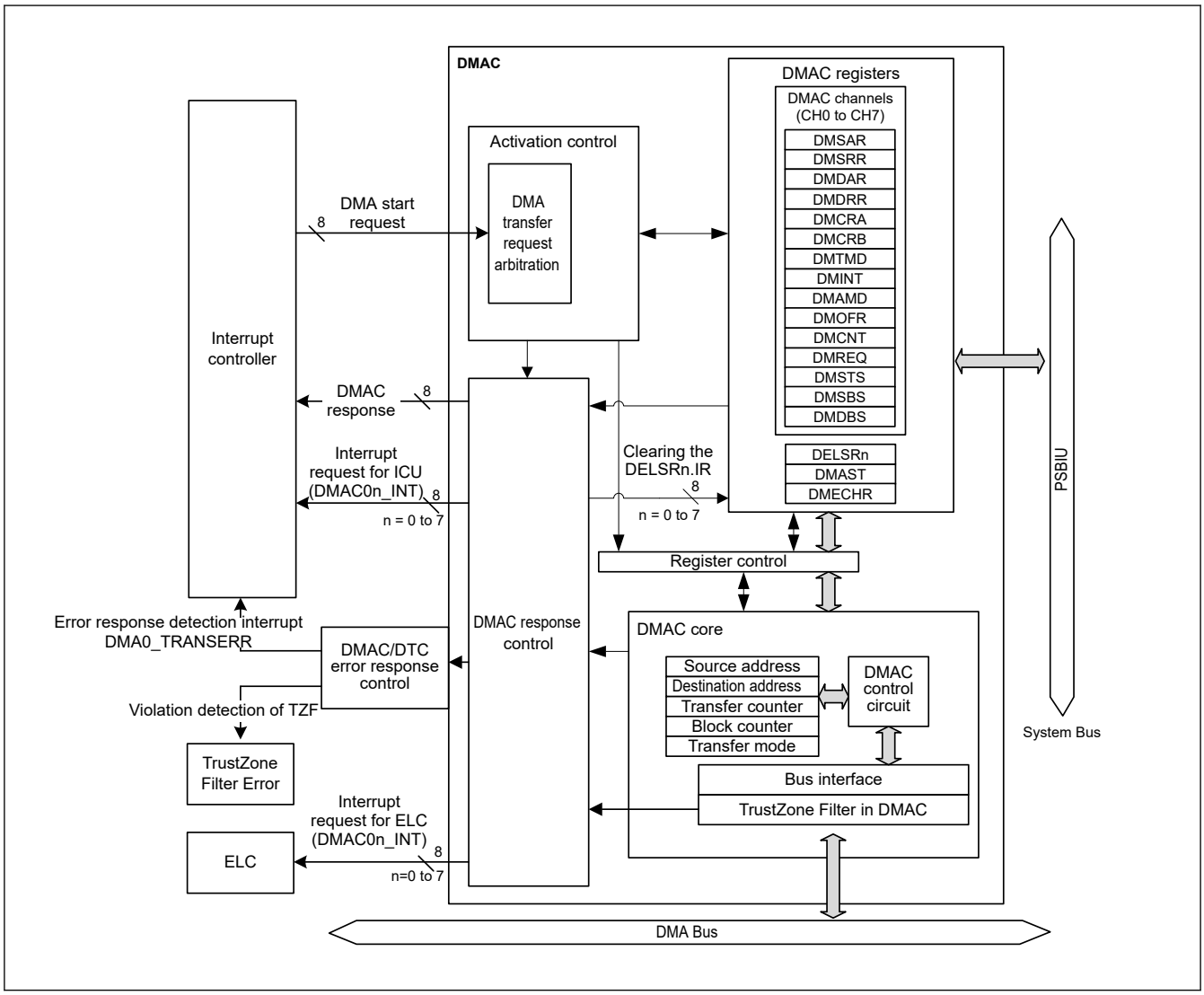


Figure 16.1 Block Diagram of DMAC

16.2 Register Descriptions

16.2.1 DMACCHSAR : DMA channel Security Attribution Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x1A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SADMAC7	SADMAC6	SADMAC5	SADMAC4	SADMAC3	SADMAC2	SADMAC1	SADMAC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SADMAC7 to SADMAC0	Security attributes of output and registers for DMAC channel 0: Secure 1: Non-secure	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

SADMACn (n = 0 to 7)

Security attributes registers for DMAC channel.

This bit determines the security attribute of the output as the master for each DMAC channel.

The controlled channel of DMAC registers are shown below.

- CPSCU.DMACCHPAR.PADMACn
- DMAC.DELSR channel n
- DMAC.DMSAR channel n
- DMAC.DMSRR channel n
- DMAC.DMDAR channel n
- DMAC.DMDRR channel n
- DMAC.DMCRA channel n
- DMAC.DMCRB channel n
- DMAC.DMTMD channel n
- DMAC.DMINT channel n
- DMAC.DMAMD channel n
- DMAC.DMOFR channel n
- DMAC.DMCNT channel n
- DMAC.DMREQ channel n
- DMAC.DMSTS channel n
- DMAC.DMSBS channel n
- DMAC.DMDBS channel n
- DMAC.DMBWR channel n

16.2.2 DMACCHPAR : DMA channel Privilege Attribution Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x1F0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	PADMAC7	PADMAC6	PADMAC5	PADMAC4	PADMAC3	PADMAC2	PADMAC1	PADMAC0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	PADMAC7 to PADMAC0	Privilege attributes of outputs and registers for DMAC channel 0: Privileged. 1: Unprivileged.	R/W
31:8	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-2, P-TYPE-1

PADMACn (n = 0 to 7)

Privilege attributes of output and registers for DMAC channel.

This bit determines the Privilege attribute of the output as the master for each DMAC channel. It also determines the privileged attributes of DMAC registers for each channel. See section 16.3.14. Channel Privilege.

The controlled the channel of DMAC registers are shown below.

- DMAC.DMSAR channel n
- DMAC.DMSRR channel n
- DMAC.DMDAR channel n
- DMAC.DMDRR channel n
- DMAC.DMCRA channel n
- DMAC.DMCRB channel n
- DMAC.DMTMD channel n
- DMAC.DMINT channel n
- DMAC.DMAMD channel n
- DMAC.DMOFR channel n
- DMAC.DMCNT channel n
- DMAC.DMREQ channel n
- DMAC.DMSTS channel n
- DMAC.DMSBS channel n
- DMAC.DMDBS channel n
- DMAC.DMBWR channel n

16.2.3 DMACSAR : DMAC Controller Security Attribution Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x34

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMASTSA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DMASTSA	DMAST Security Attribution 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1
Note: Only Secure access can write to this register. Both Secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by PRCR register.

For DMAC, security attribution is set for each channel. However, this register only sets the DMAST register security attribute.

DMASTSA bit (DMAST Security Attribution)

Security attributes of registers for DMAST. Do not write to DMASTSA bit while DMA transfer is enabled or a bus master is writing to the DMA registers.

16.2.4 DELSRn : DMAC Event Link Setting Register n (n = 0 to 7)

Base address: DMA0 = 0x4000_A800
DMA0_NS = 0x5000_A800

Offset address: 0x080+ 0x04 × n (n = 0 to 7)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	DELS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	DELS[8:0]	DMAC Event Link Select 0x00: Disable interrupts to the associated DMAC module Others: Event signal number to be linked. For details, see Table 13.4.	R/W*1
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	IR	DMAC Activation Request Status flag 0: No DMAC activation request occurred. 1: DMAC activation request occurred.	R/W*2
31:17	—	These bits are read as 0. The write value should be 0.	R/W*1

Note: P-TYPE-2 See below for access by security attribution.
Note 1. This register must be accessed by half-word or word.
Note 2. Writing 1 to the IR flag is prohibited.

This register has different secure access permission depending on the setting of Trusted Event Route Control Register(TEVTRCR).

If the security attribution is configured as Secure,

- Secure access is allowed.
- Non-secure write access is ignored and Non-secure read access is read as 0, TrustZone access error is generated.

If the security attribution is configured as Non-secure and the trusted event route is disabled,

- Secure write access is ignored and Secure read access is read as 0, TrustZone access error is generated.
- Non-secure access is allowed.

If the security attribution is configured as Non-secure and the trusted event route is enabled,

- Secure access to DELS bit is allowed.
- Non-secure write access to DELS bit is ignored and Non-secure read access to DELS bit is allowed.
- Secure write access to other bits is ignored and Secure read access to other bits is read as 0.
- Non-secure access to other bits is allowed.
- TrustZone access error is not generated.

DELS[8:0] bits (DMAC Event Link Select)

The DELS[8:0] bits link an event signal to the associated DMAC module. All DELS[8:0] bits must be written to simultaneously. Do not set the same event number in multiple DELSRn registers.

[Setting condition]

- When value is written to these bits.

[Clearing conditions]

- When 0 is written to these bits.
- When DMA transfer is stopped by the access error occurs. See section 16.5. Processing on DMA Transfer Error.

IR flag (DMAC Activation Request Status flag)

This flag is a status flag for a DMAC activation request.

This flag is associated with the DELS[8:0] bits of this register.

[Setting condition]

- When a DMAC activation request occurs from the associated peripheral module or IRQi pin.

[Clearing conditions]

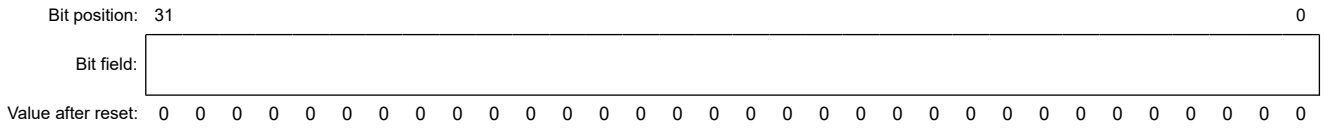
- When 0 is written to the flag
- When the DMA transfer is started after a DMAC activation request occurs.
- When DMA transfer is stopped by the access error occurs. See section 16.5. Processing on DMA Transfer Error.

Note: IR Flag is automatically cleared after completion of DMA transfer, so do not write "0" except in case of emergency, such as when an abort occurs. DMA transfer operation when "0" is written during DMA transfer can not be guaranteed.

16.2.5 DMSAR : DMA Source Address Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x00



Bit	Symbol	Function	R/W
31:0	n/a	Specifies the transfer source start address Setting range is 0x00000000 to 0xFFFFFFFF (4 GB).	R/W

Note: S-TYPE-3, P-TYPE-3

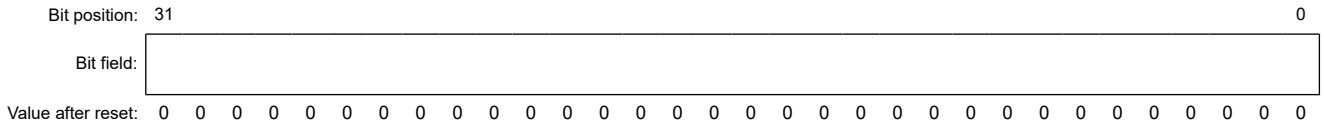
Set DMSAR while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

Note: Address alignment in this register must match the Transfer Data Size value selected in the DMTMD.SZ bit.

16.2.6 DMSRR : DMA Source Reload Address Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x20



Bit	Symbol	Function	R/W
31:0	n/a	Specifies the transfer source reload address 0x0000 0000 to 0xFFFF FFFF (4 GB)	R/W

Note: S-TYPE-3, P-TYPE-3

Set DMSRR while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer of the corresponding channel is disabled (DMCNT.DTE = 0).

DMSRR is used to store the start address of the buffer size set in DMSBS during repeat-block transfer mode. In repeat-block transfer mode, DMSAR reloads value of DMSRR after specified transfer finished.

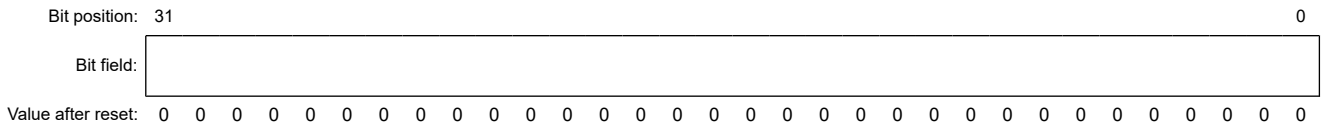
In normal transfer mode, repeat transfer mode and block transfer mode, DMSRR is not used. The setting is invalid.

Note: Address alignment in this register must match the Transfer Data Size value selected in the DMTMD.SZ bit.

16.2.7 DMDAR : DMA Destination Address Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x04



Bit	Symbol	Function	R/W
31:0	n/a	Specifies the transfer destination start address Setting range is 0x0000_0000 to 0xFFFF_FFFF (4 GB).	R/W

Note: S-TYPE-3, P-TYPE-3

Set DMDAR while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer of the corresponding channel is disabled (DMCNT.DTE = 0).

Note: Address alignment in this register must match the Transfer Data Size value selected in the DMTMD.SZ bits.

16.2.8 DMDRR : DMA Destination Reload Address Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x24

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Specifies the transfer destination reload address Setting range is 0x0000_0000 to 0xFFFF_FFFF (4 GB).	R/W

Note: S-TYPE-3, P-TYPE-3

Set DMDRR while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer of the corresponding channel is disabled (DMCNT.DTE = 0).

DMDRR is used to store the start address of the buffer size set in DMDBS during repeat-block transfer mode. In repeat-block transfer mode, DMDAR reloads the value of DMDRR after the specified transfer is finished.

In normal transfer mode, repeat transfer mode and block transfer mode, DMDRR is not used. The setting is invalid.

Note: Address alignment in this register must match the Transfer Data Size value selected in the DMTMD.SZ bits.

16.2.9 DMCRA : DMA Transfer Count Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	DMCRAH[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DMCRAL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DMCRAL[15:0]	Lower bits of transfer count Specifies the number of transfer operations.	R/W
25:16	DMCRAH[9:0]	Upper bits of transfer count Specifies the number of transfer operations.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Set the same value for DMCRAH and DMCRAL in repeat transfer mode, block transfer mode, and repeat-block transfer mode. Bits 15 to 10 are fixed to 0 in repeat transfer mode, block transfer mode, and repeat-block transfer mode.

(1) Normal Transfer Mode (DMTMD.MD[1:0] = 00b)

DMCRAL functions as a 16-bit transfer counter.

The number of transfer operations is one when the setting is 0x0001, and 65,535 when it is 0xFFFF. The value is decremented by one each time data is transferred.

When the setting is 0x0000, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running function).

Free running function is not selected by DMTMD.TKP bit in normal transfer mode.

DMCRAH is not used in normal transfer mode. Write 0x0000 to DMCRAH.

(2) Repeat Transfer Mode (DMTMD.MD[1:0] = 01b)

DMCRAH specifies the repeat size and DMCRAL functions as a 10-bit transfer counter.

The number of transfer operations is one when the setting is 0x001, 1023 when it is 0x3FF, and 1024 when it is 0x000. In repeat transfer mode, a value in the range of 0x000 to 0x3FF (1 to 1024) can be set for DMCRAH and DMCRAL.

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.

The value in DMCRAL is decremented by one each time data is transferred until it reaches 0x000, at which the value in DMCRAH is loaded into DMCRAL.

(3) Block Transfer Mode (DMTMD.MD[1:0] = 10b)

DMCRAH specifies the block size and DMCRAL functions as a 10-bit block size counter.

The block size is one when the setting is 0x001, 1023 when it is 0x3FF, and 1024 when it is 0x000. In block transfer mode, a value in the range of 0x000 to 0x3FF can be set for DMCRAH and DMCRAL.

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.

The value in DMCRAL is decremented by one each time data is transferred until it reaches 0x000, at which the value in DMCRAH is loaded into DMCRAL.

(4) Repeat-Block Transfer Mode (DMTMD.MD[1:0] = 11b)

DMCRAH specifies the block size and DMCRAL functions as a 10-bit block size counter.

The block size is one when the setting is 0x001, 1023 when it is 0x3FF, and 1024 when it is 0x000. In repeat-block transfer mode, a value in the range of 0x000 to 0x3FF can be set for DMCRAH and DMCRAL.

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.

The value in DMCRAL is decremented by one each time data is transferred until it reaches 0x000, at which the value in DMCRAH is loaded into DMCRAL.

16.2.10 DMCRB : DMA Block Transfer Count Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DMCRBH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DMCRBL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DMCRBL[15:0]	Functions as a number of block, repeat or repeat-block transfer counter. 0x0001 to 0xFFFF (1 to 65535) 0x0000 (65536)	R/W
31:16	DMCRBH[15:0]	Specifies the number of block, repeat or repeat-block transfer operations. 0x0001 to 0xFFFF (1 to 65535) 0x0000 (65536)	R/W

Note: S-TYPE-3, P-TYPE-3

Set the same value for DMCRBH and DMCRBL in repeat transfer mode, block transfer mode and repeat-block transfer mode.

DMCRBH specifies the number of block, repeat and repeat-block transfer operations, and DMCRBL functions as a 16-bit the number of block counter in block, repeat, and repeat-block transfer mode, respectively.

The number of transfer operations is one when the setting is 0x0001, 65535 when it is 0xFFFF, and 65536 when it is 0x0000.

In repeat transfer mode, the value is decremented by one when the final data of one repeat size is transferred.

In block transfer mode and repeat-block transfer mode, the value is decremented by one when the final data of one block size is transferred.

In normal transfer mode, DMCRB is not used. The setting is invalid.

When DMTMD.TKP is 1 and the final data of one repeat size or one block size is transferred, DMCRBL reloads the value of DMCRBH automatically.

16.2.11 DMTMD : DMA Transfer Mode Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x10

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MD[1:0]		DTS[1:0]		—	TKP	SZ[1:0]		—	—	—	—	—	—	DCTG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	DCTG[1:0]	Transfer Request Source Select 0 0: Software request 0 1: Hardware request*1 1 0: Setting prohibited 1 1: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	SZ[1:0]	Transfer Data Size Select 0 0: 8 bits 0 1: 16 bits 1 0: 32 bits 1 1: Setting prohibited	R/W
10	TKP	Transfer Keeping 0: Transfer is stopped by completion of specified total number of transfer operations. 1: Transfer is not stopped by completion of specified total number of transfer operations (free-running).	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
13:12	DTS[1:0]	Repeat Area Select 0 0: The destination is specified as the repeat area or block area. 0 1: The source is specified as the repeat area or block area. 1 0: The repeat area or block area is not specified. 1 1: Setting prohibited.	R/W

Bit	Symbol	Function	R/W
15:14	MD[1:0]	Transfer Mode Select 0 0: Normal transfer 0 1: Repeat transfer 1 0: Block transfer 1 1: Repeat-block transfer	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. To select the DMAC activation source, use the DELSRn registers. For details on DMAC activation sources, see Table 13.4 in section 13, Interrupt Controller Unit (ICU).

DTS[1:0] bits (Repeat Area Select)

DTS[1:0] select either the source or destination as the repeat area in repeat or block transfer mode. In normal or repeat-block transfer mode, setting these bits is invalid.

TKP bit (Transfer Keeping)

TKP selects either stopping transfer or keeping transfer by completion of specified total number of transfer operations in repeat, block or repeat-block transfer mode. In normal transfer mode, setting this bit is invalid.

16.2.12 DMINT : DMA Interrupt Setting Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x13

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DARIE	Destination Address Extended Repeat Area Overflow Interrupt Enable 0: Disables an interrupt request for an extended repeat area overflow on the destination address. 1: Enables an interrupt request for an extended repeat area overflow on the destination address.	R/W
1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable 0: Disables an interrupt request for an extended repeat area overflow on the source address. 1: Enables an interrupt request for an extended repeat area overflow on the source address.	R/W
2	RPTIE	Repeat Size End Interrupt Enable 0: Disables the repeat size end interrupt request. 1: Enables the repeat size end interrupt request.	R/W
3	ESIE	Transfer Escape End Interrupt Enable 0: Disables the transfer escape end interrupt request. 1: Enables the transfer escape end interrupt request.	R/W
4	DTIE	Transfer End Interrupt Enable 0: Disables the transfer end interrupt request. 1: Enables the transfer end interrupt request.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

DARIE bit (Destination Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the destination address occurs while DARIE bit is set to 1, the DMCNT.DTE bit is cleared to 0. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that an interrupt by an extended repeat area overflow on the destination address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the destination address, this bit is ignored.

When set to repeat-block transfer mode, do not use this bit.

SARIE bit (Source Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the source address occurs while SARIE bit is set to 1, the DMCNT.DTE bit is cleared to 0. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that an interrupt by an extended repeat area overflow on the source address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the source address, this bit is ignored.

When set to repeat-block transfer mode, do not use this bit.

RPTIE bit (Repeat Size End Interrupt Enable)

When RPTIE bit is set to 1 in repeat transfer mode, the DMCNT.DTE bit is cleared to 0 after completion of a 1-repeat size data transfer. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DMTMD.DTS[1:0] bits are 10b (= repeat area or block area is not specified).

When this bit is set to 1 in block transfer mode, the DMCNT.DTE bit is cleared to 0 after completion of a 1-block data transfer in the same way as repeat transfer mode. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DMTMD.DTS[1:0] bits are 10b (= repeat area or block area is not specified).

When set to repeat-block transfer mode, do not use this bit.

ESIE bit (Transfer Escape End Interrupt Enable)

ESIE bit enables or disables the transfer escape end interrupt requests (repeat size end interrupt request and extended repeat area overflow interrupt request) that are generated during DMA transfer.

The transfer escape end interrupt is generated when the DMSTS.ESIF flag is set to 1 with this bit set to 1. The transfer escape end interrupt is cleared by clearing this bit or the DMSTS.ESIF flag to 0.

DTIE bit (Transfer End Interrupt Enable)

DTIE bit enables or disables the transfer end interrupt request to be generated on completion of a specified number of data transfers.

The transfer end interrupt is generated when the DMSTS.DTIF flag is set to 1 with this bit set to 1. The transfer end interrupt is cleared by clearing this bit or the DMSTS.DTIF flag to 0.

16.2.13 DMAMD : DMA Address Mode Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x14

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SM[1:0]		SADR	SARA[4:0]					DM[1:0]		DADR	DARA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	DARA[4:0]	Destination Address Extended Repeat Area Specifies the extended repeat area on the destination address. For details on the settings, see Table 16.2.	R/W

Bit	Symbol	Function	R/W
5	DADR	Destination Address Update Select After Reload 0: Only reloading. 1: Add index after reloading.	R/W
7:6	DM[1:0]	Destination Address Update Mode 0 0: Destination address is fixed. 0 1: Offset addition. 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
12:8	SARA[4:0]	Source Address Extended Repeat Area Specifies the extended repeat area on the source address. For details on the settings, see Table 16.2.	R/W
13	SADR	Source Address Update Select After Reload 0: Only reloading. 1: Add index after reloading.	R/W
15:14	SM[1:0]	Source Address Update Mode 0 0: Source address is fixed. 0 1: Offset addition. 1 0: Source address is incremented. 1 1: Source address is decremented.	R/W

Note: S-TYPE-3, P-TYPE-3

DARA[4:0] bits (Destination Address Extended Repeat Area)

DARA[4:0] bits specify the extended repeat area on the destination address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes.

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer or block transfer is selected, and when DMTMD.DTS[1:0] = 00b (the transfer destination is specified as the repeat area or block area), write 00000b in the DARA[4:0] bits.

In repeat-block transfer mode, write 00000b in the DARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DMINT.DARIE bit set to 1. Table 16.2 lists the settings and the corresponding extended repeat areas.

DADR bits (Destination Address Update Select After Reload)

In repeat-block transfer mode, this bit specifies the behavior of DMDAR after reloading DMDRR.

When this bit is set to 1, an index value ((DMDBSH-DMDBSL) × DataSize) is added to DMDAR after reloading DMDRR.

When this bit is set to 0, DMDAR only reloads DMDRR. This behavior is described in Table 16.13.

In normal, repeat or block transfer mode, this bit is ignored.

DM[1:0] bits (Destination Address Update Mode)

DM[1:0] bits select the mode of updating the destination address.

When increment is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, or 10b, the destination address is incremented by 1, 2, or 4, respectively.

When decrement is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, or 10b, the destination address is decremented by 1, 2, or 4, respectively.

When offset addition is selected, the offset specified by the DMOFR register is added to the address.

SARA[4:0] bits (Source Address Extended Repeat Area)

SARA[4:0] bits specify the extended repeat area on the source address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes.

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer source, do not specify the extended repeat area on the source address. When repeat transfer or block transfer is selected, and when DMTMD.DTS[1:0] = 01b (the transfer source is specified as the repeat area or block area), write 00000b in the SARA[4:0] bits.

In repeat-block transfer mode, write 00000b in the SARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DMINT.SARIE bit set to 1. Table 16.2 lists the settings and the corresponding extended repeat areas.

SADR bits (Source Address Update Select After Reload)

In repeat-block transfer mode, this bit specifies the behavior of DMSAR after reloading DMSRR.

When this bit is set to 1, an index value ((DMSBSH-DMSBSL) × DataSize) is added to DMSAR after reloading DMSRR.

When this bit is set to 0, DMSAR only reloads DMSRR. This behavior is described in Table 16.12.

In normal, repeat or block transfer mode, this bit is ignored.

SM[1:0] bits (Source Address Update Mode)

SM[1:0] bits select the mode of updating the source address.

When increment is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, or 10b, the source address is incremented by 1, 2, or 4, respectively.

When decrement is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, or 10b, the source address is decremented by 1, 2, or 4, respectively.

When offset addition is selected, the offset specified by the DMOFR register is added to the address.

Table 16.2 SARA[4:0] or DARA[4:0] settings and corresponding repeat areas (1 of 2)

SARA[4:0] or DARA[4:0] settings	Extended repeat area
00000b	Not specified
00001b	2 bytes specified as extended repeat area by the lower 1 bit of the address
00010b	4 bytes specified as extended repeat area by the lower 2 bits of the address
00011b	8 bytes specified as extended repeat area by the lower 3 bits of the address
00100b	16 bytes specified as extended repeat area by the lower 4 bits of the address
00101b	32 bytes specified as extended repeat area by the lower 5 bits of the address
00110b	64 bytes specified as extended repeat area by the lower 6 bits of the address
00111b	128 bytes specified as extended repeat area by the lower 7 bits of the address
01000b	256 bytes specified as extended repeat area by the lower 8 bits of the address
01001b	512 bytes specified as extended repeat area by the lower 9 bits of the address
01010b	1 Kbyte specified as extended repeat area by the lower 10 bits of the address
01011b	2 Kbytes specified as extended repeat area by the lower 11 bits of the address
01100b	4 Kbytes specified as extended repeat area by the lower 12 bits of the address
01101b	8 Kbytes specified as extended repeat area by the lower 13 bits of the address
01110b	16 Kbytes specified as extended repeat area by the lower 14 bits of the address
01111b	32 Kbytes specified as extended repeat area by the lower 15 bits of the address
10000b	64 Kbytes specified as extended repeat area by the lower 16 bits of the address
10001b	128 Kbytes specified as extended repeat area by the lower 17 bits of the address
10010b	256 Kbytes specified as extended repeat area by the lower 18 bits of the address
10011b	512 Kbytes specified as extended repeat area by the lower 19 bits of the address

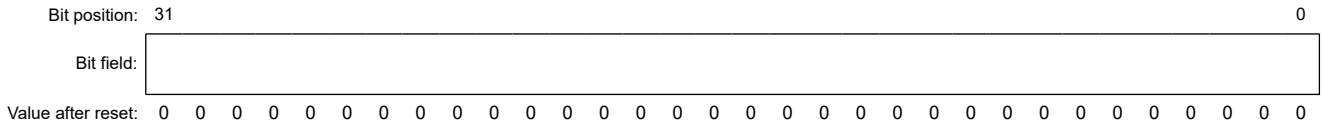
Table 16.2 SARA[4:0] or DARA[4:0] settings and corresponding repeat areas (2 of 2)

SARA[4:0] or DARA[4:0] settings	Extended repeat area
10100b	1 Mbyte specified as extended repeat area by the lower 20 bits of the address
10101b	2 Mbytes specified as extended repeat area by the lower 21 bits of the address
10110b	4 Mbytes specified as extended repeat area by the lower 22 bits of the address
10111b	8 Mbytes specified as extended repeat area by the lower 23 bits of the address
11000b	16 Mbytes specified as extended repeat area by the lower 24 bits of the address
11001b	32 Mbytes specified as extended repeat area by the lower 25 bits of the address
11010b	64 Mbytes specified as extended repeat area by the lower 26 bits of the address
11011b	128 Mbytes specified as extended repeat area by the lower 27 bits of the address
11100b to 11111b	Setting prohibited.

16.2.14 DMOFR : DMA Offset Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x18



Bit	Symbol	Function	R/W
31:0	n/a	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination. 0x00000000 to 0x00FFFFFF (0 bytes to (16 M – 1) bytes) 0xFF000000 to 0xFFFFFFFF (–16 Mbytes to –1 byte)	R/W

Note: S-TYPE-3, P-TYPE-3

Write to this register while the DMAC operation is stopped or DMA transfer is disabled (not during data transfer).

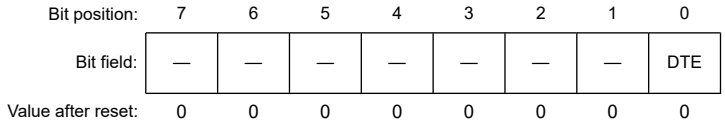
Setting bits 31 to 25 is invalid; a value of bit 24 is extended to bits 31 to 25. Reading DMOFR returns the extended value.

In repeat-block transfer mode, the offset is not specified by DMOFR when offset addition is selected, write 0 to DMOFR.

16.2.15 DMCNT : DMA Transfer Enable Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x1C



Bit	Symbol	Function	R/W
0	DTE	DMA Transfer Enable 0: Disables DMA transfer. 1: Enables DMA transfer.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

DTE bit (DMA Transfer Enable)

When the DMAST.DMST bit is set to 1 (DMAC activation is enabled) and this bit is set to 1 (DMA transfer is enabled), DMA transfer can be started for the corresponding channel.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When 0 is written to this bit.
- When the specified total volume of data transfer is completed.
- When DMA transfer is stopped by the repeat size end interrupt.
- When DMA transfer is stopped by the extended repeat area overflow interrupt.
- When DMA transfer is stopped by the access error occurs. See section 16.5. Processing on DMA Transfer Error.

If the DTE of the corresponding channel is set to 0 during DMA transfer, no new transfer requests will be accepted after the transfer ends.

16.2.16 DMREQ : DMA Software Start Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x1D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	CLRS	—	—	—	SWREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SWREQ	DMA Software Start 0: DMA transfer is not requested. 1: DMA transfer is requested.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	CLRS	DMA Software Start Bit Auto Clear Select 0: SWREQ bit is cleared after DMA transfer is started by software. 1: SWREQ bit is not cleared after DMA transfer is started by software.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

SWREQ bit (DMA Software Start)

When 1 is written to SWREQ bit, a DMA transfer request is generated. After DMA transfer is started in response to the request, this bit is cleared to 0 if the CLRS bit is set to 0. This bit is not cleared to 0 while the CLRS bit is set to 1. In this case, a DMA transfer request can be issued again after completion of a transfer.

Note that, however, setting this bit is valid and DMA transfer by software is enabled only when the DMTMD.DCTG[1:0] bits are set to 00b (DMAC activation source is software).

Setting this bit is invalid when the DMTMD.DCTG[1:0] bits are set to a value other than 00b.

To start DMA transfer by software with the CLRS bit being 0, ensure that the SWREQ bit is 0, and then write 1 to the SWREQ bit.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When a DMA transfer request by software is accepted and DMA transfer is started while the CLRS bit is set to 0 (the SWREQ bit is cleared after DMA transfer is started by software).
- When 0 is written to this bit.

When DMA transfer is stopped by the access error occurs. See section 16.5. Processing on DMA Transfer Error

CLRS bit (DMA Software Start Bit Auto Clear Select)

CLRS bit specifies whether to clear the SWREQ bit to 0 after DMA transfer is started in response to the DMA transfer request generated by setting the SWREQ bit to 1. With this bit set to 0, the SWREQ bit is cleared to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not cleared to 0. In this case, a DMA transfer request can be issued again after completion of a transfer.

16.2.17 DMSTS : DMA Status Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x1E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ACT	—	—	DTIF	—	—	—	ESIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ESIF	Transfer Escape End Interrupt Flag 0: A transfer escape end interrupt has not been generated. 1: A transfer escape end interrupt has been generated.	R/W*1
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	DTIF	Transfer End Interrupt Flag 0: A transfer end interrupt has not been generated. 1: A transfer end interrupt has been generated.	R/W*1
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	ACT	DMAC Active Flag 0: DMAC is in the idle state. 1: DMAC is operating.	R

Note: S-TYPE-3, P-TYPE-3
Note 1. Only 0 can be written to clear the flag.

ESIF flag (Transfer Escape End Interrupt Flag)

This flag indicates that the transfer escape end interrupt has been generated.

[Setting conditions]

- When 1-repeat size data transfer is completed in repeat transfer mode with the DMINT.RPTIE bit set to 1.
- When 1-block data transfer is completed in block transfer mode with the DMINT.RPTIE bit set to 1.
- When an extended repeat area overflow on the source address occurs while the DMINT.SARIE bit is set to 1 and the DMAMD.SARA[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer source address).
- When an extended repeat area overflow on the destination address occurs while the DMINT.DARIE bit is set to 1 and the DMAMD.DARA[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer destination address).

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the DMCNT.DTE bit.

DTIF flag (Transfer End Interrupt Flag)

This flag indicates that the transfer end interrupt has been generated.

[Setting conditions]

- When the specified number of unit-transfers are completed in normal transfer mode (the value of DMCRAL becoming 0 on completion of transfer).
- When the specified number of repeat transfer operations are completed in repeat transfer mode (the value of DMCRBL becoming 0 on completion of transfer with DMTMD.TKP = 0 or the value of DMCRBL reloading DMCRBH with DMTMD.TKP = 1).
- When the specified number of blocks have been transferred in block transfer mode and repeat-block transfer mode (the value of DMCRBL becoming 0 on completion of transfer with DMTMD.TKP = 0 or the value of DMCRBL reloading DMCRBH with DMTMD.TKP = 1).

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the DMCNT.DTE bit.

ACT flag (DMAC Active Flag)

This flag indicates whether the DMAC is in the idle or active state.

[Setting condition]

- When the DMAC starts data transfer operation.

[Clearing condition]

- When data transfer in response to one transfer request is completed.

16.2.18 DMSBS : DMA Source Buffer Size Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DMSBSH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DMSBSL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DMSBSL[15:0]	Functions as data transfer counter in repeat-block transfer mode See Table 16.3 for available settings.	R/W
31:16	DMSBSH[15:0]	Specifies the repeat-area size in repeat-block transfer mode See Table 16.3 for available settings.	R/W

Note: S-TYPE-3, P-TYPE-3

Set the same value for DMSBSH and DMSBSL in repeat-block transfer mode. Write 0x00000000 to DMSBS in normal, repeat and block transfer mode.

DMSBSH specifies buffer size and DMSBSL functions as a 16-bit buffer size counter in repeat-block transfer mode. In repeat-block transfer mode, source repeat area is specified by DMSBSH.

When address update mode is incremented address or decremented address, this register means the numbers of data of whole buffer. When address update mode is offset addition, this register means the numbers of data of an individual buffer. In offset addition, setting DMSBSH and DMSBSL to 0x0000 is prohibited. When final data of one buffer size is transferred,

DMSBSL reloads value of DMSBSH. When address update mode is fixed address, this register is ignored. Table 16.3 shows the setting values of DMA Source Buffer Size Register corresponding to Transfer Data Size in Source Address Update Mode.

Table 16.3 Available setting for DMSBS register in repeat-block transfer mode

Source address update mode (DMAMD.SM)	Transfer data size (DMTMD.SZ)	Available setting for DMSBSH and DMSBSL bits
Source address is fixed (SM = 00b)	Don't care	0x0000 (DMSBS is not used)
Offset addition (SM = 01b)	8 bits (SZ = 00b)	0x0001 to 0xFFFF (1 to 65535)
	16 bits (SZ = 01b)	0x0001 to 0x7FFF (1 to 32767)
	32 bits (SZ = 10b)	0x0001 to 0x3FFF (1 to 16383)
Source address is incremented or decremented (SM = 1xb)	Don't care	0x0000 (infinite) 0x0001 to 0xFFFF (1 to 65535)

In normal, repeat and block transfer mode, DMSBS is not used. The setting is invalid.

16.2.19 DMDBS : DMA Destination Buffer Size Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DMDBSH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DMDBSL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DMDBSL[15:0]	Functions as data transfer counter in repeat-block transfer mode. See Table 16.4 for available settings.	R/W
31:16	DMDBSH[15:0]	Specifies the repeat-area size in repeat-block transfer mode. See Table 16.4 for available settings.	R/W

Note: S-TYPE-3, P-TYPE-3

Set the same value for DMDBSH and DMDBSL in repeat-block transfer mode. Write 0x00000000 to DMDBS in normal, repeat and block transfer mode.

DMDBSH specifies buffer size and DMDBSL functions as a 16-bit buffer size counter in repeat-block transfer mode. In repeat-block transfer mode, destination repeat area is specified by DMDBSH.

When address update mode is incremented address or decremented address, this register means the numbers of data of whole buffer. When address update mode is offset addition, this register means the numbers of data of an individual buffer. In offset addition, setting DMDBSH and DMDBSL to 0x0000 is prohibited. When final data of one buffer size is transferred, DMDBSL reloads value of DMDBSH. When address update mode is fixed address, this register is ignored. Table 16.4 shows the setting values of Destination Buffer Size Register corresponding to Transfer Data Size in Destination Address Update Mode.

Table 16.4 Available setting for DMDBS register in repeat-block transfer mode (1 of 2)

Destination address update mode (DMAMD.DM)	Transfer data size (DMTMD.SZ)	Available setting for DMDBSH and DMDBSL bits
Destination address is fixed (DM = 00b)	Don't care	0x0000 (DMDBS is not used)

Table 16.4 Available setting for DMDBS register in repeat-block transfer mode (2 of 2)

Destination address update mode (DMAMD.DM)	Transfer data size (DMTMD.SZ)	Available setting for DMDBSH and DMDBSL bits
Offset addition (DM = 01b)	8 bits (SZ = 00b)	0x0001 to 0xFFFF (1 to 65535)
	16 bits (SZ = 01b)	0x0001 to 0x7FFF (1 to 32767)
	32 bits (SZ = 10b)	0x0001 to 0x3FFF (1 to 16383)
Destination address is incremented or decremented (DM = 1xb)	Don't care	0x0000 (infinite) 0x0001 to 0xFFFF (1 to 65535)

In normal, repeat and block transfer mode, DMDBS is not used. The setting is invalid.

16.2.20 DMBWR : DMA Bufferable Write Enable Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x30

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	BWE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BWE	Bufferable Write Enable 0: Disables Bufferable Write 1: Enables Bufferable Write	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

BWE bit (Bufferable Write Enable)

BWE bit indicates bufferable write is either enabled or disabled.

If this bit is 1, even if the write access as DMAC is completed, the actual slave write may not have ended.

When this bit is 1, it requests an early response by writing write data to the temporary buffer if possible to the slave. Therefore, even if write access as DMAC is completed, the actual slave write may not be finished. If an error occurs during write access to the target slave, the error may not be detected and the transfer may not be automatically stopped. In such cases, the error response detection interrupt (DMA_TRANSERR) will not occur.

See section 14.7.2. Operations When a Bus Error Occurs for slave groups that support bufferable writes and slave groups for which error response detection interrupts cannot be generated.

Some individual slave modules also support bufferable writes, so Refer to the respective module chapters.

[Setting condition]

- When 1 is written to this bit

[Clearing conditions]

- When 0 is written to this bit

16.2.21 DMAST : DMA Module Activation Register

Base address: DMA0 = 0x4000_A800
DMA0_NS = 0x5000_A800

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	DMST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DMST	DMAC Operation Enable 0: DMAC activation is disabled. 1: DMAC activation is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE2

DMST bit (DMAC Operation Enable)

Setting the DMAST.DMST to 1 enables DMAC activation for all channels. When the DMST bit is set to 1 (DMAC activation is enabled), and 1 is written to the DMCNT.DTE bit (DMA transfer is enabled) for multiple channels, all associated channels can be placed in the transfer request ready state at the same time.

When the DMST bit clears to 0 during DMA transfer, DMA transfer is suspended after the current data transfer associated with a single transfer request completes. To resume DMA transfer, set the DMST bit to 1 again.

[Setting condition]

- When 1 is written to this bit.

[Clearing condition]

- When 0 is written to this bit.

16.2.22 DMECHR : DMAC Error Channel Register

Base address: DMA0 = 0x4000_A800
DMA0_NS = 0x5000_A800

Offset address: 0x40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMESTA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	DMECHSAM	—	—	—	—	—	DMECH		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	DMECH	DMAC Error channel Indicates the channel number causing the error 0 0 0: Error occurred on Channel 0 0 0 1: Error occurred on Channel 1 0 1 0: Error occurred on Channel 2 ⋮ ⋮ 1 1 1: Error occurred on Channel 7	R
7:3	—	These bits are read as 0. The write value should be 0.	R

Bit	Symbol	Function	R/W
8	DMECHSAM	DMAC Error channel Security Attribution Monitor Indicates the security attribution of a channel causing the error 0: Secure channel 1: Non-secure channel	R
15:9	—	These bits are read as 0. The write value should be 0.	R
16	DMESTA	DMAC Error Status 0: No DMA transfer error occurred 1: DMA transfer error occurred	R/W*1
31:17	—	These bits are read as 0. The write value should be 0.	R

Note: P-TYPE-2 See below for access by security attribution.
Note 1. Writing to DMESTA depends on the value of DMECHSAM

When reading this register, it can be accessed from both secure and non-secure access.

When writing this register, it depends on DMECHR.DMESA.

- When DMESA = 1, it can be accessed from secure and non-secure access.
- When DMESA = 0, it can be accessed from secure. An error is returned when write access is performed in a non-secure access.

This register is cleared by a reset caused by a transfer error. Please select interrupt (DMA0_TRANSERR) in BUS.OADCFG.OAD when you want to debug the program.

DMECH[2:0] bit (DMAC Error channel)

When a transfer error due to DMA transfer occurs, it stores the channel of DMAC that was violated.

When reset was selected in MMPUOAD.OAD and MSAOAD.OAD, Since this register is also reset. Please select NMI when you want to debug the program.

[Set condition]

- When the DMAC transfer error occurs and DMESTA = 0.

[Clearing condition]

- When 1 is written to DMESTA.

DMECHSAM bit (DMAC Error channel Security Attribution Monitor)

When a transfer error due to DMA transfer occurs, it indicates the security attribution of the violating DMAC channel.

When reset was selected in MMPUOAD.OAD and MSAOAD.OAD, Since this register is also reset. Please select NMI when you want to debug the program.

[Set condition]

- When the DMAC transfer error occurs and DMESTA = 0.

[Clearing condition]

- When 1 is written to DMESTA.

DMESTA bit (DMAC Error Status)

Indicates whether or not a DMA transfer error occurred.

DMECH, DMECHSAM, DMESTA are cleared by writing 1 to DMESTA. Writing 0 to DMESTA is ignored.

When reset was selected in MMPUOAD.OAD and MSAOAD.OAD, Since this register is also reset. Please select NMI when you want to debug the program.

[Set condition]

- When the DMAC transfer error occurs.

[Clearing condition]

- When 1 is written to DMESTA.

Note: When DMECHSAM = 1, it can be cleared in the Secure state and Non-secure state. DMECHSAM = 0, it can not be cleared in the Non-secure state.

16.3 Operation

16.3.1 Transfer Mode

16.3.1.1 Normal Transfer Mode

In normal transfer mode, one data is transferred by one transfer request. A maximum of 65535 can be set as the number of transfer operations using the DMCRAL register. When these bits are set to 0x0000, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free-running function). Setting DMCRB register is invalid in normal transfer mode. Except in free-running function, a transfer end interrupt request can be generated after completion of the specified number of transfer operations.

Table 16.5 summarizes the register update operation in normal transfer mode, and Figure 16.2 shows the operation in normal transfer mode.

Table 16.5 Register update operation in normal transfer mode

Register	Function	Update operation after completion of a transfer by one transfer request
DMSAR	Transfer source address	Increment/decrement/fixed/offset addition
DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition
DMCRAL	Transfer count	Decremented by one/not updated (in free running function)
DMCRAH	—	Not updated (Not used in normal transfer mode)
DMCRB	—	Not updated (Not used in normal transfer mode)

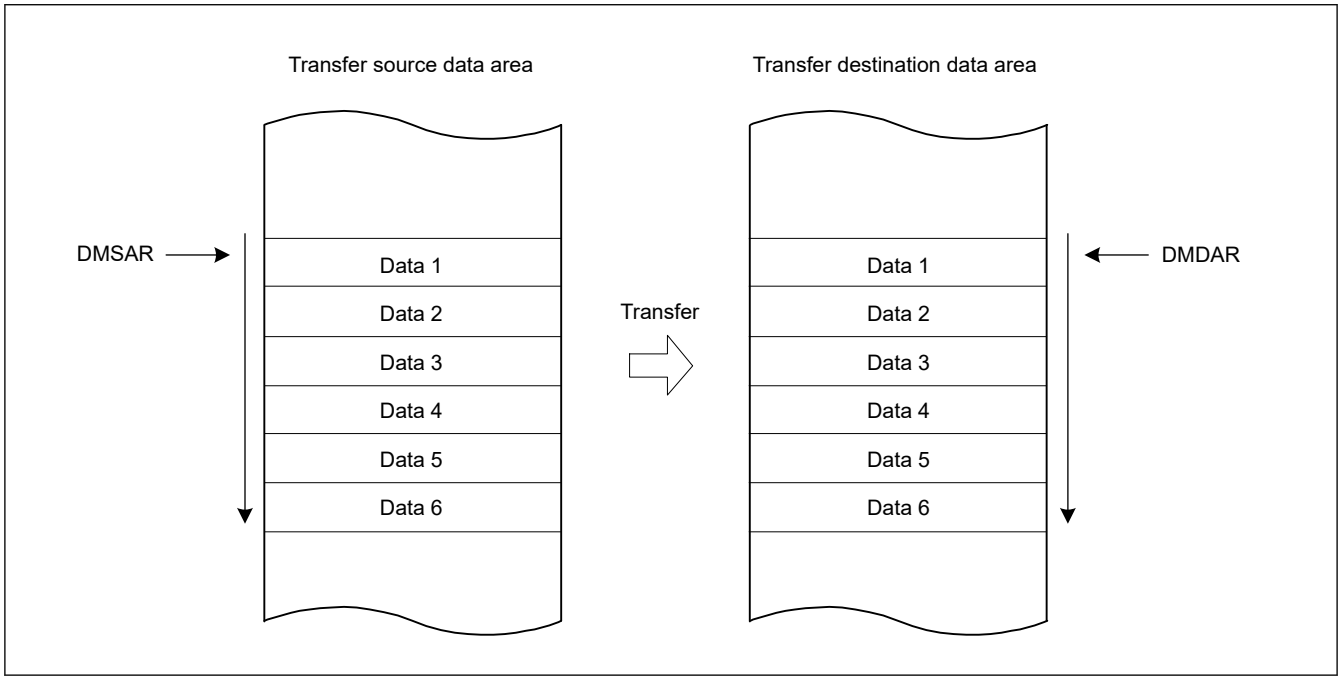


Figure 16.2 Operation in normal transfer mode

16.3.1.2 Repeat Transfer Mode

In repeat transfer mode, one data is transferred by one transfer request.

A maximum of 1K data can be set as a total repeat transfer size using DMCRA register.

A maximum of 64K can be set as the number of repeat transfer operations using DMCRB register; therefore, a maximum of 64M data (1K data × 64K counts of repeat transfer operations) can be set as a total data transfer size.

Either the transfer source or transfer destination can be specified as a repeat area. When transfer of the repeat size data is completed, the address of the specified repeat area (DMSAR or DMDAR) returns to the transfer start address. When data of the specified repeat size has all been transferred in repeat transfer mode, DMA transfer can be stopped, and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DMCNT.DTE bit in the repeat size end interrupt handling.

A transfer end interrupt request can be generated after completion of the specified number of repeat transfer operations.

Table 16.6 summarizes the register update operation in repeat transfer mode, and Figure 16.3 shows the operation in repeat transfer mode.

Table 16.6 Register update operation in repeat transfer mode

Register	Function	Update operation after completion of a transfer by one transfer request	
		When DMCRAL register is not 1	When DMCRAL register is 1 (Transfer of the last data in repeat size)
DMSAR	Transfer source address	Increment/decrement/fixed/offset addition	• DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition • DMTMD.DTS[1:0] = 01b Initial value of DMSAR • DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition	• DMTMD.DTS[1:0] = 00b Initial value of DMDAR • DMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition • DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
DMCRAH	Repeat size	Not updated	Not updated
DMCRAL	Transfer count	Decremented by one	DMCRAH
DMCRBH	Number of repeat transfer operations	Not updated	Not updated
DMCRBL	Count of repeat transfer operations	Not updated	Decremented by one

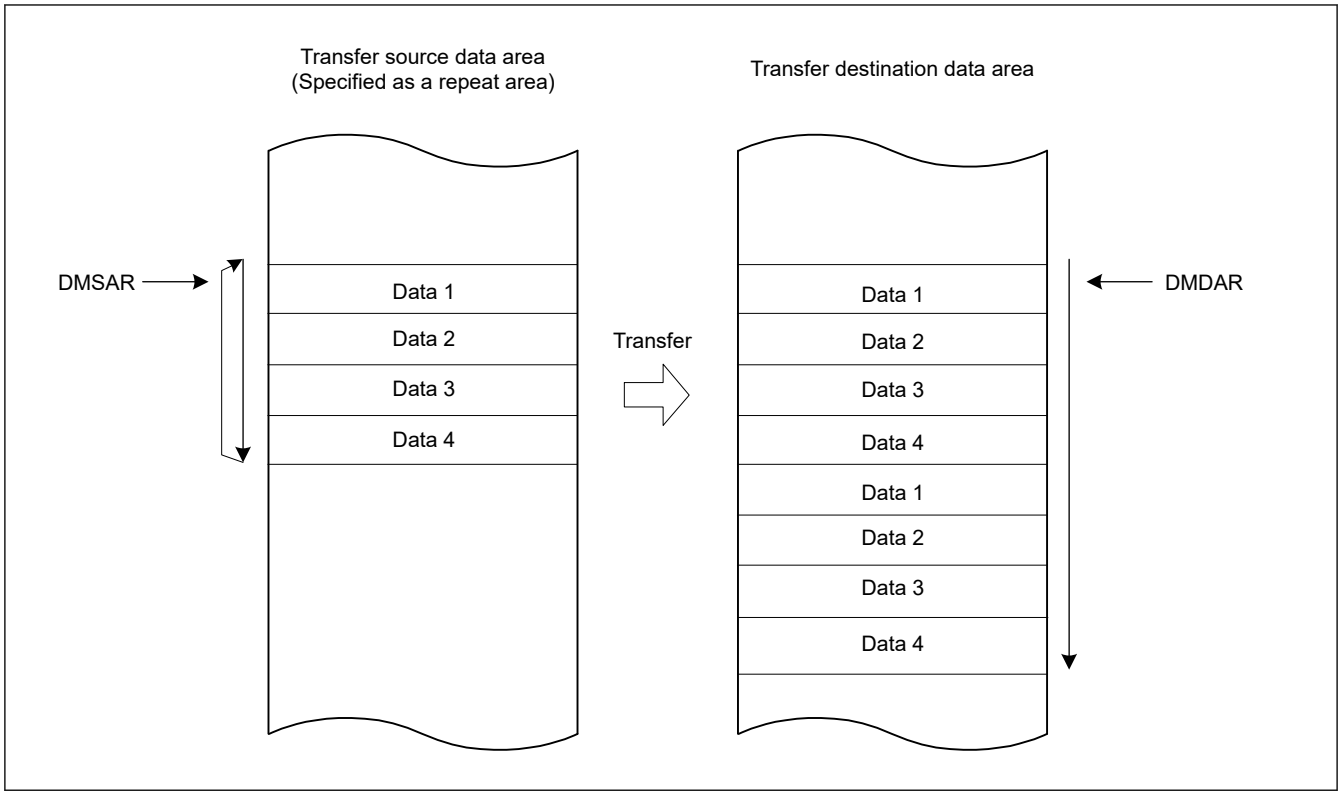


Figure 16.3 Operation in repeat transfer mode

16.3.1.3 Block Transfer Mode

In block transfer mode, a single block data is transferred by one transfer request.

A maximum of 1K data can be set as a total block transfer size using DMCRA register.

A maximum of 64K can be set as the number of block transfer operations using DMCRB register; therefore, a maximum of 64M data (1K data × 64K counts of block transfer operations) can be set as a total data transfer size.

Either the transfer source or transfer destination can be specified as a block area. When transfer of a single block data is completed, the address of the specified block area (DMSAR or DMDAR) returns to the transfer start address. When a single block data has all been transferred in block transfer mode, DMA transfer can be stopped, and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DMCNT.DTE bit in the repeat size end interrupt handling.

Transfer end interrupt request can be generated after completion of the specified number of block transfer operations.

Table 16.7 summarizes the register update operation in block transfer mode, and Figure 16.4 shows the operation in block transfer mode.

Table 16.7 Register update operation in block transfer mode (1 of 2)

Register	Function	Update operation after completion of single-block transfer by one transfer request
DMSAR	Transfer source address	• DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition • DMTMD.DTS[1:0] = 01b Initial value of DMSAR • DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
DMDAR	Transfer destination address	• DMTMD.DTS[1:0] = 00b Initial value of DMDAR • DMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition • DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition

Table 16.7 Register update operation in block transfer mode (2 of 2)

Register	Function	Update operation after completion of single-block transfer by one transfer request
DMCRAH	Block size	Not updated
DMCRAL	Transfer count	DMCRAH
DMCRBH	Number of block transfer operations	Not updated
DMCRBL	Count of block transfer operations	Decremented by one

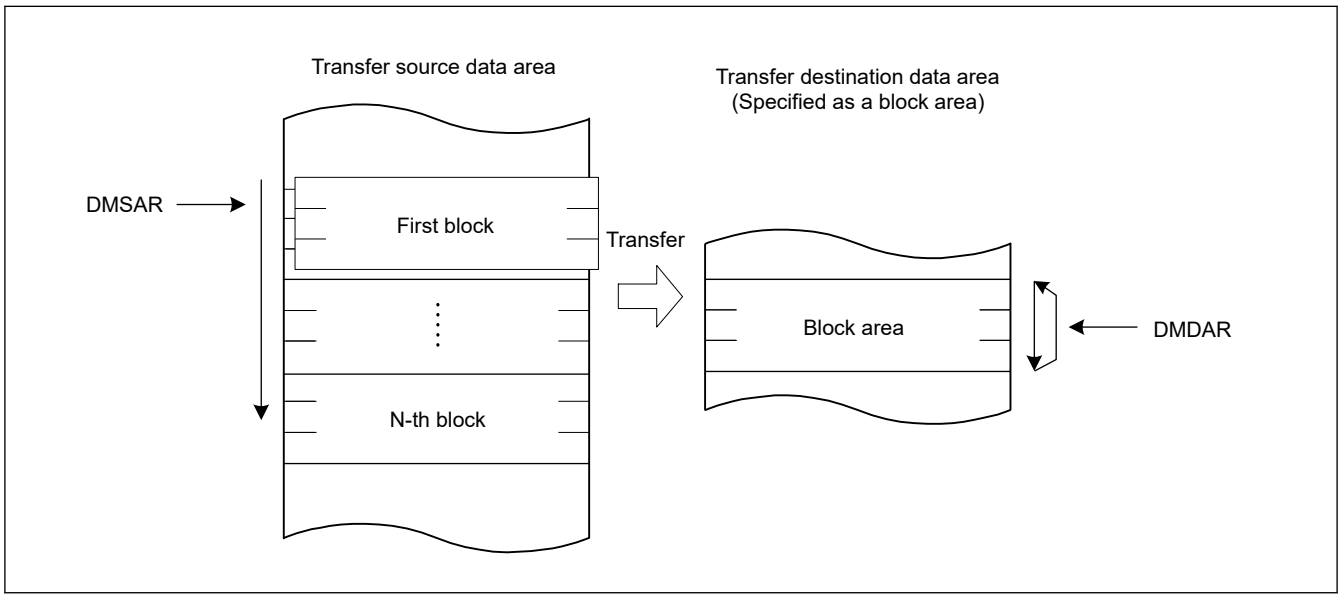


Figure 16.4 Operation in block transfer mode

16.3.1.4 Repeat-Block Transfer Mode

Repeat-block transfer is the operation mode with the following functions added to the block transfer function.

Repeat function: Added function (ring buffer) to repeat specified address area.

Offset function: Multiple areas with offset can be specified within one block transfer.

The repeat function and the offset function can be used for both the transfer source and the transfer destination of repeat-block transfer.

Figure 16.5 shows an example of adding a repeat function to the transfer destination.

Figure 16.6 shows repeat-block transfer with an offset to the transfer destination.

In repeat-block transfer mode, a single block data is transferred by one transfer request.

A maximum of 1K data can be set as a total block transfer size using DMCRA of the DMACn.

A maximum of 64K can be set as the number of block transfer operations using DMCRB of the DMACn; therefore, a maximum of 64M data (1K data × 64K counts of block transfer operations) can be set as a total data transfer size.

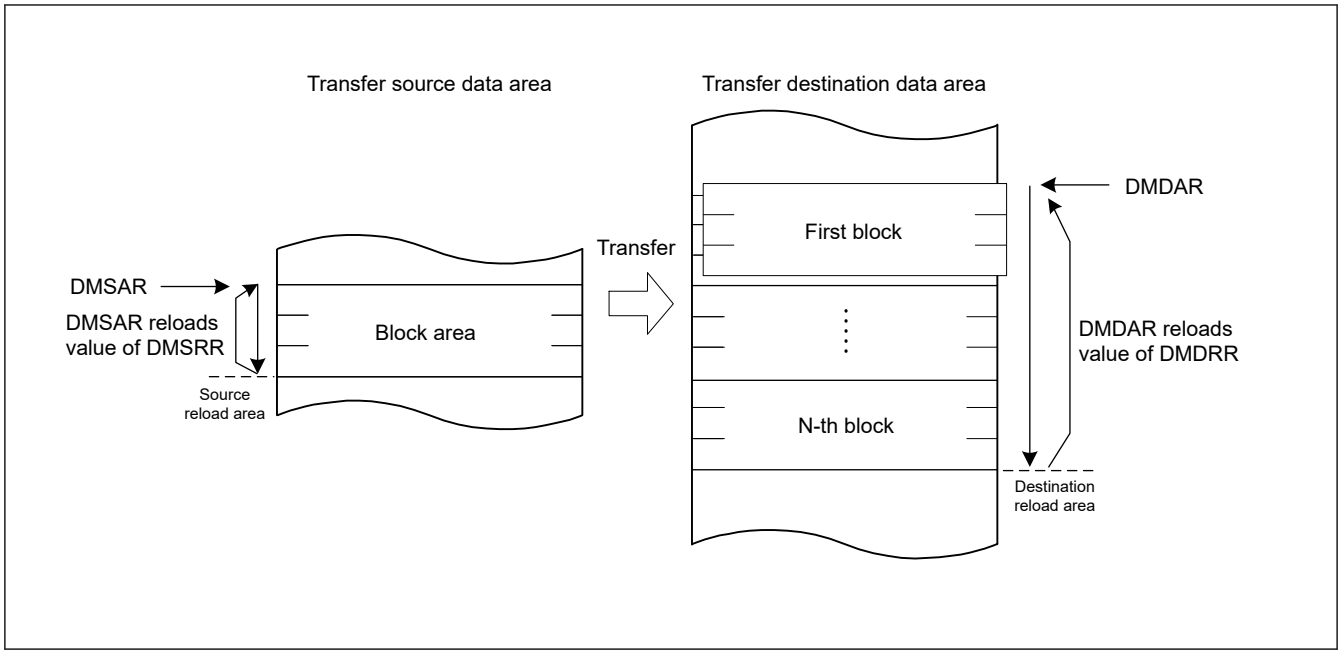


Figure 16.5 Operation in repeat block transfer mode

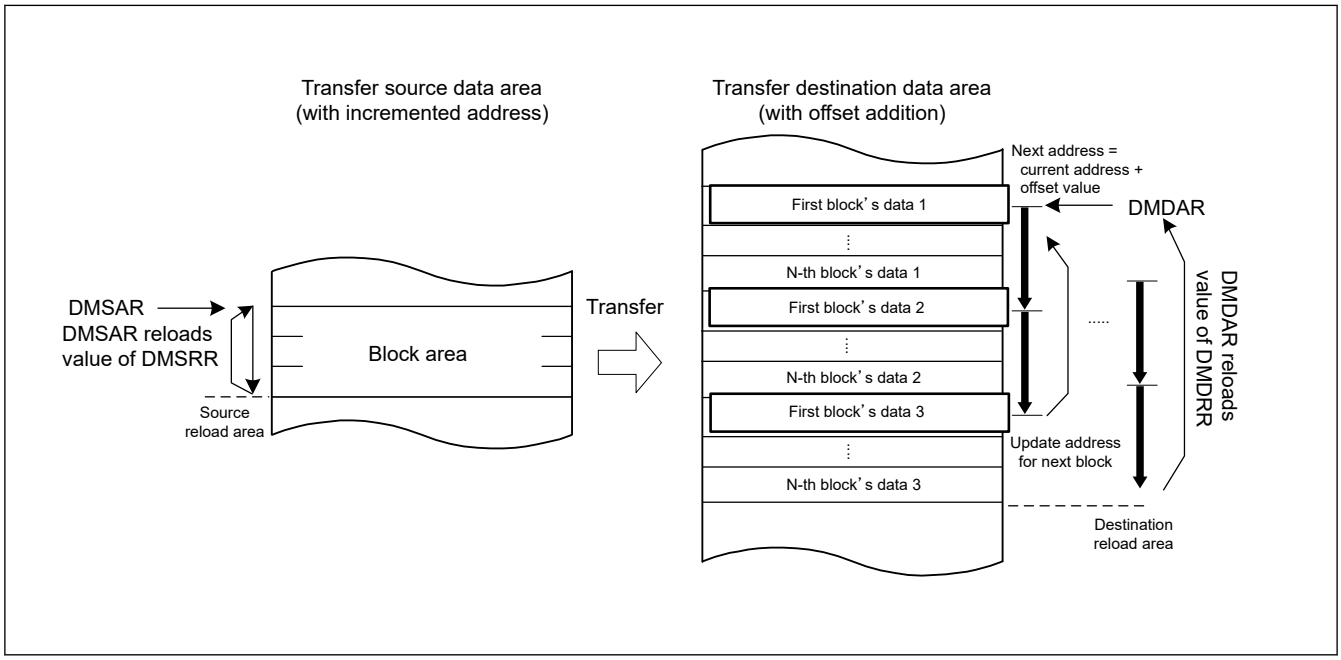


Figure 16.6 Operation in repeat-block transfer mode with offset addition

Table 16.8 to Table 16.13 summarize the register update operations in repeat-block transfer mode.

For more information about address update function in repeat-block transfer mode, see section 16.3.5. Address Update Function in Repeat-Block Transfer Mode.

Table 16.8 Register update operation associated with source area in repeat-block transfer mode (fixed address DMAMD.SM[1:0] = 00b) (1 of 2)

Register	Function	Update operation after single data is transferred		
		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMSRR	Transfer source reload address	Not updated	Not updated	Not updated

Table 16.8 Register update operation associated with source area in repeat-block transfer mode (fixed address DMAMD.SM[1:0] = 00b) (2 of 2)

Register	Function	Update operation after single data is transferred		
		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMSAR	Transfer source address	Not updated	Not updated	Not updated
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decremented by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]

Table 16.9 Register update operation associated with destination area in repeat-block transfer mode (fixed address DMAMD.DM[1:0] = 00b)

Register	Function	Update operation after single data is transferred		
		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMDRR	Transfer destination reload address	Not updated	Not updated	Not updated
DMDAR	Transfer destination address	Not updated	Not updated	Not updated
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decremented by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]

Table 16.10 Register update operation associated with source area in repeat-block transfer mode (incremented or decremented address DMAMD.SM[1:0] = 10b or 11b) (1 of 2)

Register	Function	Update operation after single data is transferred					
		DMSBSL[15:0] is not 1			DMSBSL[15:0] is 1		
		DMCRAL [15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)		DMCRAL [15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1		DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMSRR	Transfer source reload address	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated

Table 16.10 Register update operation associated with source area in repeat-block transfer mode (incremented or decremented address DMAMD.SM[1:0] = 10b or 11b) (2 of 2)

Register	Function	Update operation after single data is transferred					
		DMSBSL[15:0] is not 1			DMSBSL[15:0] is 1		
		DMCRAL [15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)		DMCRAL [15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1		DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMSAR	Transfer source address when DMTMD.SM [1:0] = 10b	Incremented by Data Size			DMSRR		
	Transfer source address when DMTMD.SM [1:0] = 11b	Decremented by Data Size			DMSRR		
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]
DMSBSH[15:0]	Source buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMSBSL[15:0]	Count of transfer data in source buffer	Decremented by 1	Decremented by 1	Decremented by 1	DMSBSH	DMSBSH	DMSBSH
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decremented by 1	0	Not updated	Decremented by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]			DMCRBH[15:0]

Table 16.11 Register update operation associated with destination area in repeat-block transfer mode (incremented or decremented address DMAMD.DM[1:0] = 10b or 11b) (1 of 2)

Register	Function	Update operation after single data is transferred					
		DMDBSL[15:0] is not 1			DMDBSL[15:0] is 1		
		DMCRAL [15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)		DMCRAL [15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1		DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMDRR	Transfer destination reload address	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated

Table 16.11 Register update operation associated with destination area in repeat-block transfer mode (incremented or decremented address DMAMD.DM[1:0] = 10b or 11b) (2 of 2)

Register	Function	Update operation after single data is transferred					
		DMDBSL[15:0] is not 1			DMDBSL[15:0] is 1		
		DMCRAL [15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)		DMCRAL [15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1		DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMDAR	Transfer destination address when DMTMD.DM [1:0] = 10b	Incremented by Data Size			DMDRR		
	Transfer destination address when DMTMD.DM [1:0] = 11b	Decremented by Data Size			DMDRR		
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]
DMDBSH[15:0]	Destination buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMDBSL[15:0]	Count of transfer data in destination buffer	Decremented by 1	Decremented by 1	Decremented by 1	DMDBSH	DMDBSH	DMDBSH
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decremented by 1	0	Not updated	Decremented by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]			DMCRBH[15:0]

Table 16.12 Register update operation associated with source area in repeat-block transfer mode (offset addition DMAMD.SM[1:0] = 01b) (1 of 2)

Register	Function	DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)			
			DMSBSL[15:0] is not 1		DMSBSL[15:0] is 1	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1	DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMSRR	Transfer source reload address	Not updated	Not updated	Not updated	Not updated	Not updated

Table 16.12 Register update operation associated with source area in repeat-block transfer mode (offset addition DMAMD.SM[1:0] = 01b) (2 of 2)

Register	Function	DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)			
			DMSBSL[15:0] is not 1		DMSBSL[15:0] is 1	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1	DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMSAR	Transfer source address when DMAMD.SADR = 0	Offset addition by DMSBSH	DMSRR		DMSRR	
	Transfer source address when DMAMD.SADR = 1		DMSRR + (DMS-BSH - DMSBSL) × DataSize			
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]	DMCRAH[9:0]	DMCRAH[9:0]
DMSBSH[15:0]	Source buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated
DMSBSL[15:0]	Count of transfer data in source buffer	Not updated	Decremented by 1	Decremented by 1	DMSBSH	DMSBSH
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decremented by 1	0	Decremented by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]		DMCRBH[15:0]

Table 16.13 Register update operation associated with destination area in repeat-block transfer mode (offset addition DMAMD.DM[1:0] = 01b) (1 of 2)

Register	Function	DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)			
			DMDBSL[15:0] is not 1		DMDBSL[15:0] is 1	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1	DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMDRR	Transfer destination reload address	Not updated	Not updated	Not updated	Not updated	Not updated
DMDAR	Transfer destination address when DMAMD.DADR = 0	Offset addition by DMDBSH	DMDRR		DMDRR	
	Transfer destination address when DMAMD.DADR = 1		DMDRR + (DMDBSH - DMDBSL) × DataSize			
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]	DMCRAH[9:0]	DMCRAH[9:0]

Table 16.13 Register update operation associated with destination area in repeat-block transfer mode (offset addition DMAMD.DM[1:0] = 01b) (2 of 2)

Register	Function	DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)			
			DMDBSL[15:0] is not 1		DMDBSL[15:0] is 1	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1	DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMDBSH[15:0]	Destination buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated
DMDBSL[15:0]	Count of transfer data in destination buffer	Not updated	Decremented by 1	Decremented by 1	DMDBSH	DMDBSH
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decremented by 1	0	Decremented by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]		DMCRBH[15:0]

16.3.2 Extended Repeat Area Function

The DMAC supports a function to specify the extended repeat areas on the transfer source and destination addresses. With the extended repeat areas set, the address registers repeatedly indicate the addresses of the specified extended repeat areas.

The extended repeat areas can be specified separately to the transfer source address register (DMSAR) and transfer destination address register (DMDAR).

The extended repeat area on the source address is specified by the DMAMD.SARA[4:0] bits. The extended repeat area on the destination address is specified by the DMAMD.DARA[4:0] bits. The size can be specified separately for the source and destination sides.

However, the area (of transfer source or transfer destination) which is specified as the repeat area or block area should not be specified as the extended repeat area.

When the address register value reaches the end address of the extended repeat area and the extended repeat area overflows, DMA transfer is stopped and an interrupt by an extended repeat area overflow can be requested. When an overflow occurs in the extended repeat area on the transfer source while the DMINT.SARIE bit is set to 1, the DMSTS.ESIF flag is set to 1 and the DMCNT.DTE bit is cleared to 0 to stop DMA transfer. At this time, if the DMINT.ESIE bit is set to 1, an interrupt by an extended repeat area overflow is requested. When the DMINT.DARIE bit is set to 1, the destination address register becomes a target to apply the function. DMA transfer can be resumed by writing 1 to the DMCNT.DTE bit in the interrupt handling.

Figure 16.7 shows an example of the extended repeat area operation.

Eight bytes are specified as an extended repeat area by the lower three bits of DMSAR (DMAMD.SARA[4:0] bits = 00011b). The data size is eight bits (DMTMD.SZ[1:0] = 00b).

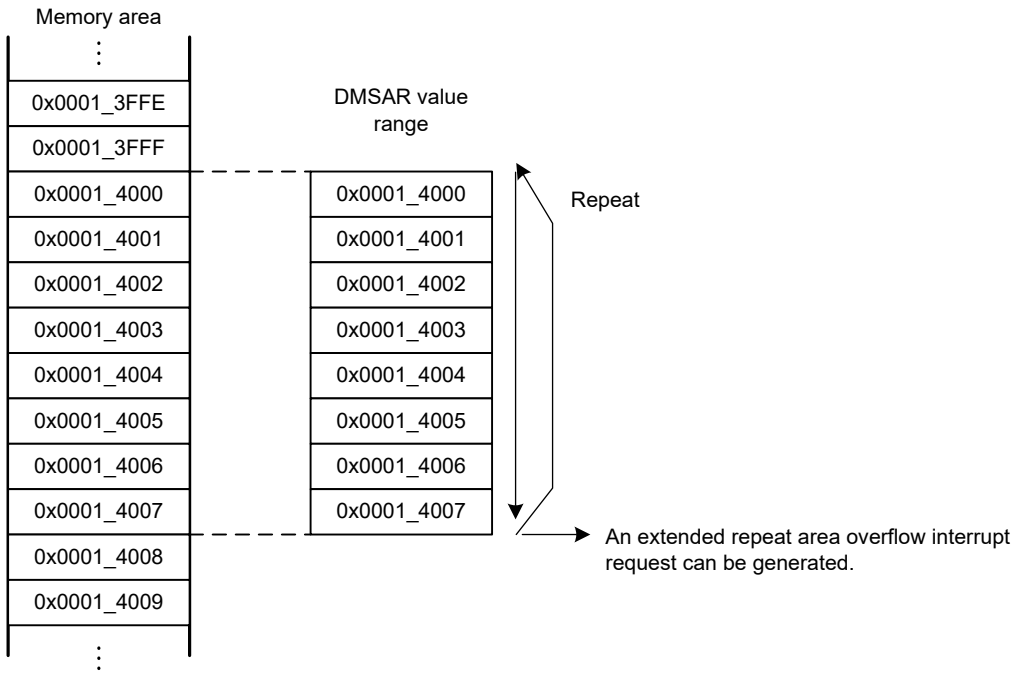


Figure 16.7 Example of extended repeat area operation

When an interrupt by an extended repeat area overflow is used in block transfer mode, the following should be taken into consideration.

When a transfer is stopped by an interrupt by an extended repeat area overflow, the address register must be set so that the block size is a power of 2 or the block size boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block, the interrupt by the overflow is suspended until transfer of the block is completed, and the transfer overruns.

Figure 16.8 shows an example when the extended repeat area function is used in block transfer mode.

Eight bytes are specified as an extended repeat area by the lower three bits of DMSAR (DMAMD.SARA[4:0] bits = 00011b), block transfer mode with block size 5 is set (DMCRA = 0x0005_0005), and transfer source address is not specified as a block area. Data size is eight bits (DMTMD.SZ[1:0] bits = 00b).

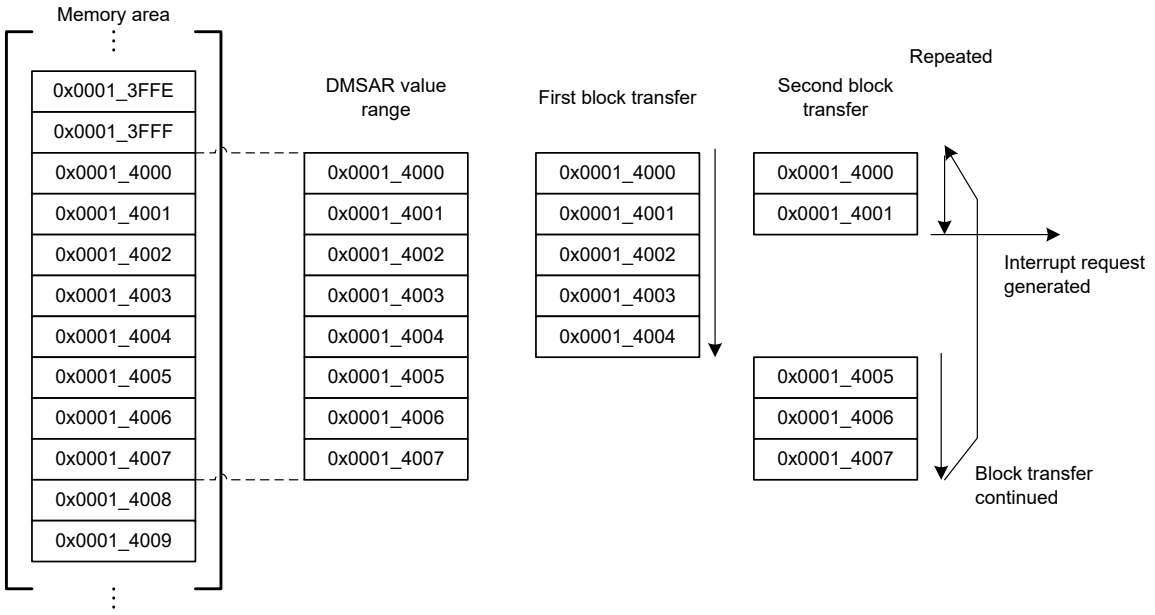


Figure 16.8 Example of extended repeat area function in block transfer mode

16.3.3 Free-running Function

The DMAC supports free-running function. This function allows to transfer repeatedly without reconfiguring in interrupt handler.

16.3.3.1 In Normal Transfer Mode

In normal transfer mode, when DMCRA.DMCRAL bits are set to 0x0000, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped.

For more information, see section 16.3.1.1. Normal Transfer Mode.

16.3.3.2 In Other Transfer Modes

In repeat, block and repeat-block transfer mode, the DMAC supports free-running function using the DMTMD.TKP bit. If the DMTMD.TKP bit is to be set to 1, the transfer is not stopped by completion of specified total number of transfer operations and reloads DMCRBH repeatedly.

Figure 16.9 show an example of block transfer operation without free-running function.

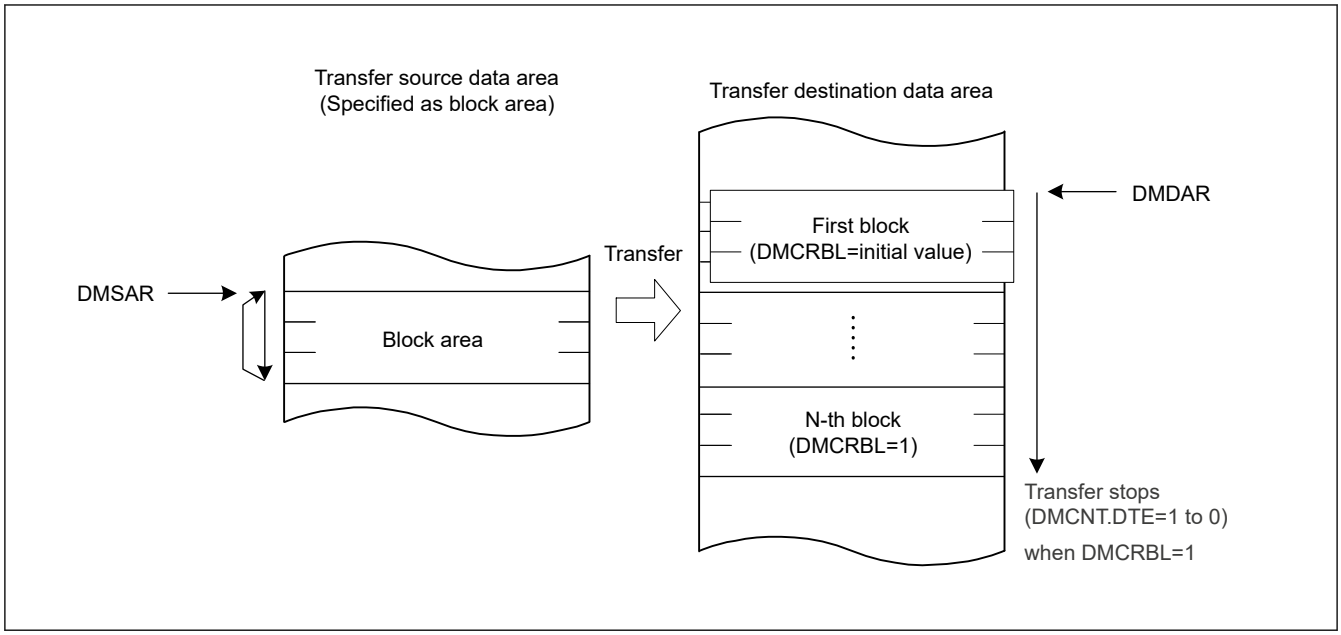


Figure 16.9 Operation in block transfer mode when DMTMD.TKP bit is set to 0

Figure 16.10 show an example of block transfer operation with free-running function.

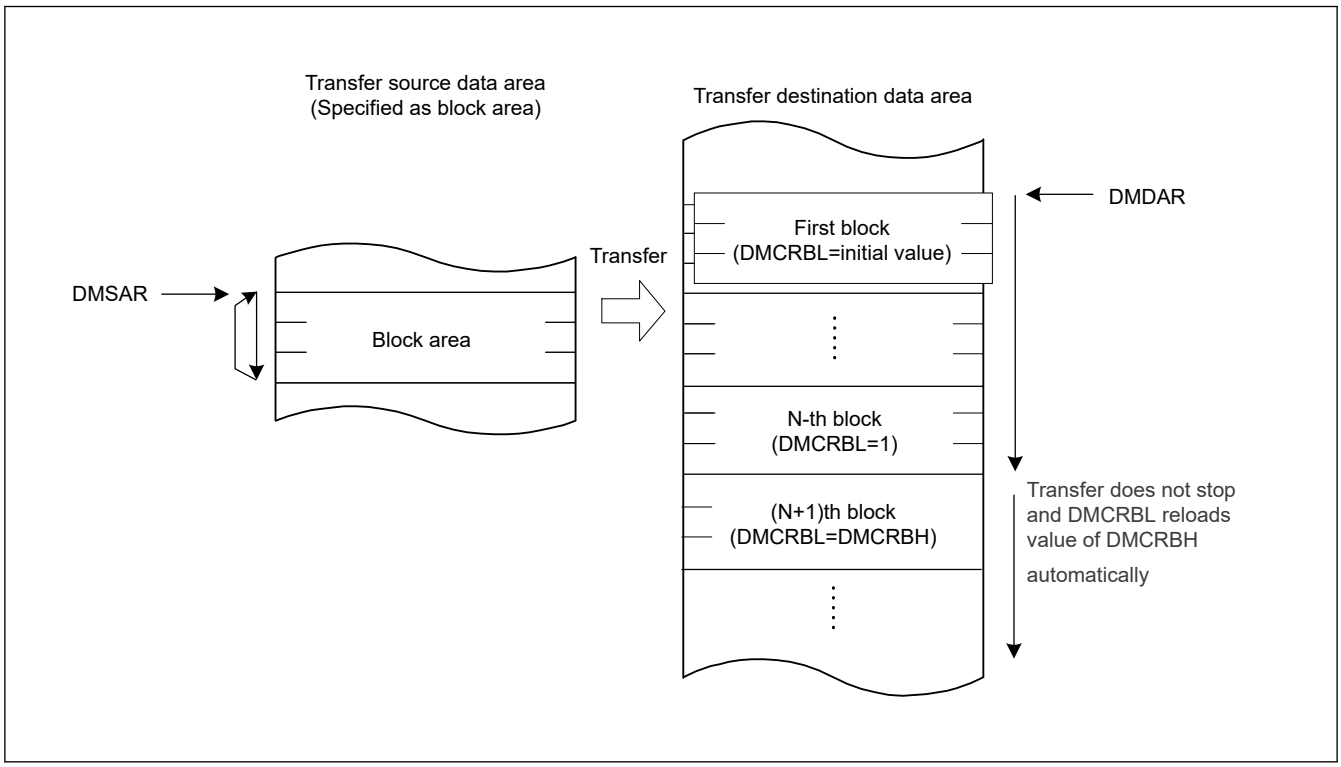


Figure 16.10 Operation in block transfer mode when DMTMD.TKP bit is set to 1

16.3.4 Address Update Function using Offset

The source and destination addresses can be updated by fixing, increment, decrement, or offset addition. In normal, repeat and block transfer mode, when the offset addition is selected, the offset specified by the DMA offset register (DMOFR) is added to the address every time the DMAC performs one data transfer. This function realizes a data transfer where addresses are allocated to separated areas.

Offset subtraction can also be realized by setting a negative value in DMOFR. In this case, the negative value must be 2's complement.

DMSBS or DMDBS are used instead of DMOFR in repeat-block transfer mode. For more information section 16.3.1.4. Repeat-Block Transfer Mode

Table 16.14 shows the address update method in each address update mode.

Table 16.14 Address update method in each address update mode

Address update mode	Settings of DMAMD.SM[1:0] and DMAMD.DM[1:0] for address update modes	Address update method (for different SZ[1:0] settings in DMTMD)		
		SZ[1:0] = 00b	SZ[1:0] = 01b	SZ[1:0] = 10b
Address fixed	00b	Fixed		
Offset addition	01b	+DMOFR*1		
Increment	10b	+1	+2	+4
Decrement	11b	–1	–2	–4

Note 1. When setting a negative value in the DMA Offset Register, the value must be in two's complement, obtained by the following formula:
two's complement of a negative offset value = ~ (offset) + 1 (~ = bit inversion)

16.3.4.1 Basic Transfer Using Offset Addition

Figure 16.11 shows an example of address updating using offset addition.

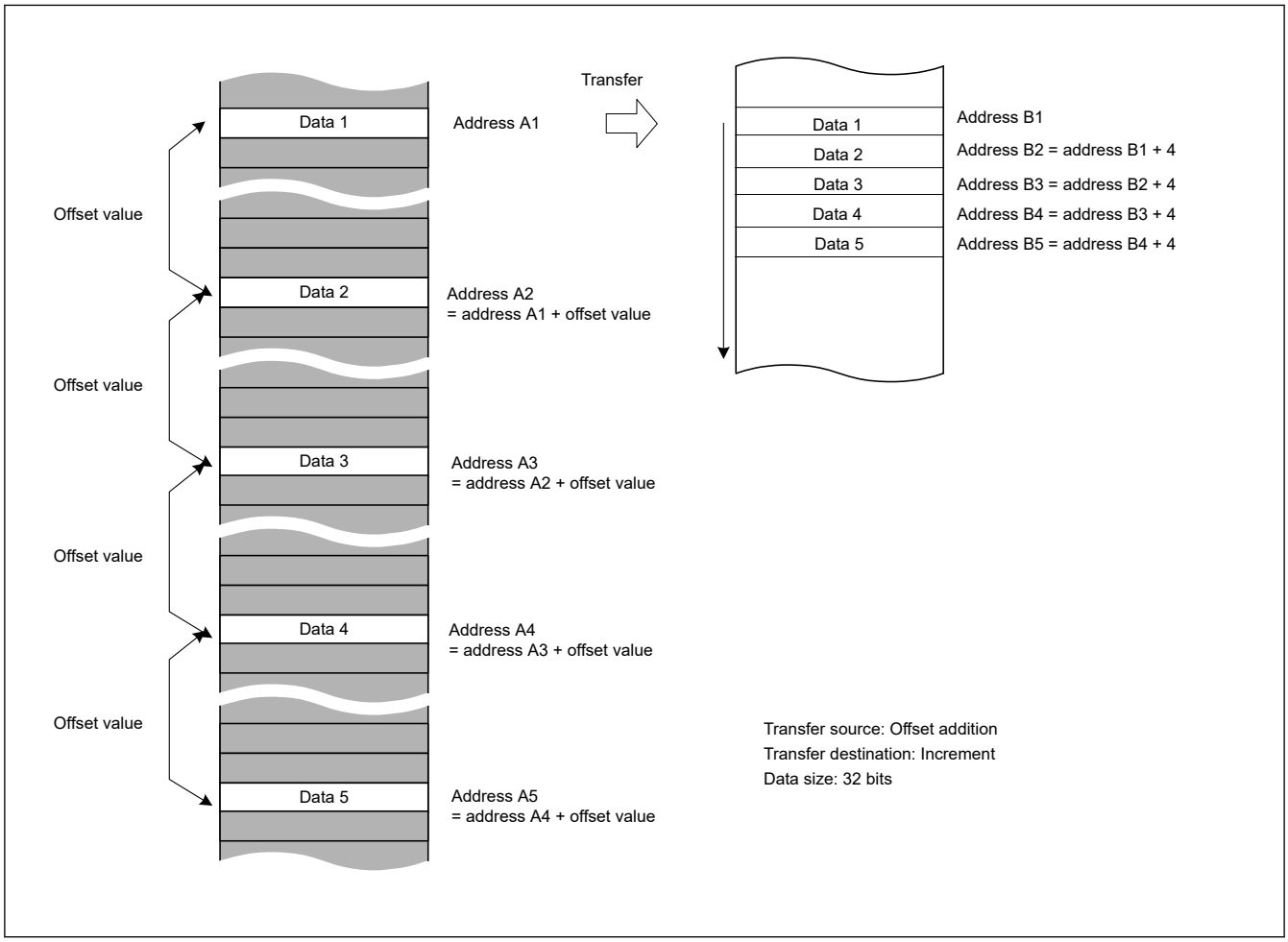


Figure 16.11 Example of address updating by offset addition

Figure 16.11 shows the setting of the following.

- The transfer data is 32 bits long.
- Offset addition is set as the transfer source address update mode.

- Increment is set as the transfer destination address update mode.

The second and subsequent data is each read from the transfer source address obtained by adding the offset value to the previous address. The data read from the addresses at the specified intervals is written to the continuous locations on the destination.

16.3.4.2 Example of XY Conversion Using Offset Addition

Figure 16.12 shows the XY conversion using offset addition in repeat transfer mode.

Settings are as follows:

- DMAMD.SM — Transfer source address update mode: Offset addition.
- DMAMD.DM — Transfer destination address update mode: Destination address is incremented.
- DMTMD.SZ — Transfer data size select: 32 bits.
- DMTMD.MD — Transfer mode select: Repeat transfer.
- DMTMD.DTS — Repeat area select: The source is specified as the repeat area.
- DMOFR — Offset address: 0x10.
- DMCRA — Repeat size: 0x4.
- DMINT.RPTIE — The repeat size end interrupt is enabled.

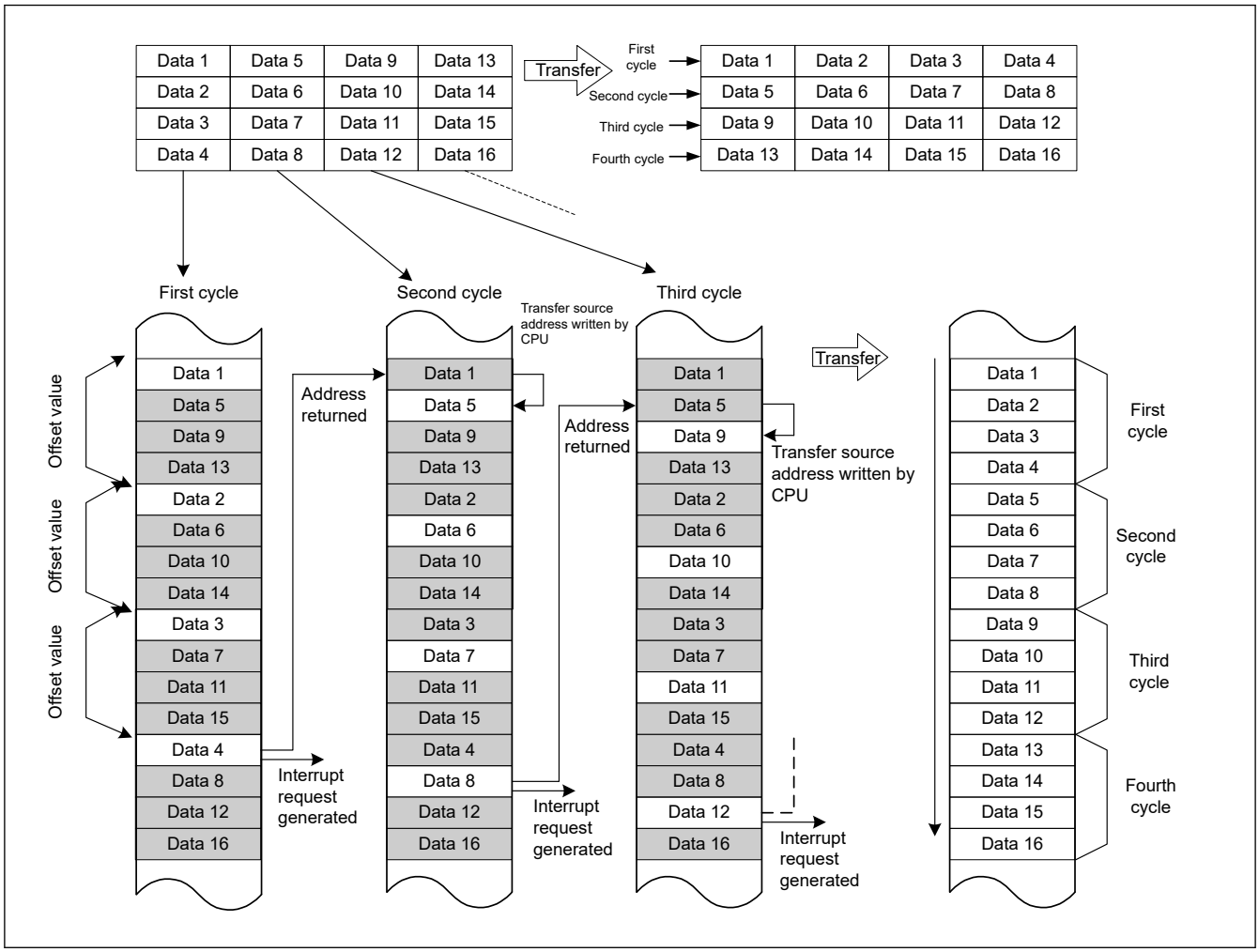


Figure 16.12 XY conversion operation using offset addition in repeat transfer mode

When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to continuous transfer destination addresses. When data 4 is transferred:

- The repeat size of data transfer is complete.
- The transfer source address returns to the transfer start address (the address of data 1 on the transfer source).
- A repeat size end interrupt is requested.

During the time this interrupt pauses the transfer, the following operations are performed.

- DMSAR — Rewrite the DMA transfer source address to the address of data 5
  (with the above example, the data 1 address + 4).
- DMCNT — Set the DTE bit to 1.

The DMA transfer is resumed from the state when the DMA transfer is stopped. After that, the operations described above are repeated until the transfer source data is transposed to the destination area (XY conversion).

Figure 16.13 shows a flowchart of the XY conversion.

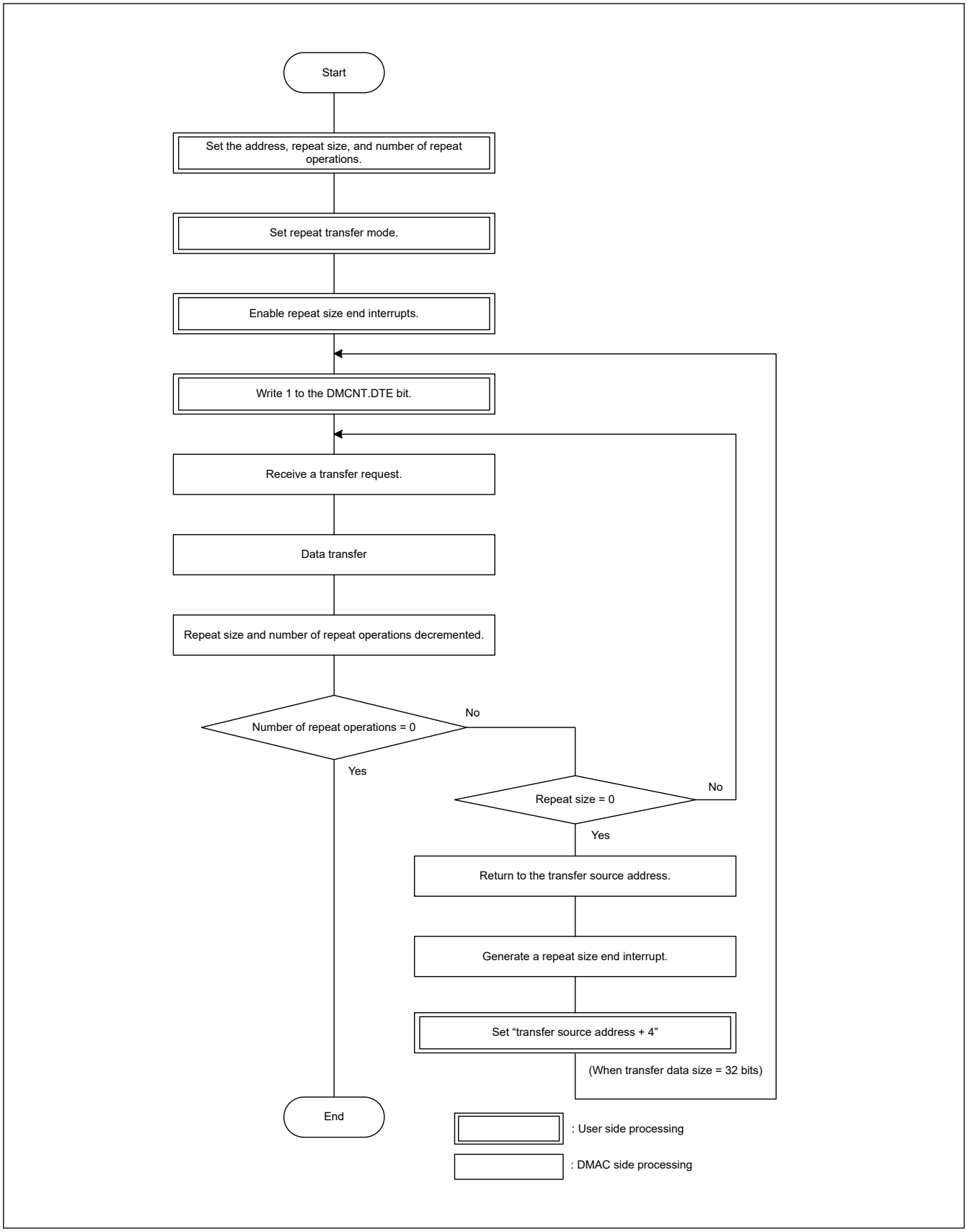


Figure 16.13 XY conversion flowchart using offset addition in repeat transfer mode

16.3.5 Address Update Function in Repeat-Block Transfer Mode

Repeat-block transfer mode is an extension of repeat transfer mode and block transfer mode. However, the detailed behavior of the address update is different from these two modes. Here are the details of the address update function in repeat-block transfer mode.

16.3.5.1 Fixed Address Mode

When DMAMD.SM[1:0] is set to 00b, the address update mode of the source is fixed address. And when DMAMD.DM[1:0] is set to 00b, the address update mode of the destination is fixed address.

In fixed address, the address is not updated from the initial value of DMSAR and DMDAR. If the block size (DMCRA) is larger than 1, the same data will be transferred multiple times for one request.

Figure 16.14 shows address update in fixed address.

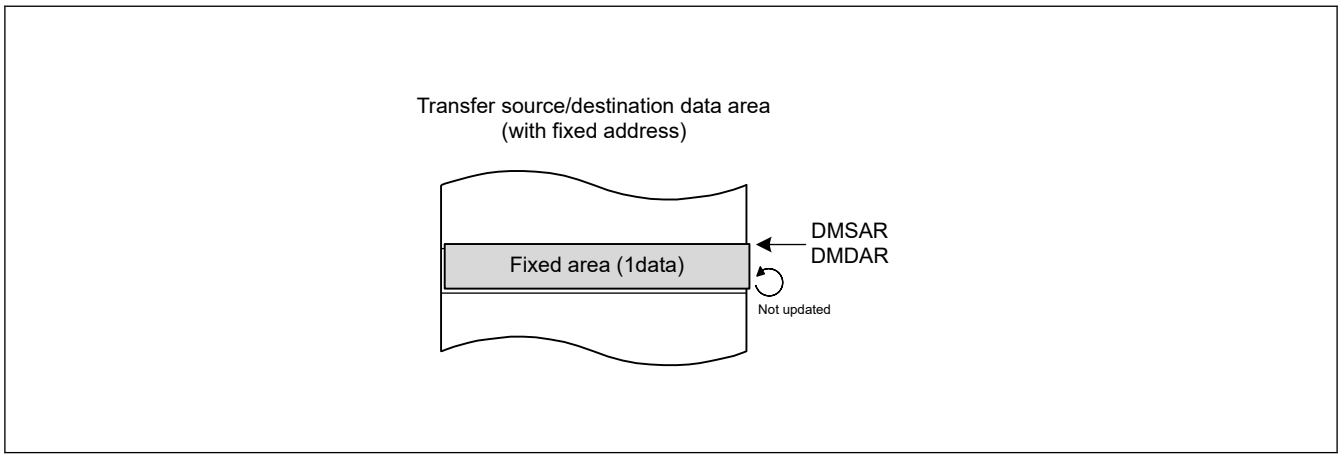


Figure 16.14 Address update in fixed address

16.3.5.2 Incremental and Decremental Address Mode

When DMAMD.SM[1:0] is set to 10b, the address update mode of the source is incremental address. And when DMAMD.DM[1:0] is set to 10b, the address update mode of the destination is incremental address. When DMAMD.SM[1:0] is set to 11b, the address update mode of the source is decremental address. And when DMAMD.DM[1:0] is set to 11b, the address update mode of the destination is decremental address.

In these update modes, the address is incremented or decremented according to the setting of DMTMD.SZ[1:0].

In these update modes DMSBS and DMDBS indicates a reload area. The unit of DMSBS and DMDBS is "number of data". At the start of transfer, DMSBSL and DMDBSL, which is the lower 16 bits of DMSBS and DMDBS, operates as a down counter and decrements each time one data transfer is performed. When the value becomes 1, DMSAR and DMDAR reloads the value of DMSRR and DMDRR.

Figure 16.15 shows address update in incremental address.

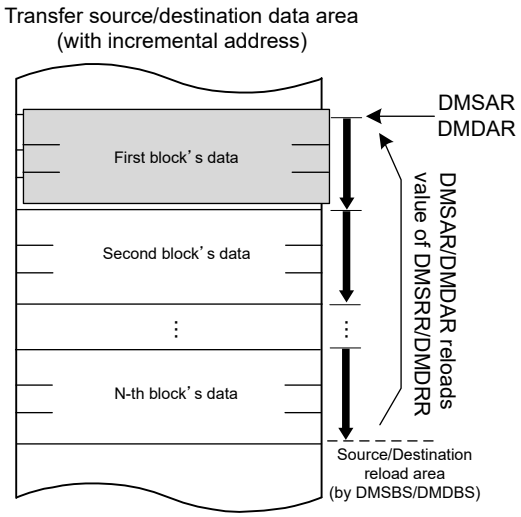


Figure 16.15 Address update in incremental address

16.3.5.3 Offset Addition Mode

When DMAMD.SM[1:0] is set to 01b, the address update mode of the source is offset addition. And when DMAMD.DM[1:0] is set to 01b, the address update mode of the destination is offset addition.

In offset addition, DMSBS and DMDBS indicates reload area and also works as an access offset value. Unlike other transfer modes, DMOFR register is not used in repeat-block transfer mode. In offset addition, the unit of DMSBS and DMDBS is the number of blocks. When the transfer starts, DMCRAL operates as a down counter, DMSAR and DMDAR reloads the value of DMSRR and DMDRR every time one block is transferred. In addition, DMSBSL and DMDBSL, which is the lower 16 bits of DMSBS and DMDBS, also operates as a down counter and decrements every time one block is transferred. When the DMSBS and DMDBS value becomes 1, DMSAR and DMDAR reloads the value of DMSRR and DMDRR.

When DMAMD.SADR and DMAMD.DADR is set to 0, offset addition operation of the same area is repeated. DMDAR only reloads DMDRR. Figure 16.16 shows address update in offset addition with DMAMD.SADR and DMAMD.DADR=0.

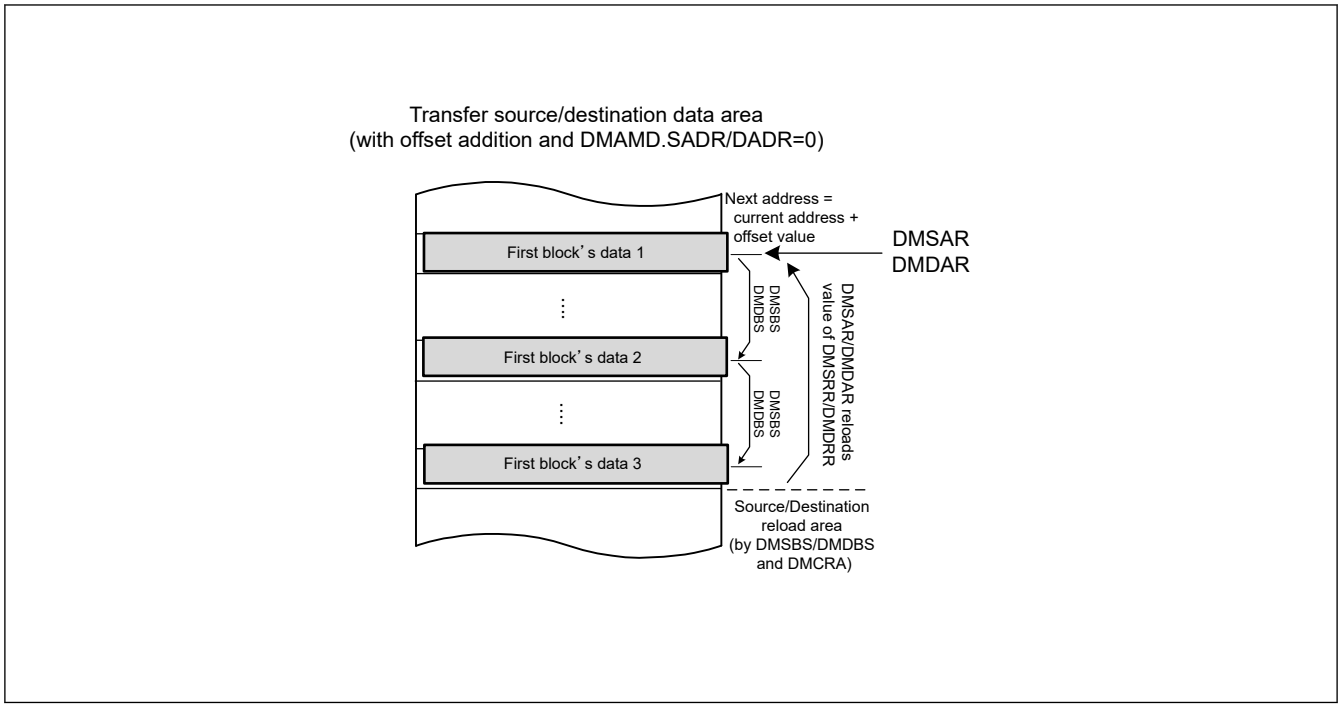


Figure 16.16 Address update in offset addition with DMAMD.SADR and DMAMD.DADR = 0

When DMAMD.SADR and DMAMD.DADR is set to 1, the address is incremented by one data unit after DMSRR and DMDRR is reloaded by DMCRAL=1. In other words, an index value ((DMDBSH-DMDBSL) × DataSize) is added to DMDAR after DMDRR is reloaded. This behavior is used to implement multiple ring buffers. Figure 16.17 shows address update in offset addition with DMAMD.SADR and DMAMD.DADR=1.

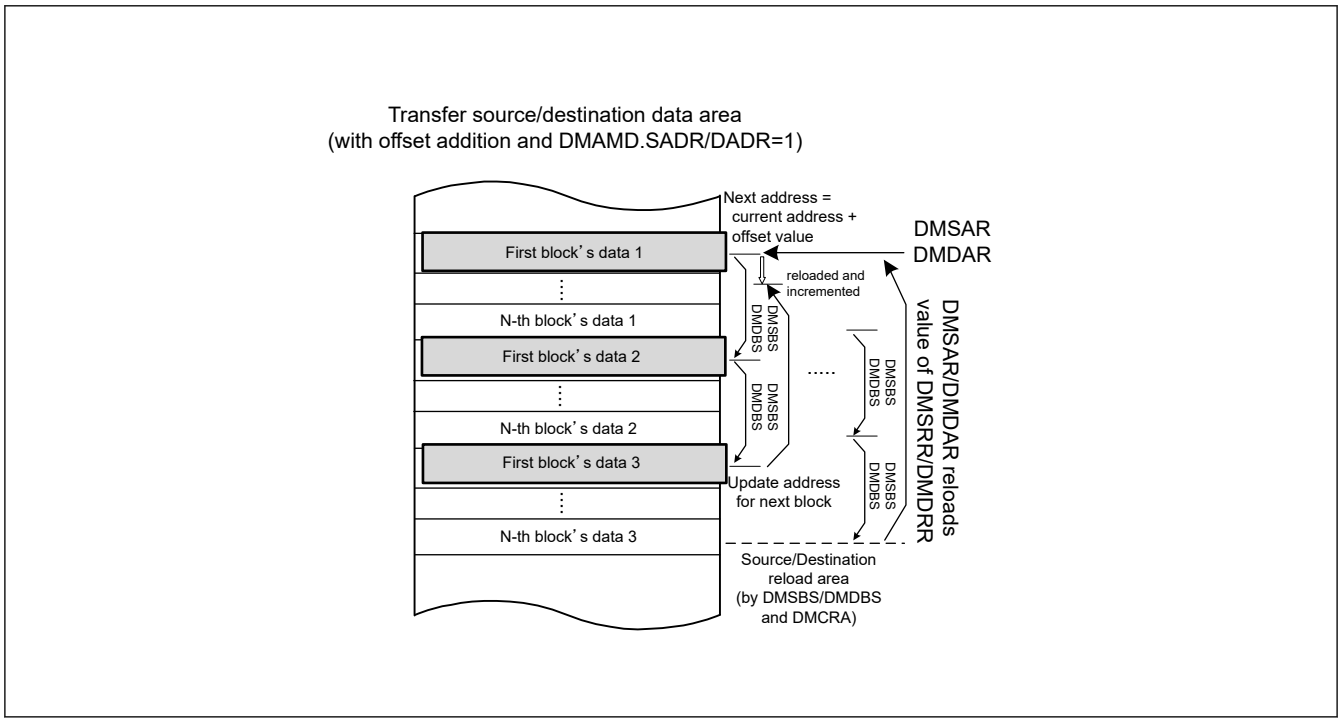


Figure 16.17 Address update in offset addition with DMAMD.SADR and DMAMD.DADR = 1

16.3.6 Example of Using Repeat-Block Transfer Mode

In repeat-block transfer mode, it is possible to realize repeated access to interval data and single or multiple ring buffers by combining the above address update modes. Following sections shows some usage examples.

16.3.6.1 Interval Address to Single Ring Buffer

Figure 16.18 shows an example of reading interval ADDRn registers (data register) of ADC12 module and storing it in single ring buffer. It transfers 2 data every 4 halfwords per 1 request. DMSAR is incremented by one data every one request. This can be achieved by setting the transfer source to offset addition and DMAMD.SADR=1, the block size (DMCRA) to 2, and the transfer source offset (DMSBS) to 4. Table 16.15 shows setting of this example.

Table 16.15 Setting of use case: from interval address to single ring buffer

Register	Value	Description
DMSAR, DMSRR	0x4033_2020	Initial source address
DMDAR, DMDRR	0x2000_0000	Initial destination address
DMTMD.SZ[1:0]	01b	Data size is halfword
DMAMD.SADR	1	Incremental source address after reloading
DMAMD.SM[1:0]	01b	Source update mode is offset addition
DMAMD.DM[1:0]	10b	Destination update mode is incremental address
DMCRAH, DMCRAL	2	Transfer block size
DMSBSH, DMSBSL	4	Source whole buffer size (unit is ‘blocks’) and Source access offset (unit is ‘data’)
DMDBSH, DMDBSL	N × 2 (DMCRA)	Destination buffer size (unit is ‘data’)

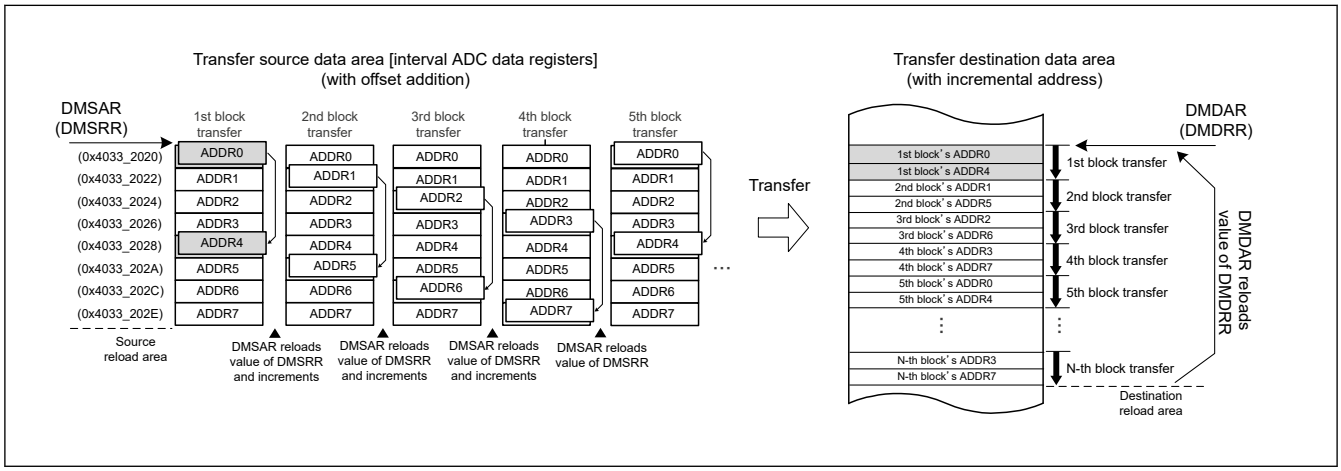


Figure 16.18 Example of use case: from interval address to single ring buffer

16.3.6.2 Unaligned Ring Buffer to Single Ring Buffer

Figure 16.19 shows an example of reading ADBUFn registers of ADC12 module (conversion result storage ring buffer) incrementally and storing it in single ring buffer. In this example, wrapping occurs because ADBUFn overflows in the fourth scan, but transfer source address of DMAC is also updated accordingly. This can be realized by setting the transfer source to incremental address and setting the DMSBS register to 16 which is the length of ADBUFn. This makes it possible to continue transfer without performing CPU processing using interrupts. Table 16.16 shows setting of this example.

Table 16.16 Setting of use case: from unaligned ring buffer to single ring buffer (1 of 2)

Register	Value	Description
DMSAR, DMSRR	0x4033_20B0	Initial source address
DMDAR, DMDRR	0x2000_0000	Initial destination address
DMTMD.SZ[1:0]	01b	Data size is halfword
DMAMD.SM[1:0]	10b	Source update mode is incremental address
DMAMD.DM[1:0]	10b	Destination update mode is incremental address

Table 16.16 Setting of use case: from unaligned ring buffer to single ring buffer (2 of 2)

Register	Value	Description
DMCRAH, DMCRAL	5	Transfer block size
DMSBSH, DMSBSL	16	Source buffer size (unit is ‘data’)
DMDBSH, DMDBSL	N × 5(DMCRA)	Destination buffer size (unit is ‘data’)

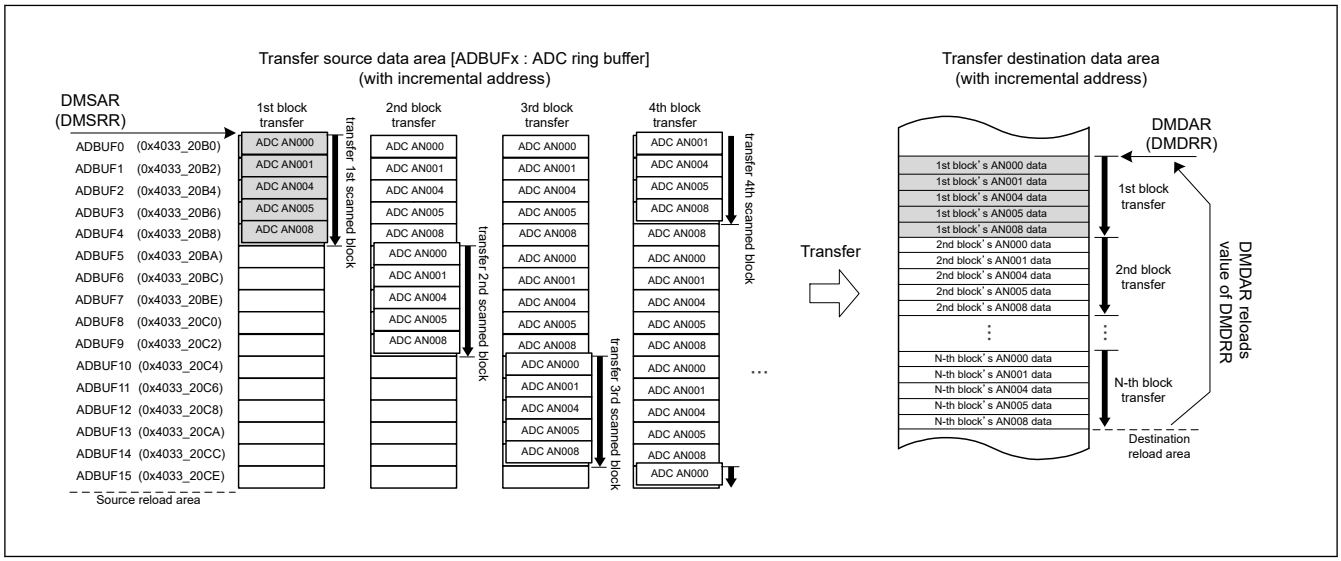


Figure 16.19 Example of use case: from unaligned ring buffer to single ring buffer

16.3.6.3 Single Block to Multi Ring Buffer

Figure 16.20 shows an example of storing the continuous ADDRn registers (data register) of ADC12 module individually in multiple ring buffers. In this example, a ring buffer in which only the first element (ADDR0) in a single block is arranged in transfer order is created at the destination. Also, in the next area, create a ring buffer in which only the second element (ADDR1) is arranged in transfer order. In the following case, create a ring buffer of length N, which is defined by DMDBS. And the number of data elements in the block is 3, which is defined by DMCRA. Table 16.17 shows setting of this example.

Table 16.17 Setting of use case: from single block to multi ring buffer

Register	Value	Description
DMSAR, DMSRR	0x4033_2020	Initial source address
DMDAR, DMDRR	0x2000_0000	Initial destination address
DMTMD.SZ[1:0]	01b	Data size is halfword
DMAMD.DADR	1	Incremental destination address after reloading
DMAMD.SM[1:0]	10b	Source update mode is incremental address
DMAMD.DM[1:0]	01b	Destination update mode is offset addition
DMCRAH, DMCRAL	3	Transfer block size
DMSBSH, DMSBSL	3	Source buffer size (unit is ‘data’)
DMDBSH, DMDBSL	N	Destination whole buffer size (unit is ‘blocks’) and Destination access offset (unit is ‘data’)

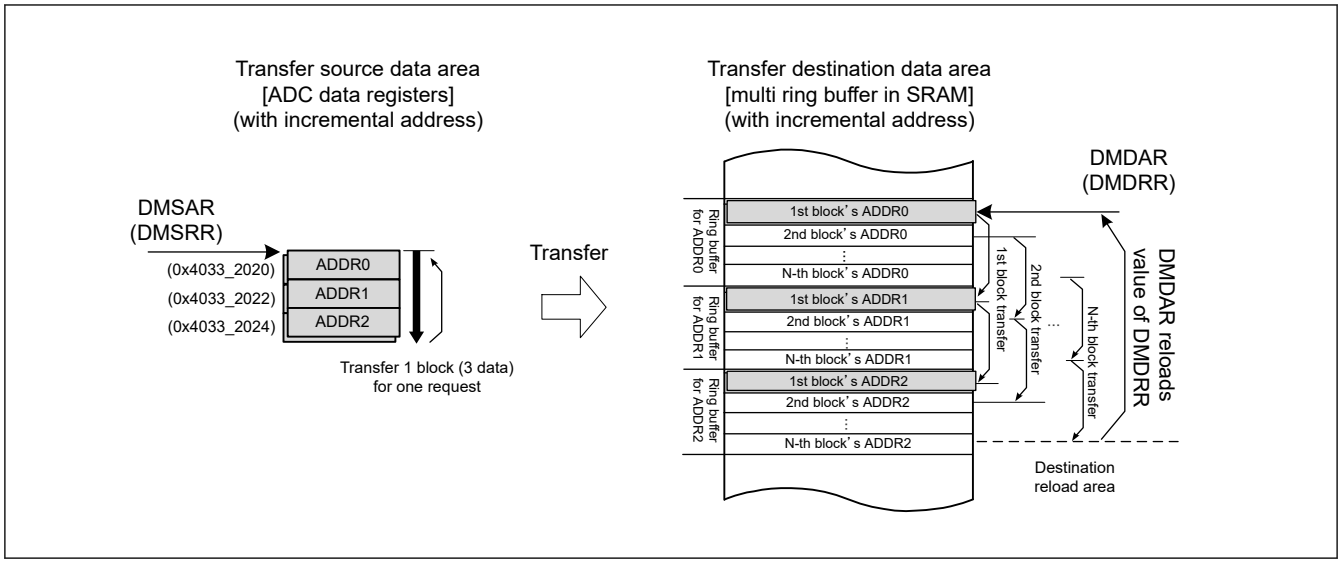


Figure 16.20 Example of use case: from single block to multi ring buffer

16.3.7 Activation Sources

Software, interrupt requests from the peripheral modules, and external interrupt requests can all be specified as DMAC activation sources. Set the DMTMD.DCTG[1:0] bits to select the activation source.

16.3.7.1 DMAC Activation by Software

When DMA transfer is started by software, follow below procedure.

1. Set the DMTMD.DCTG[1:0] bits to 00b.
2. Set the DMCNT.DTE bit to 1 (DMA transfer is enabled).
3. Set the DMAST.DMST bit set to 1 (DMAC activation enabled).
4. Set the DMREQ.SWREQ bit to 1 (DMA requested).

When the DMAC is activated by software while the DMREQ.CLRS bit is 0, the DMREQ.SWREQ bit is cleared to 0 after data transfer is started in response to a DMA transfer request.

When the DMAC is activated by software while the CLRS bit is 1, the SWREQ bit is not cleared to 0 after data transfer is started. In this case, a DMA transfer request is issued again after completion of a transfer.

16.3.7.2 DMAC Activation through Interrupt Requests from On-Chip Peripheral Modules or External Interrupt Requests

You can specify interrupt requests from on-chip peripheral modules and external interrupt requests as DMAC activation sources. The activation sources can be selected individually for each channel in DELSRn.DELS[8:0] (n = 0 to 7).

To start DMA transfer through an interrupt request from an on-chip peripheral module or an external interrupt request, follow the procedures as indicated below.

1. Set the DELSR.DELS[8:0] bits to the event number (select the DMAC event link).
2. Set the DMTMD.DCTG[1:0] bits to 01b (interrupts from the peripheral modules and the external interrupt pins).
3. Set the DMCNT.DTE bit to 1 (enable DMA transfer).
4. Set the DMAST.DMST bit set to 1 (DMAC activation enabled)

For interrupt requests specified as DMAC activation sources, see Table 13.3, in section 13, Interrupt Controller Unit (ICU).

16.3.8 Operation Timing

The following timing charts show the minimum number of execution cycles.

Figure 16.21 and Figure 16.22 show DMAC operation timing examples.

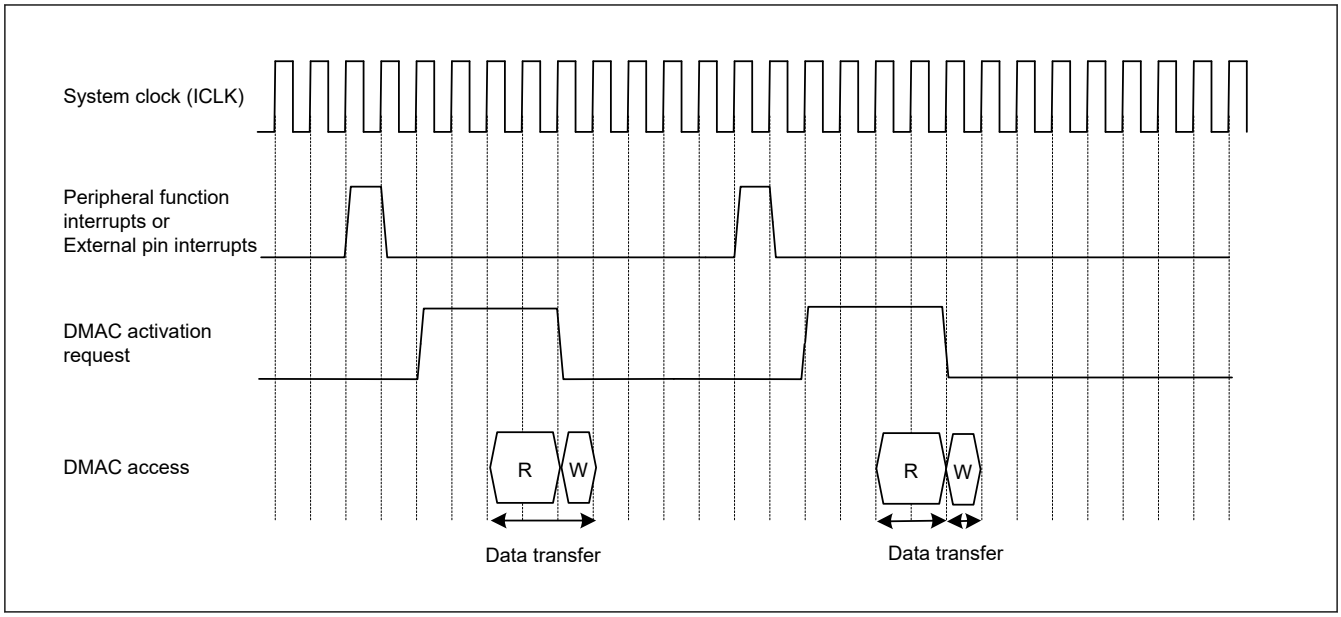


Figure 16.21 DMAC operation timing example 1 with DMAC activation by Interrupt from peripheral module or external interrupt input pin, in normal transfer mode or repeat transfer mode

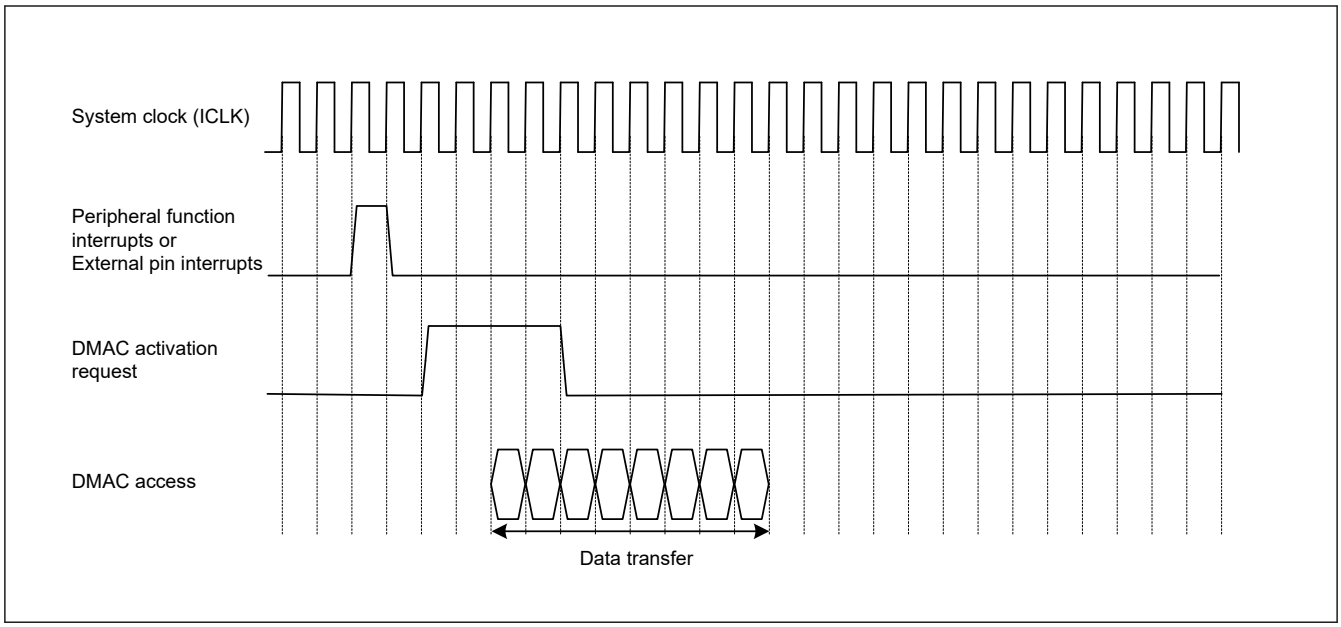


Figure 16.22 DMAC operation timing example 2 with DMAC activation by interrupt from peripheral module or external interrupt input pin, in block transfer mode with block size = 4

16.3.9 Activating the DMAC

Table 16.18 shows the register setting procedure of normal, repeat and block transfer mode and Table 16.19 shows register setting procedure of repeat-block transfer mode.

Table 16.18 Register Setting Procedure of Normal Transfer Mode, Repeat Transfer Mode and Block Transfer Mode (1 of 2)

No.	Step Name	Description
1	Disable the peripheral function as the DMAC request source.	To use peripheral function interrupts as DMAC activation sources. Disable the control register for the peripheral function.
2	Disable the IRQn pin as the DMAC request source.	To use external pin interrupts as DMAC activation sources.
3	Set the DELSR.DELS[8:0] bits to 0x00.	Disable the DMAC request.

Table 16.18 Register Setting Procedure of Normal Transfer Mode, Repeat Transfer Mode and Block Transfer Mode (2 of 2)

No.	Step Name	Description
4	Clear the DMCNT.DTE bit to 0	Disable DMA transfer.
5	Set the interrupt request as a DMAC request source in the DELSR register.	To use internal peripheral interrupts or external pin interrupts as DMAC activation sources. Enable the interrupt bit for the activation source. Set the DMAC activation source.
6	Set the peripheral module as a DMAC request source	To use peripheral function interrupt as a DMAC activation source. Set the control register for the peripheral function without starting it.
7	Set the IRQn pin function by using the ICU.	To use external pin interrupt as a DMAC activation source. Set the IRQn pin function by using the Interrupt Controller Unit.
8	Set the DMAMD.DM[1:0] bits Set the DMAMD.SM[1:0] bits Set the DMAMD.DARA[4:0] bits Set the DMAMD.SARA[4:0] bits	Set the Transfer destination address update mode bits Set the Transfer source address update mode bits Set the Transfer destination address extended repeat area bits Set the Transfer source address extended repeat area bits
9	Set the DMTMD.DCTG[1:0] bits Set the DMTMD.SZ[1:0] bits Set the DMTMD.DTS[1:0] bits Set the DMTMD.MD[1:0] bits Set the DMTMD.TKP bit	Set the Transfer request select bits Set the Data transfer size bits Set the Repeat area select bits Set the Transfer mode select bits Set the transfer keeping select bit
10	Set the DMSAR register Set the DMDAR register Set the DMCRA register	Set the transfer source start address. Set the transfer destination start address. Set the number of transfer operations.
11	Set the DMCRB register	To use block transfer mode or repeat transfer mode. Set the number of block transfer operations.
12	Set the DMOFR register	To use the address update function with offset. Set the offset value.
13	Set the DMINT.DTIE bit to 1	To use the DMA transfer end interrupts. Enable DMAC transfer end interrupts.
14	Set the DMINT.RPTIE bit Set the DMINT.SARIE bit Set the DMINT.DARIE bit Set the DMINT.ESIE bit to 1	To use the DMA transfer escape end interrupts Set the repeat size end interrupt. Set the transfer source address extended repeat area overflow interrupt. Set the transfer destination address extended repeat area overflow interrupt. Enable the DMA transfer escape end interrupt.
15	Set the DMCNT.DTE bit to 1	Enable DMA transfer.
16	Set the DMAST.DMST bit to 1	Enable DMAC operation. *1 Common settings for DMAC
17	Start the peripheral function as a DMAC request source	To use peripheral function interrupt as a DMAC activation source
18	Enable the IRQn pin as a DMAC request source	To use external pin interrupt as a DMAC activation source
19	End of initial settings	For activation by software On completion of the initial settings, writing 1 to the DMA software start bit (DMREQ.SWREQ) starts DMA transfer.

Note 1. The DMAST.DMST bit setting does not necessarily have to follow the settings for the individual activation sources.

Table 16.19 Register Setting Procedure of Repeat-Block Transfer Mode (1 of 2)

No.	Step Name	Description
1	Disable the peripheral function as the DMAC request source.	To use peripheral function interrupts as DMA activation sources. Disable the control register for the peripheral function.
2	Disable the IRQ pin as the DMAC request source.	To use external pin interrupts as DMA activation sources.
3	Set the DELSR.DELS[8:0] bits to 0x00.	Disable the DMAC request.
4	Clear the DMCNT.DTE bit to 0	Disable DMAC transfer.

Table 16.19 Register Setting Procedure of Repeat-Block Transfer Mode (2 of 2)

No.	Step Name	Description
5	Set the interrupt request as a DMAC request source in the DELSR register.	To use internal peripheral interrupts or external pin interrupts as DMA activation sources. Enable the interrupt bit for the activation source. Set the DMAC activation source.
6	Set the peripheral module as a DMAC request source	To use peripheral function interrupt as a DMA activation source. Set the control register for the peripheral function without starting it.
7	Set the IRQ pin function by using the Interrupt Controller Unit.	To use external pin interrupt as a DMA activation source. Set the IRQ pin function by using the Interrupt Controller Unit.
8	Set the DMAMD.DM[1:0] bits Set the DMAMD.SM[1:0] bits Set the DMAMD.DADR bit Set the DMAMD.SADR bit	Set the Transfer destination address update mode bits Set the Transfer source address update mode bits Set the Transfer destination address update select after reload Set the Transfer source address update select after reload
9	Set the DMTMD.DCTG[1:0] bits Set the DMTMD.SZ[1:0] bits Set the DMTMD.MD[1:0] bits Set the DMTMD.TKP bit	Set the Transfer request select bits Set the Data transfer size bits Set the Transfer mode to repeat-block transfer mode Set the transfer keeping select bit
10	Set the DMSAR register Set the DMDAR register Set the DMSRR register Set the DMDRR register Set the DMCRA register Set the DMCRB register	Set the transfer source start address Set the transfer destination start address Set the initial value of source start address Set the initial value of destination start address Set the number of transfer operations Set the number of block transfer operations
11	Set the DMSBS register Set the DMDBS register	To use the address update function with incremental, decremental or offset Set the source buffer size and access offset Set the destination buffer size and access offset
12	Set the DMINT.DTIE bit to 1	To use DMA transfer end interrupts. Enable DMAC transfer end interrupts.
13	Set the DMCNT.DTE bit to 1	Enable DMAC transfer
14	Set the DMAST.DMST bit to 1	Enable DMAC operation. *1
15	Start the peripheral function as a DMAC request source	To use peripheral function interrupt as a DMA activation source
16	Enable the IRQ pin as a DMAC request source	To use external pin interrupt as a DMA activation source
17	End of initial settings	For activation by software On completion of the initial settings, writing 1 to the DMA software start bit (DMREQ.SWREQ) starts DMA transfer.

Note 1. The DMAST.DMST bit setting does not necessarily have to follow the settings for the individual activation sources.

16.3.10 Starting DMA Transfer

To enable the DMA transfer, set the DMCNT.DTE bit to 1 (enable the DMA transfer), and then set the DMAST.DMST bit to 1 (enable the DMAC activation).

New activation requests are not accepted during the transfer of another DMAC channel or DTC. When the preceding transfer is complete, channel arbitration selects the DMA transfer request of the highest priority channel, and the DMA transfer of that channel starts. When the DMA transfer starts, the DMSTS.ACT flag is set to 1 (the DMAC is in the active state).

16.3.11 Registers during DMA Transfer

The DMAC registers are updated by a DMA transfer. The value to be updated differs according to the other settings and the transfer state. The registers to be updated are DMSAR, DMDAR, DMCRA, DMCRB, DMSBS, DMDBS, DMCNT, and DMSTS.

DMA Source Address Register (DMSAR)

When data has been transferred in response to one transfer request, the contents of DMSAR are updated to the address to be accessed by the next transfer request.

For details on register update operation in each transfer mode, see Table 16.5 to Table 16.13.

DMA Destination Address Register (DMDAR)

When data has been transferred in response to one transfer request, the contents of DMDAR are updated to the address to be accessed by the next transfer request.

For details on register update operation in each transfer mode, see Table 16.5 to Table 16.13.

DMA Transfer Count Register (DMCRA)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, see Table 16.5 to Table 16.13.

DMA Block Transfer Count Register (DMCRB)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, see Table 16.5 to Table 16.13.

DMA Source Buffer Size Register (DMSBS)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, see Table 16.8 to Table 16.13.

DMA Destination Buffer Size Register (DMDBS)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, see Table 16.8 to Table 16.13.

DMA Transfer Enable Bit (DMCNT.DTE)

Although the DMCNT.DTE bit enables or disables data transfer by the register write access, it is automatically cleared to 0 by the DMAC according to the DMA transfer state.

The conditions for clearing this bit by the DMAC are as follows:

- When the specified total volume of data transfer is completed
- When DMA transfer is stopped by the repeat size end interrupt
- When DMA transfer is stopped by the extended repeat area overflow interrupt
- When DMA transfer error occurs

Writing to the registers for the channels when the corresponding DMCNT.DTE bit is set to 1 is prohibited (except for DMCNT). In this case, writing must be performed after the bit is cleared to 0.

DMAC Active Flag (DMSTS.ACT)

The DMSTS.ACT flag indicates whether the DMACn is in the idle or active state.

This flag is set to 1 when the DMAC starts data transfer, and is cleared to 0 when data transfer in response to one transfer request is completed.

Even when DMA transfer is stopped by writing 0 to the DMCNT.DTE bit during DMA transfer, this flag remains 1 until DMA transfer is completed.

Transfer End Interrupt Flag (DMSTS.DTIF)

The DMSTS.DTIF flag is set to 1 after DMA transfer of the total transfer size of data is completed.

When both this flag and the DMINT.DTIE bit are set to 1, a transfer end interrupt is requested.

This flag is set to 1 when the DMA transfer bus cycle is completed and the DMSTS.ACT flag is cleared to 0 indicating the DMA transfer end.

This flag is automatically cleared to 0 when the DMCNT.DTE bit is set to 1 during the interrupt handling.

Transfer Escape End Interrupt Flag (DMSTS.ESIF)

The DMSTS.ESIF flag is set to 1 when a repeat size end interrupt or extended repeat area overflow interrupt is requested. When this bit and the DMINT.ESIE bit are set to 1, a transfer escape end interrupt is requested.

This flag is set to 1 when the bus cycle of the DMA transfer having caused the interrupt request is completed and the DMSTS.ACT flag is cleared to 0 indicating the DMA transfer end.

This flag is automatically cleared to 0 when the DMCNT.DTE bit is set to 1 during an interrupt handling.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set.

For details, see section 13, Interrupt Controller Unit (ICU).

16.3.12 Channel Priority

When multiple DMA transfer requests are present, the DMAC determines the priority of channels that have DMA transfer requests.

- The channel priority is fixed as follows: Channel 0 > Channel 1 > Channel 2 > Channel 3… > Channel 7 (Channel 0: Highest).

When a DMA transfer request is generated during data transfer, channel arbitration is started after the final data has been transferred, and DMA transfer of the higher-priority channel starts.

16.3.13 Channel Security

The secure attribute can be set with CPSCU.DMACCHSAR.SADMACn for each DMAC channel.

When the CPSCU.DMACCHSAR.SADMACn bit is 0.

- When the corresponding channel transfers, it behaves as a secure master.
- The register of corresponding channel has a secure attribute.

When the CPSCU.DMACCHSAR.SADMACn bit is 1.

- When the corresponding channel transfers, it behaves as a non-secure master.
- The register of corresponding channel has a non-secure attribute.

Refer to the Security chapter and BUS chapter for areas accessible to secure and non-secure masters.

Do not change to the CPSCU.DMACCHSAR while DMA transfer.

16.3.14 Channel Privilege

The privileged attribute can be set with CPSCU.DMACCHPAR.DMACCHPARn for each DMAC channel n.

When the CPSCU.DMACCHPAR.DMACCHPARn bit is 0.

- When the corresponding channel transfers, it behaves as a privileged master.
- The registers of channel are protected from a unprivileged access.

When the CPSCU.DMACCHPAR.DMACCHPARn bit is 1.

- The transfer of DMAC channel is unprivileged access for both read and write.
- The registers of channel are unprivileged attributes.

Do not change to the CPSCU.DMACCHPAR while DMA transfer.

16.4 Ending DMA Transfer

The operation for ending DMA transfer depends on the transfer end conditions. When DMA transfer ends, the DMCNT.DTE bit and the DMSTS.ACT flag are changed from 1 to 0, indicating that DMA transfer has ended.

16.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations

(1) In Normal Transfer Mode (DMTMD.MD[1:0] = 00b)

When the value of DMCRAL changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is cleared to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

(2) In Repeat Transfer Mode (DMTMD.MD[1:0] = 01b)

When the value of DMCRBL changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is cleared to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.

If the DMTMD.TKP bit is 1 (in free-running function), the DMSTS.DTIF bit is set to 1, but the DMCNT.DTE bit is not cleared to 0.

(3) In Block Transfer Mode (DMTMD.MD[1:0] = 10b)

When the value of DMCRBL changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is cleared to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set.

For details, see section 13, Interrupt Controller Unit (ICU).

If the DMTMD.TKP bit is 1 (in free-running function), the DMSTS.DTIF bit is set to 1, but the DMCNT.DTE bit is not cleared to 0.

(4) In Repeat-Block Transfer Mode (DMTMD.MD[1:0] = 11b)

When the value of DMCRBL changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is cleared to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 13, Interrupt Controller Unit (ICU).

If the DMTMD.TKP bit is 1 (in free-running function), the DMSTS.DTIF bit is set to 1, but the DMCNT.DTE bit is not cleared to 0.

16.4.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, a repeat size end interrupt is requested when transfer of a 1-repeat size of data is completed while the DMINT.RPTIE bit is set to 1. When the interrupt is requested to complete DMA transfer, the DMCNT.DTE bit is cleared to 0 and the DMSTS.ESIF flag is set to 1 even if the DMTMD.TKP bit is 1 (in free-running function). If the DMINT.ESIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC. Here, the transfer can be resumed by writing 1 to the DMCNT.DTE bit.

A repeat size end interrupt can be requested also in block transfer mode. In block transfer mode, the interrupt is requested in the same way as in repeat transfer mode when transfer of a 1-block size data is completed.

Repeat size end interrupt cannot be requested in repeat-block transfer mode.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 13, Interrupt Controller Unit (ICU).

16.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area occurs while the extended repeat area is specified and the DMINT.SARIE or DMINT.DARIE bit is set to 1 even if the DMTMD.TKP bit is 1 (in free-running function), an interrupt by an extended repeat area overflow is requested. When the interrupt is requested, the DMA transfer is terminated, the DMCNT.DTE bit is cleared to 0, and the ESIF flag in DMSTS is set to 1. If the DMINT.ESIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.

Even if an interrupt by an extended repeat area overflow is requested during a read cycle, the following write cycle is performed.

In block transfer mode, even if an interrupt by an extended repeat area overflow is requested during a 1-block transfer, the remaining data in the block is transferred; transfer is terminated after a block transfer.

An interrupt by an extended repeat area overflow cannot be requested in repeat-block transfer mode.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 13, Interrupt Controller Unit (ICU).

16.5 Processing on DMA Transfer Error

DMA transfer error occurs with MSAU error, the Slave TrustZone Filter error, the Master MPU error, the Slave Bus Error or the Illegal Access Error. If the access error occurs during the DMA transfer, the DMAC immediately stops the transfer of error occurred channel. At this time, the ICU setting of the corresponding channel is also cleared. If there is a request other than the channel which caused the error, it will be re-arbitration as it is.

When the transfer error occurs, DMCNT.DTE of the error causing channel is set to 0. Also, the error response is informed to the DELSRn of the corresponding channel is cleared. Write back to each register is not performed. The information of the channel that caused the error is set in DMECHR.

16.6 Interrupts

16.6.1 Transfer End Interrupt

Each DMAC channel can output an interrupt request (DMAC0n_INT) to the CPU or the DTC after transfer in response to one request is completed.

When the transfer destination is the external bus, an interrupt request is generated upon completion of data write to the write buffer not to the actual transfer destination.

In repeat-block transfer mode, do not enable escape transfer end interrupt.

Table 16.20 lists the relation among the interrupt sources, the interrupt status flags, and the interrupt enable bits. Figure 16.23 shows the schematic logic diagram of interrupt outputs (DMACn (n = 0 to 7)). Figure 16.24 shows the DMAC interrupt handling routine to resume/terminate DMA transfer.

Table 16.20 Relation among interrupt sources, interrupt status flags, and interrupt enable bits

Interrupt sources		Interrupt enable bits	Interrupt status flags	Request output enable bits
Transfer end		—	DMSTS.DTIF	DMINT.DTIE
Escape transfer end	Repeat size end	DMINT.RPTIE	DMSTS.ESIF	DMINT.ESIE
	Source address extended repeat area overflow	DMINT.SARIE		
	Destination address extended repeat area overflow	DMINT.DARIE		

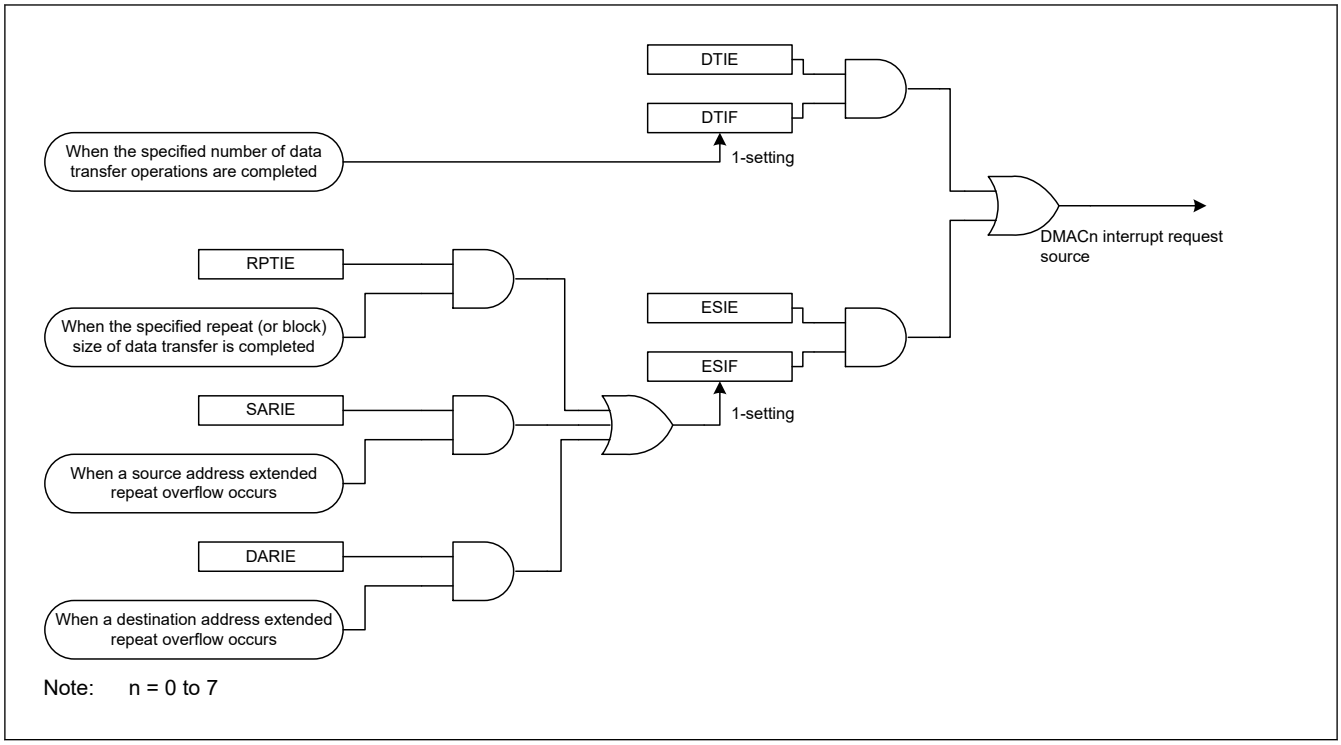


Figure 16.23 Schematic logic diagram of interrupt output source (DMACn)

Specifically, the different procedures are used for canceling an interrupt to restart DMA transfer in the following two cases:

- When terminating a DMA transfer
- When continuing a DMA transfer

16.6.1.1 When Discontinuing or Terminating a DMA Transfer

Write 0 to the DMSTS.DTIF flag to clear a transfer end interrupt, and to the DMSTS.ESIF flag to clear a repeat size interrupt and an extended repeat area overflow interrupt. The corresponding DMACn channel remains in the stop state. When starting another DMA transfer after that, set the appropriate registers, and set the DMCNT.DTE bit to 1 (DMA transfer enabled).

16.6.1.2 When Continuing a DMA Transfer

Write 1 to the DMCNT.DTE bit. The DMSTS.ESIF flag is automatically cleared to 0 (interrupt source cleared), and DMA transfer is resumed.

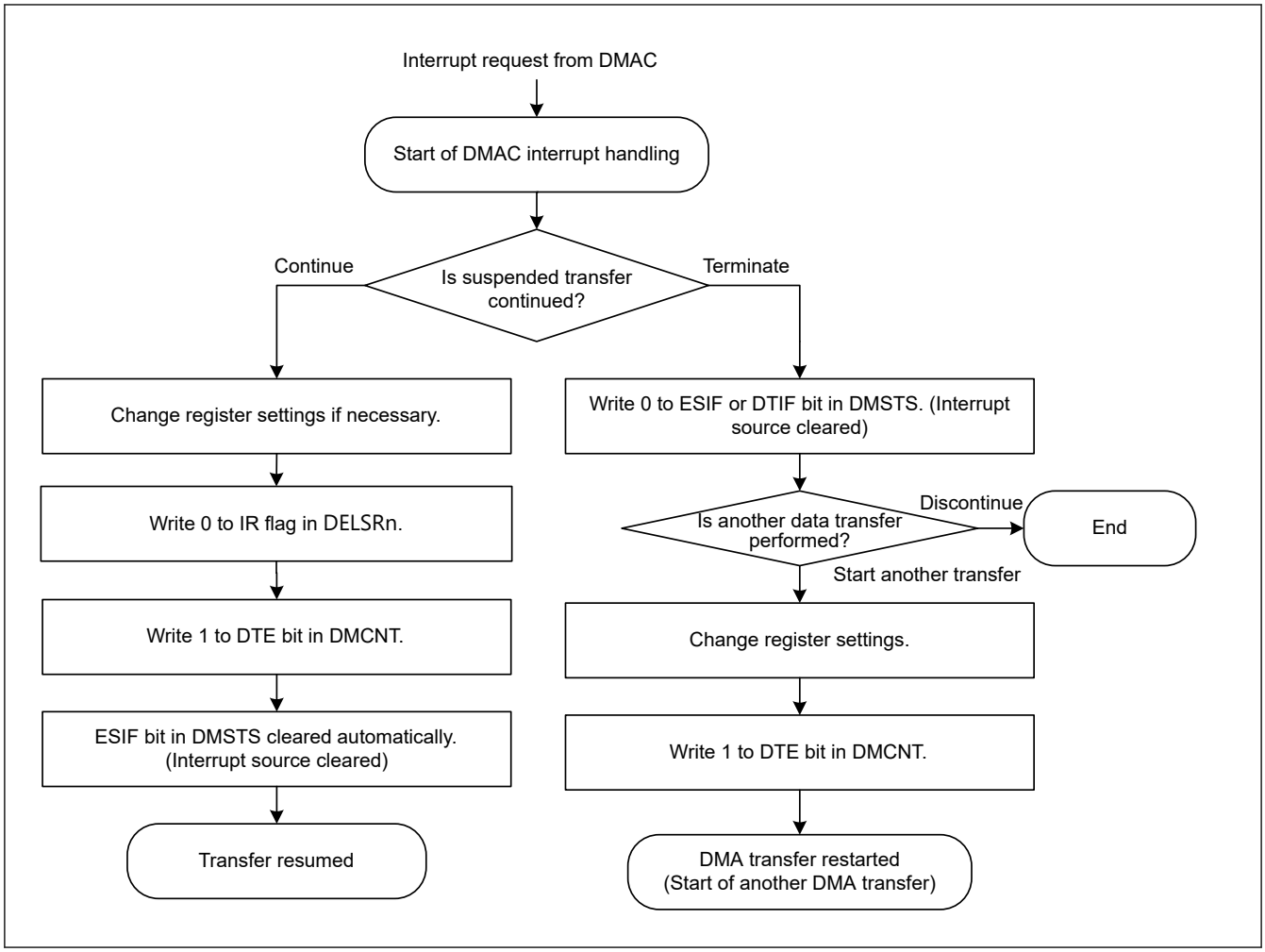


Figure 16.24 DMAC Interrupt Handling Routine to Resume/Terminate DMA Transfer

16.6.2 Transfer Error Interrupt

Error response detection interrupt request (DMA0_TRANSERR) is generated from the DMAC/DTC when the transfer error is detected during DMAC transfer.

The types of interrupts that occur when a DMAC transfer error occurs are listed in the Table 16.21. The Table 16.21 also shows error information stored when a transfer error occurs.

Table 16.21 Interrupt and error information due to DMAC transfer error cause

Transfer error factor	NMI/RESET*1 Request	Interrupt Request	Bus Error Status	Error Address Error R/W	Error Channel Information
MSAU Error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. MSERRSTAT	BUS.BMSA4ERRADD BUS.BMSA4ERRRW	DMAC. DMECHR
Master MPU Error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. MMERRSTAT	BUS.BUS4ERRADD BUS.BUS4ERRRW	DMAC. DMECHR
Illegal Access Error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. ILERRSTAT	BUS.BUS4ERRADD BUS.BUS4ERRRW	DMAC. DMECHR
Slave Bus Error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. SLERRSTAT	BUS.BUS4ERRADD BUS.BUS4ERRRW	DMAC. DMECHR
TrustZone Filter error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. STERRSTAT	BUS.BUS4ERRADD BUS.BUS4ERRRW	DMAC. DMECHR

Note 1. If ICU.NMIER.BUSEN is enabled with DMA0_TRANSERR set in ICU.IELSR, NMI and an interrupt will occur due to a transfer error caused by DMA. Only NMI can be generated by not setting DMA_TRANSERR to ICU.IELSR.

16.7 Event Link

Each DMAC channel outputs an event link request signal (DMAC0n_INT) every time it completes a data transfer, or a block transfer in block transfer mode.

When the transfer destination is the external bus, an event link request signal is generated when the writing to the write buffer is accepted.

For details, see section 18, Event Link Controller (ELC).

16.8 Low-Power Consumption Function

Before entering the module-stop state, Software Standby mode or Deep Software Standby mode, you must first set the DMAST.DMST bit to 0 (the DMAC module suspended) and use the settings in the sections that follow.

(1) Module-stop function

Writing 1 to the MSTPCRA.MSTPA22 bit enables the module-stop function of the DMAC. If a DMA transfer is in progress when 1 is written to the MSTPA22 bit, the transition to the module-stop state proceeds after the DMA transfer ends. Access to the DMAC registers is prohibited while the MSTPA22 bit is 1. Writing 0 to the MSTPA22 bit releases the DMAC from the module-stop state.

(2) Software Standby mode and Deep Software Standby mode

Use the settings described in section 10.7.3.1. Transition to Software Standby Mode, or in section 10.7.4.1. Transition to Deep Software Standby Mode .

If DMA transfer operations are in progress when the WFI instruction is executed, the DMA transfer completes before the transition to Software Standby mode or Deep Software Standby mode.

(3) Notes on low power consumption function

For information on the WFI instruction and register settings, see section 10.8.7. Timing of WFI Instructions.

To perform a DMA transfer after returning from a low power mode, set the DMAST.DMST bit to 1 again. To use a request that is generated in Software Standby mode as an interrupt request to the CPU but not as a DMAC startup request, specify the CPU as the interrupt request destination, as described in section 13.5.5. Selecting Interrupt Request Destinations, and then execute the WFI instruction.

16.9 Usage Notes

16.9.1 DMA Transfer to External Devices

In a DMA transfer to an external device, the DMSTS.ACT flag may change to 0 (the DMAC suspended) before the external bus access ends after the final data write is started.

16.9.2 Access to the Registers during DMA Transfer

Do not write to the following registers while the DMSTS.ACT flag of the same channel is set to 1 (DMAC active state) or the DMCNT.DTE bit of the same channel is set to 1 (DMA transfer enabled):

- DELSR
- DMSAR
- DMDAR
- DMCRA
- DMCRB
- DMTMD
- DMINT
- DMAMD
- DMOFR
- DMSBS

- DMDBS
- DMSRR
- DMDRR
- DMBWR
- ICUSARC
- DMACSAR

16.9.3 DMA Transfer to Reserved Areas

DMA transfer to the reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on the reserved areas, see section 4, Address Space.

16.9.4 Setting of DMAC Event Link Setting Register of the Interrupt Controller Unit (DELSRn)

The DMAC event link setting register (DELSRn) should be set while the DMA transfer enable bit (DMCNT.DTE) is cleared to 0 (DMA transfer is disabled). Moreover, the DTC activation enable register (ICU.IELSRn.DTCE (n = 0 to 95)) that corresponds to the same event number that has been set by the DELSRn register should not be set to 1. For details on the ICU.IELSRn.DTCE, see section 13, Interrupt Controller Unit (ICU).

16.9.5 Suspending or Restarting DMAC Activation

To suspend a DMAC activation request, write 0x00 to the DMAC Event Link select bits (DELSRn.DELS[8:0]). To restart the DMA transfer, write the event number to the DELSRn.DELS[8:0] bits following the settings shown in section 16.3.9. Activating the DMAC.

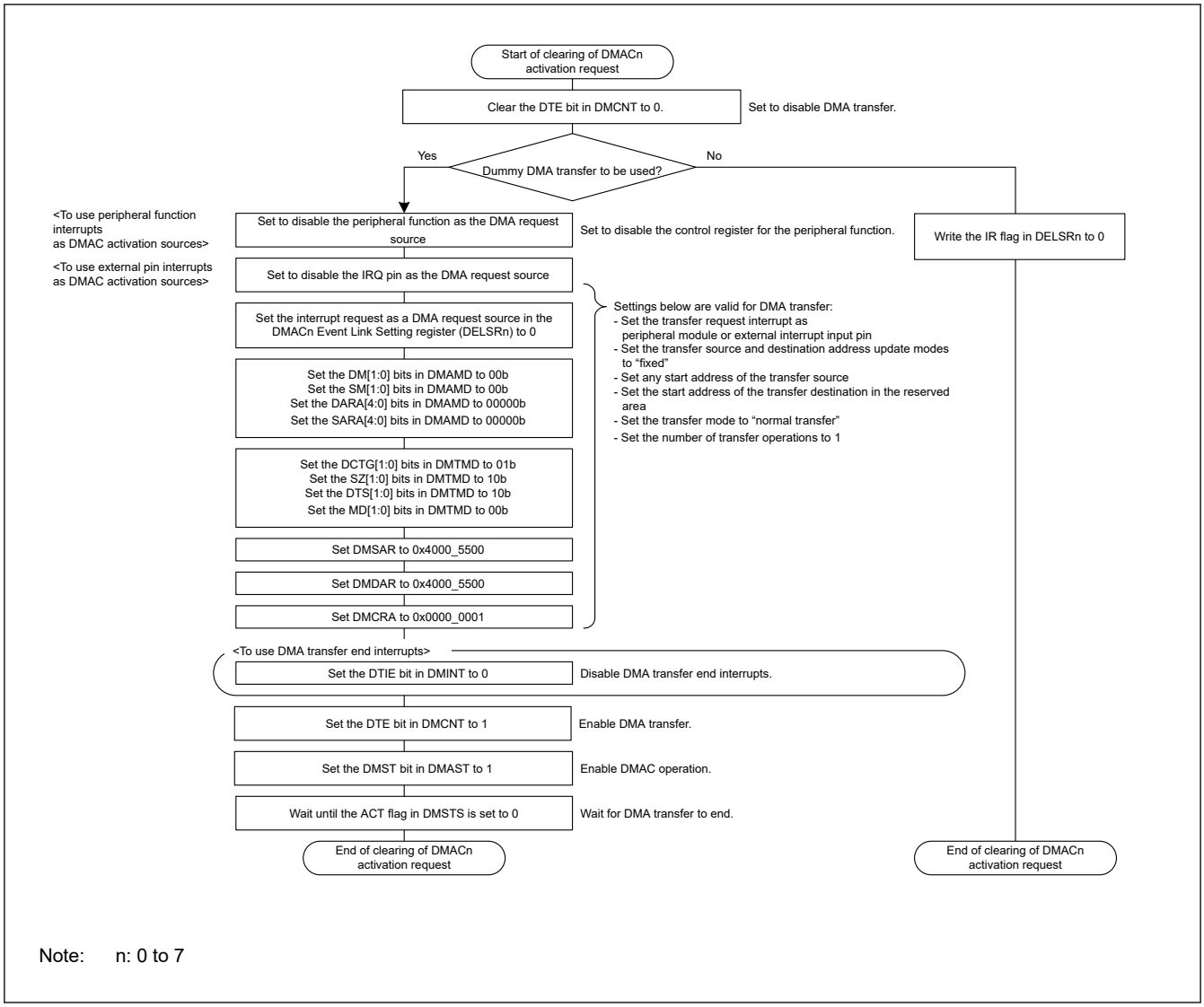
16.9.6 Precautions for Resuming DMA Transfer

A DMAC activation request might occur in the next request after a DMA transfer completes. If this happens, the DMA transfer starts and the DMAC activation request is held in the DMAC. To prevent this, stop the DMAC activation requests by setting the DELSRn.DELS[8:0] bits to 0.

When a DMAC activation request occurs after the last round of the DMA transfer is generated, clear the DMAC activation request with either of the following approaches.

- Clear the DMAC activation request with a DMA dummy transfer.
- Set the DMCNT.DTE bit to 0 and then set the DELSRn.IR flag to 0.

See Figure 16.25.



Note: n: 0 to 7

Figure 16.25 Example of register setting procedure to clear the DMAC activation interrupt

16.9.7 Restriction for OSPI

There is a restriction to using OSPI, see section 37.4.5. Restriction in 8D-8D-8D profile 1.0 format.

17. Data Transfer Controller (DTC)

17.1 Overview

A Data Transfer Controller (DTC) module is provided for transferring data when activated by an interrupt request.

Table 17.1 lists the DTC specifications and Figure 17.1 shows DTC block diagram.

Table 17.1 DTC specifications

Parameter	Description
Transfer modes	• Normal transfer mode A single activation leads to a single data transfer. • Repeat transfer mode A single activation leads to a single data transfer. The transfer address returns to the start address after the number of data transfers reaches the specified repeat size. The maximum number of repeat transfers is 256 and the maximum data transfer size is 256 × 32 bits (1024 bytes) • Block transfer mode A single activation leads to a transfer of a single block. The maximum block size is 256 × 32 bits = 1024 bytes.
Transfer channel	• Channel transfer can be associated with the interrupt source (transferred by a DTC activation request from the ICU) • Multiple data units can be transferred on a single activation source (chain transfer) • Chain transfers are selectable to either execute when the counter is 0, or always execute.
Transfer space	• 4 GB area from 0x0000_0000 to 0xFFFF_FFFF, excluding reserved areas
Data transfer units	• Single data unit: 1 byte (8 bits), 1 halfword (16 bits), 1 word (32 bits) • Single block size: 1 to 256 data units.
CPU interrupt source	• An interrupt request can be generated to the CPU on a DTC activation interrupt • An interrupt request can be generated to the CPU after a single data transfer • An interrupt request can be generated to the CPU after a data transfer of a specified volume.
Processing on DTC transfer error	• When the DTC transfer error occurs, it stops the transfer that caused the error • Request to clear the register for activation request of DTC error number to ICU
Error response detection interrupt	Generated when the DTC transfer error occurs
Event link function	An event link request is generated after one data transfer (for block, after one block transfer)
Read skip	Read of transfer information can be skipped
Write-back skip	When the transfer source or destination address is specified as fixed, a write-back of transfer information can be skipped
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security and Privilege attribution can be set for each activation source

Note: Security attribution Register of DTC is described in ICU.ICUSARG, ICU.ICUSARH and ICU.ICUSARI

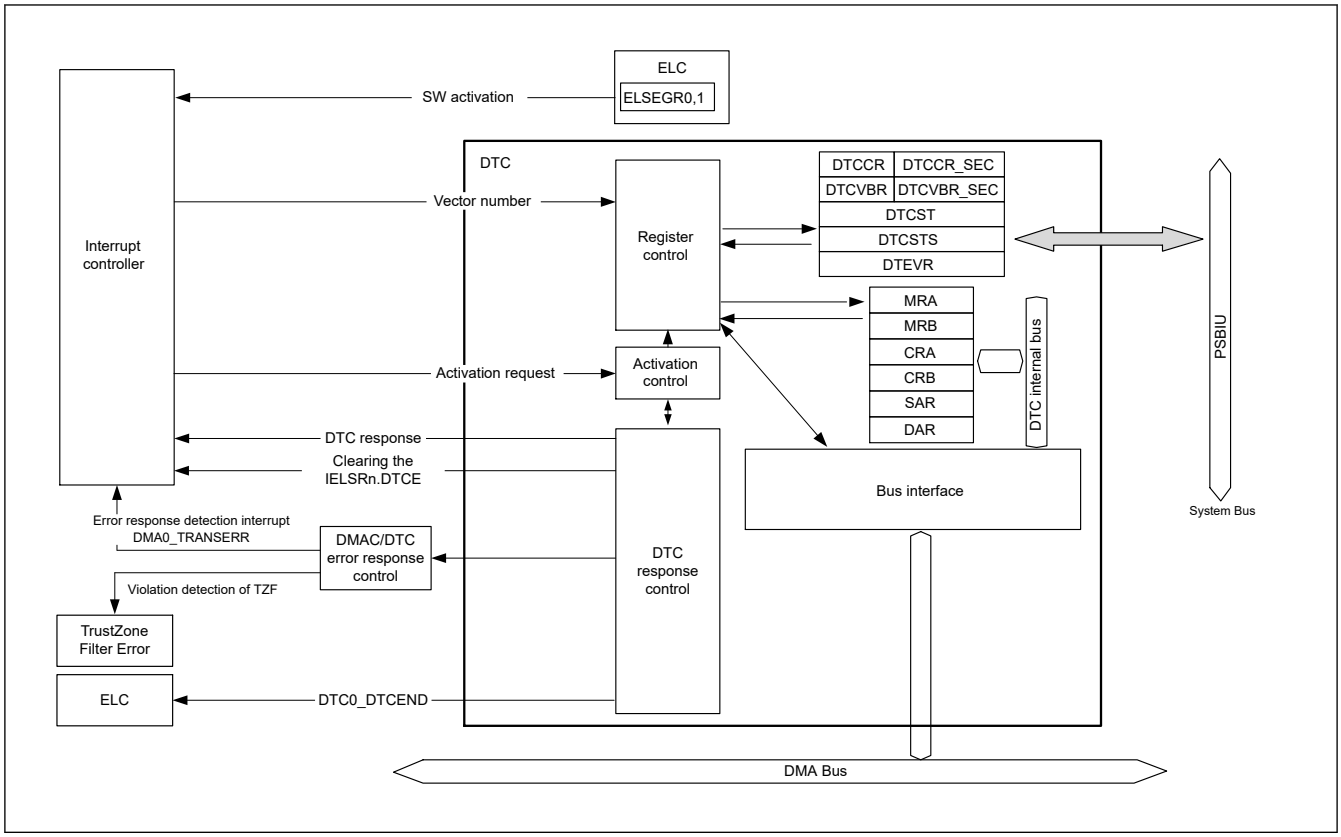


Figure 17.1 DTC block diagram

See section 13.1. Overview in section 13, Interrupt Controller Unit (ICU) for the connections between the DTC and NVIC in the CPU.

17.2 Register Descriptions

MRA, MRB, SAR, DAR, CRA, and CRB are all DTC internal registers that cannot be directly accessed from the CPU. Values to be set in these DTC internal registers are placed in the SRAM area as transfer information. When an activation request is generated, the DTC reads the transfer information from the SRAM area and sets it in its internal registers. After the data transfer ends, the internal register contents are written back to the SRAM area as transfer information.

17.2.1 DTCSAR : DTC Controller Security Attribution Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DTCSTSA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DTCSTSA	DTC Security Attribution 0: Secure. 1: Non-secure.	R/W

Bit	Symbol	Function	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

This register only sets the DTCST security attribute.

DTCSTSA bit (DTC Security Attribution)

Security attributes of registers for DTCST and DTCSTS.

Do not write to the DTCSTSA bit while DTC transfer is enabled or a bus master is writing to the DTC registers.

17.2.2 MRA : DTC Mode Register A

Base address: DTCVBR_SEC
DTCVBR

Offset address: 0x03 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 17.3.1. Allocating Transfer Information and DTC Vector Table.)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MD[1:0]		SZ[1:0]		SM[1:0]		—	—
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
1:0	—	The read values are undefined. The write value should be 0.	—
3:2	SM[1:0]	Transfer Source Address Addressing Mode 0 0: Address in the SAR register is fixed (write-back to SAR is skipped.) 0 1: Address in the SAR register is fixed (write-back to SAR is skipped.) 1 0: SAR value is incremented after data transfer: +1 when SZ[1:0] = 00b +2 when SZ[1:0] = 01b +4 when SZ[1:0] = 10b 1 1: SAR value is decremented after data transfer: -1 when SZ[1:0] = 00b -2 when SZ[1:0] = 01b -4 when SZ[1:0] = 10b	—
5:4	SZ[1:0]	DTC Data Transfer Size 0 0: Byte (8-bit) transfer 0 1: Halfword (16-bit) transfer 1 0: Word (32-bit) transfer 1 1: Setting prohibited	—
7:6	MD[1:0]	DTC Transfer Mode Select 0 0: Normal transfer mode 0 1: Repeat transfer mode 1 0: Block transfer mode 1 1: Setting prohibited	—

The MRA register cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 0x03) and DTC transfers it automatically to and from the MRA register. See section 17.3.1. Allocating Transfer Information and DTC Vector Table.

17.2.3 MRB : DTC Mode Register B

Base address: DTCVBR_SEC
DTCVBR

Offset address: 0x02 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 17.3.1. Allocating Transfer Information and DTC Vector Table.)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CHNE	CHNS	DISEL	DTS	DM[1:0]		—	—
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
1:0	—	The read values are undefined. The write value should be 0.	—
3:2	DM[1:0]	Transfer Destination Address Addressing Mode 0 0: Address in the DAR register is fixed (write-back to DAR is skipped) 0 1: Address in the DAR register is fixed (write-back to DAR is skipped) 1 0: DAR value is incremented after data transfer: +1 when MRA.SZ[1:0] = 00b +2 when SZ[1:0] = 01b +4 when SZ[1:0] = 10b 1 1: DAR value is decremented after data transfer: -1 when MRA.SZ[1:0] = 00b -2 when SZ[1:0] = 01b -4 when SZ[1:0] = 10b	—
4	DTS	DTC Transfer Mode Select 0: Select transfer destination as repeat or block area. 1: Select transfer source as repeat or block area.	—
5	DISEL	DTC Interrupt Select 0: Generate an interrupt request to the CPU when specified data transfer is complete. 1: Generate an interrupt request to the CPU each time DTC data transfer is performed.	—
6	CHNS	DTC Chain Transfer Select 0: Chain transfer is continuous. 1: Chain transfer occurs only when the transfer counter changes from 1 to 0 or 1 to CRAH.	—
7	CHNE	DTC Chain Transfer Enable 0: Chain transfer is disabled. 1: Chain transfer is enabled.	—

The MRB register cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 0x02) and DTC transfers it automatically to and from the MRB register. See section 17.3.1. Allocating Transfer Information and DTC Vector Table.

DM[1:0] bits (Transfer Destination Address Addressing Mode)

The DM[1:0] bits are to fix the address of the DAR register or specify increment / decrement of the DAR register after transfer.

DTS bit (DTC Transfer Mode Select)

The DTS bit specifies whether the transfer source or destination is the repeat or block area in repeat or block transfer mode.

DISEL bit (DTC Interrupt Select)

The DISEL bit specifies the condition for generating an interrupt request to the CPU.

CHNS bit (DTC Chain Transfer Select)

The CHNS bit selects the chain transfer condition. When CHNE is 0, the CHNS setting is ignored. For details on the conditions for chain transfer, see Table 17.3.

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the activation source flag is not cleared, and an interrupt request to the CPU is not generated.

CHNE bit (DTC Chain Transfer Enable)

The CHNE bit enables chain transfer. The chain transfer condition is selected by the CHNS bit. For details on chain transfer, see section 17.4.6. Chain Transfer. When the chain transfer settings are CHNE = 1 and CHNS = 0, the transfer count according to the transfer settings will be -1 for each transfer request. Therefore, the transfer count for each transfer information must match.

17.2.4 SAR : DTC Transfer Source Register

Base address: DTCVBR_SEC
DTCVBR

Offset address: 0x04 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 17.3.1. Allocating Transfer Information and DTC Vector Table.)

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

The SAR sets the transfer source start address and cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x04) and DTC transfers it automatically to and from the SAR register. See section 17.3.1. Allocating Transfer Information and DTC Vector Table.

Misalignment is prohibited for DTC transfers. Bit[0] must be 0 when MRA.SZ[1:0] = 01b, and bit[1] and bit[0] must be 0 when MRA.SZ[1:0] = 10b.

17.2.5 DAR : DTC Transfer Destination Register

Base address: DTCVBR_SEC
DTCVBR

Offset address: 0x08 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 17.3.1. Allocating Transfer Information and DTC Vector Table.)

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

The DAR sets the transfer destination start address and cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x08) and DTC transfers it automatically to and from the DAR register. See section 17.3.1. Allocating Transfer Information and DTC Vector Table.

Misalignment is prohibited for DTC transfers. Bit[0] must be 0 when MRA.SZ[1:0] = 01b, and bit[1] and bit[0] must be 0 when MRA.SZ[1:0] = 10b.

17.2.6 CRA : DTC Transfer Count Register A

Base address: DTCVBR_SEC
DTCVBR

Offset address: 0x0E + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 17.3.1. Allocating Transfer Information and DTC Vector Table.)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	CRAL	Transfer Counter A Lower Register Specify the transfer count.	—
15:8	CRAH	Transfer Counter A Upper Register Specify the transfer count.	—

Note: The function depends on the transfer mode.
Note: Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

The CRA register consists of 16 bits. CRAL is the lower 8 bits and CRAH is the upper 8 bits. CRA is used in normal mode.

CRAL and CRAH are used in repeat transfer mode and block transfer mode.

The CRA register cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x0E) and DTC transfers it automatically to and from the CRA register. See section 17.3.1. Allocating Transfer Information and DTC Vector Table.

(1) Normal transfer mode (MRA.MD[1:0] = 00b)

In normal transfer mode, CRA functions as a 16-bit transfer counter. The transfer count is 1, 65535, and 65536 when the set value is 0x0001, 0xFFFF, and 0x0000, respectively. The CRA value is decremented (-1) on each data transfer.

(2) Repeat transfer mode (MRA.MD[1:0] = 01b)

In repeat transfer mode, the CRAH register holds the transfer count and the CRAL register functions as an 8-bit transfer counter. The transfer count is 1, 255, and 256 when the set value is 0x01, 0xFF, and 0x00, respectively. The CRAL value is decremented (-1) on each data transfer. When it reaches 0x00, the CRAH value is transferred to CRAL.

(3) Block transfer mode (MRA.MD[1:0] = 10b)

In block transfer mode, the CRAH register holds the block size and the CRAL register functions as an 8-bit block size counter. The transfer count is 1, 255, and 256 when the set value is 0x01, 0xFF, and 0x00, respectively. The CRAL value is decremented (-1) on each data transfer. When it reaches 0x00, the CRAH value is transferred to CRAL.

17.2.7 CRB : DTC Transfer Count Register B

Base address: DTCVBR_SEC
DTCVBR

Offset address: 0x0C + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 17.3.1. Allocating Transfer Information and DTC Vector Table.)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

The CRB sets the block transfer count for block transfer mode. The transfer count is 1, 65535, and 65536 when the set value is 0x0001, 0xFFFF, and 0x0000, respectively. The CRB value is decremented (-1) when the final data of a single block size is transferred. When normal transfer mode or repeat transfer mode is selected, this register is not used, and the set value is ignored.

The CRB cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x0C) and DTC transfers it automatically to and from the CRB register. See section 17.3.1. Allocating Transfer Information and DTC Vector Table.

17.2.8 DTCCR : DTC Control Register

Base address: DTC0_NS = 0x5000_AC00

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	RRS	—	—	—	—
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	—	This bit is read as 1. The write value should be 1.	R/W
4	RRS	DTC Transfer Information Read Skip Enable for Non-secure 0: Transfer information read is not skipped 1: Transfer information read is skipped when vector numbers match	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE7, P-TYPE2

RRS bit (DTC Transfer Information Read Skip Enable for Non-secure)

The RRS bit enables skipping of transfer information reads when vector numbers match. The DTC vector number is compared with the vector number in the previous activation process. When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transfer information. However, when the previous transfer is a chain transfer, the transfer information is read regardless of the RRS bit.

When the transfer counter (CRA register) becomes 0 during the previous normal transfer and when the transfer counter (CRB register) becomes 0 during the previous block transfer, the transfer information is read regardless of the RRS bit value.

17.2.9 DTCCR_SEC : DTC Control Register for Secure Region

Base address: DTC0 = 0x4000_AC00

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	RRSS	—	—	—	—
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	—	This bit is read as 1. The write value should be 1.	R/W
4	RRSS	DTC Transfer Information Read Skip Enable for Secure 0: Transfer information read is not skipped. 1: Transfer information read is skipped when vector numbers match.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE6, P-TYPE2

RRSS bit (DTC Transfer Information Read Skip Enable for Secure)

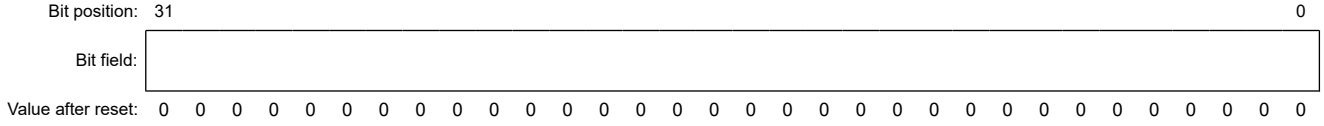
The RRSS bit enables skipping of transfer information reads when vector numbers match. The DTC vector number is compared with the vector number in the previous activation process. When these vector numbers match and the RRSS bit is set to 1, DTC data transfer is performed without reading the transfer information. However, when the previous transfer is a chain transfer, the transfer information is read regardless of the RRSS bit.

When the transfer counter (CRA register) becomes 0 during the previous normal transfer and when the transfer counter (CRB register) becomes 0 during the previous block transfer, the transfer information is read regardless of the RRSS bit value.

17.2.10 DTCVBR : DTC Vector Base Register

Base address: DTC0_NS = 0x5000_AC00

Offset address: 0x04



Bit	Symbol	Function	R/W
31:0	n/a	DTC Vector Base Address for Non-secure region Set the DTC vector base address. The lower 10 bits should be 0.	R/W

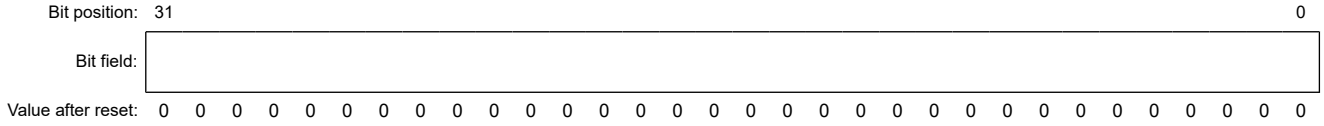
Note: S-TYPE7, P-TYPE2

The DTCVBR sets the base address for calculating the DTC vector table address, which can be set in the range of 0x0000_0000 to 0xFFFF_FFFF (4 GB) in 1 KB units.

17.2.11 DTCVBR_SEC : DTC Vector Base Register for Secure Region

Base address: DTC0 = 0x4000_AC00

Offset address: 0x14



Bit	Symbol	Function	R/W
31:0	n/a	DTC Vector Base Address for Secure region Set DTC Vector Base Address for Secure region. The lower 10 bits should be 0.	R/W

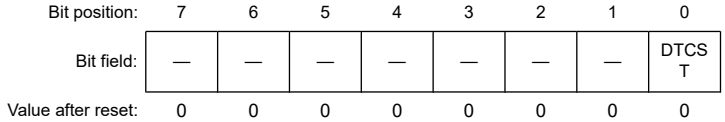
Note: S-TYPE6, P-TYPE2

The DTCVBR_SEC sets the base address for calculating the DTC vector table address, which can be set in the range of 0x0000_0000 to 0xFFFF_FFFF (4 GB) in 1-KB units.

17.2.12 DTCST : DTC Module Start Register

Base address: DTC0 = 0x4000_AC00
DTC0_NS = 0x5000_AC00

Offset address: 0x0C



Bit	Symbol	Function	R/W
0	DTCST	DTC Module Start 0: DTC module stopped 1: DTC module started	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE2

DTCST bit (DTC Module Start)

Set the DTCST bit to 1 to enable the DTC to accept transfer requests. When this bit is set to 0, transfer requests are no longer accepted. If this bit is set to 0 during a data transfer, the accepted transfer request is active until processing completes.

DTCST must be set to 0 before transitioning to one of the following state or mode:

- Module-stop state
- Software Standby mode
- Deep Software Standby mode

For details on these transitions, see section 17.10. Low Power Consumption Function and section 10, Low Power Modes.

17.2.13 DTCSTS : DTC Status Register

Base address: DTC0 = 0x4000_AC00
DTC0_NS = 0x5000_AC00

Offset address: 0x0E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ACT	—	—	—	—	—	—	—	VECN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	VECN[7:0]	DTC-Activating Vector Number Monitoring These bits indicate the vector number for the activation source when a DTC transfer is in progress. The value is only valid if a DTC transfer is in progress (ACT flag is 1).	R
14:8	—	These bits are read as 0.	R
15	ACT	DTC Active Flag 0: DTC transfer operation is not in progress. 1: DTC transfer operation is in progress.	R

Note: S-TYPE3, P-TYPE2

VECN[7:0] bits (DTC-Activating Vector Number Monitoring)

While transfer by the DTC is in progress, the VECN[7:0] bits indicate the vector number associated with the activation source for the transfer. The value read from the VECN[7:0] bits is valid if the ACT flag is 1, indicating a DTC transfer in progress, and invalid if the ACT flag is 0, indicating no DTC transfer is in progress.

ACT flag (DTC Active Flag)

The ACT flag indicates the state of the DTC transfer operation.

[Setting condition]

- When the DTC is activated by a transfer request.

[Clearing condition]

- When transfer by the DTC, in response to a transfer request, is complete.

17.2.14 DTEVR : DTC Error Vector Register

Base address: DTC0 = 0x4000_AC00
DTC0_NS = 0x5000_AC00

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DTESTA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	DTEVSAM	DTEV[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DTEV[7:0]	DTC Error Vector Number These bits represent error vector of the DTC.	R
8	DTEVSAM	DTC Error Vector Number SA Monitor Indicates the SA of vector number causing the error. 0: Secure vector number 1: Non-secure vector number	R
15:9	—	These bits are read as 0. The write value should be 0.	R
16	DTESTA	DTC Error Status Flag 0: No DTC transfer error occurred 1: DTC transfer error occurred	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R

Note: P-TYPE2 See below for access by security attribution.
Note: Writing to DTESTA depends on the value of DTEVSAM

When reading this register, it can be accessed from both Secure and Non-secure access.

When writing this register, it depends on DTEVR.DTESTA.

- When DMESA = 1, it can be accessed from Secure and Non-secure access.
- When DMESA = 0, it can be accessed from Secure. An error is returned when write access is performed in a Non-secure access.

This register is cleared by a reset caused by a transfer error. Please select interrupt (DMA0_TRANSERR) in BUS.OADCFG.OAD when you want to debug the program.

DTEV[7:0] bit (DTC Error Vector Number)

When a transfer error due to DTC transfer occurs, it stores the channel of DTC that was violated.

When reset was selected in MMPUOAD.OAD and MSAOAD.OAD, Since this register is also reset. Please select NMI when you want to debug the program.

[Setting condition]

- When the DTC transfer error occurs and DTESTA = 0.

[Clearing condition]

- When 1 is written to DTEVR.DTESTA.

DTEVSAM bit (DTC Error Vector Number SA Monitor)

When a transfer error due to DTC transfer occurs, it indicates the SA of the violating DTC vector number.

When reset was selected in MMPUOAD.OAD and MSAOAD.OAD, Since this register is also reset. Please select NMI when you want to debug the program.

[Setting condition]

- When the DTC transfer error occurs and DTESTA = 0.

[Clearing condition].

- When 1 is written to DTEVR.DTESTA.

DTESTA bit (DTC Error Status Flag)

Indicates whether or not a DTC transfer error occurred.

DTEV, DTEVSAM, DTESTA are cleared by writing 1 to DTESTA.

Writing 0 to DTESTA is ignored.

When reset was selected in MMPUOAD.OAD and MSAOAD.OAD, Since this register is also reset. Please select NMI when you want to debug the program.

[Setting condition]

- When the DTC transfer error occurs.

[Clearing condition]

- When 1 is written to DTEVR.DTESTA.

Note: When DTEVSAM = 1, it can be cleared in the Secure state and Non-secure state. DTEVSAM = 0, it cannot be cleared in the Non-secure state.

17.3 Activation Sources

The DTC is activated by an interrupt request. Setting the ICU.IELSRn.DTCE bit to 1 enables activation of the DTC by the associated interrupt. The selector output *n* number set in ICU.IELSRn is defined as the interrupt vector number, where n = 0 to 95. For an enabled interrupt, the specific DTC interrupt source associated with each interrupt vector number *n* is selected in ICU.IELSRn.IELS[8:0] where n = 0 to 95, as listed in section 13.3.2. Event Number in section 13, Interrupt Controller Unit (ICU). For activation by software, see section 18.2.2. ELSEGRn : Event Link Software Event Generation Register n (n = 0, 1).

The interrupt vector number is equivalent to the DTC vector table number. After the DTC accepted an activation request, it does not accept another activation request until the transfer for that single request is complete, regardless of the priority of the requests. When multiple activation requests are generated during a DTC transfer, the highest priority request is accepted on completion of the transfer. When multiple activation requests are generated while the DTC Module Start bit (DTCST.DTCST) is 0, the DTC accepts the highest priority request when DTCST.DTCST is subsequently set to 1. The smaller interrupt vector number has higher priority.

The DTC performs the following operations at the start of a single data transfer or for a chain transfer, after the last of the consecutive transfers:

- On completion of a specified round of data transfer, the ICU.IELSRn.DTCE bit is set to 0, and an interrupt request is sent to the CPU.
- If the MRB.DISEL bit is 1, an interrupt request is sent to the CPU on completion of a data transfer.
- For other transfers, the ICU.IELSRn.IR flag of the activation source is set to 0 at the start of the data transfer.

17.3.1 Allocating Transfer Information and DTC Vector Table

The DTC reads the start address of the transfer information associated with each activation source from the vector table and reads the transfer information starting at that address.

DTC has two vector tables, Non-secure side or Secure side. Because the interrupt vector number that serves as a trigger for DTC is divided into Non-secure or Secure. Place the vector table of the interrupt vector number of SA = 1 in DTCVBR which is the Non-secure side. Place the vector table of interrupt number SA = 0 in DTCVBR_SEC which is the Secure side.

The vector table must be located so that the lower 10 bits of the base address (start address) are 0. Use the DTC Vector Base Register (DTCVBR) to set the base address of the DTC vector table. Transfer information is allocated in the SRAM area. In the SRAM area, the start address of the transfer information *n* with vector number *n* must be 4n added to the base address in the vector table.

When setting the DTC transfer attribution for the privilege access or non-privileged access, allocating the vector table and transfer information as follows. The vector table should be located in the privileged area. Transfer information with the privileged attribution is located in the privileged area. Transfer information with the unprivileged attribution is located in the Unprivileged area. Privileged area is set by MMPU (see section 15, Memory Protection Unit (MPU)). Privileged and non-privileged access settings are located at bit 0 of the address corresponding to each vector number in the vector table.

Figure 17.2 shows the relationship between the DTC vector table and transfer information.

Figure 17.3 shows the bit arrangement in the vector table.

Figure 17.4 shows the allocation of transfer information in the SRAM area.

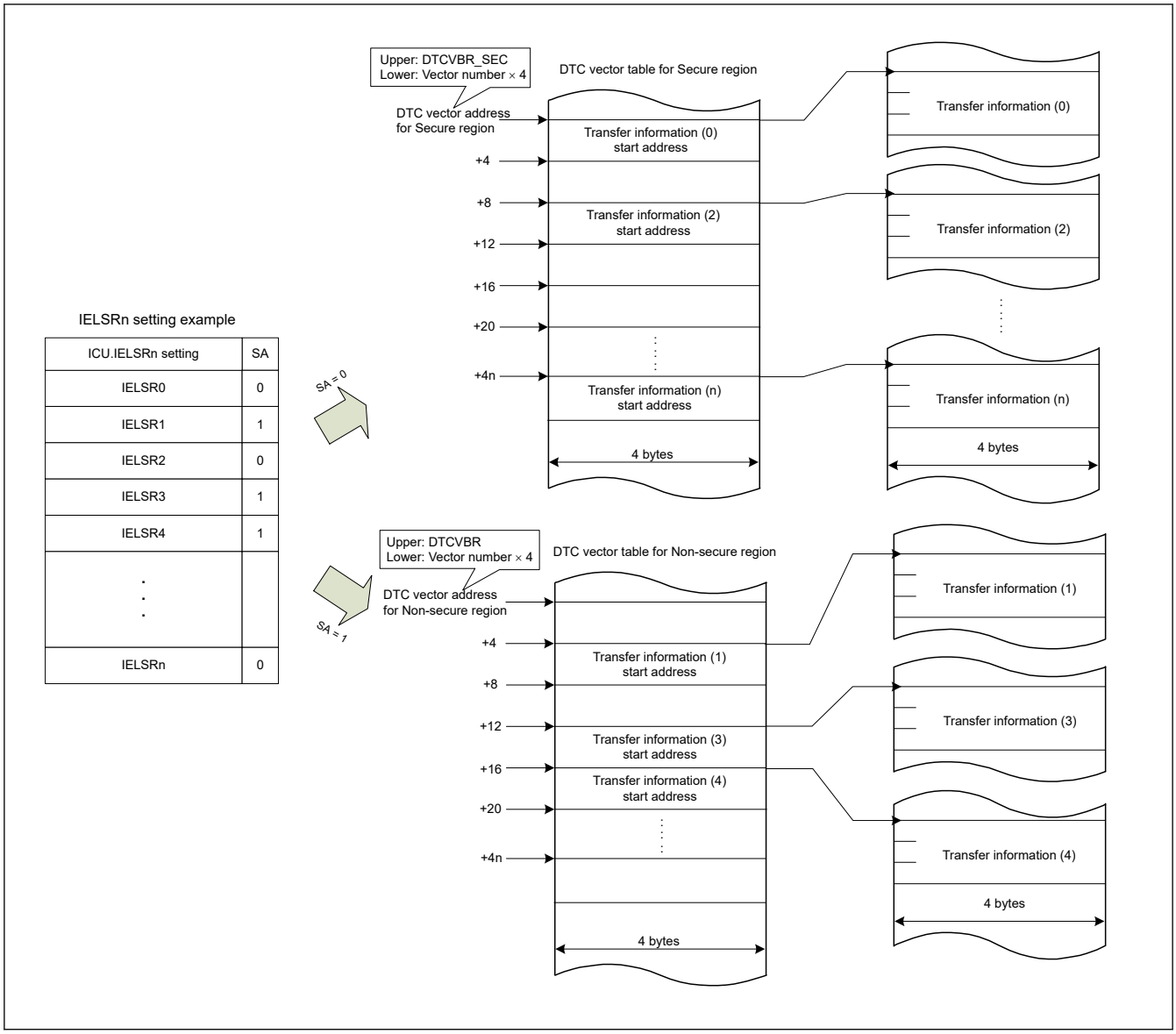


Figure 17.2 DTC vector table and transfer information

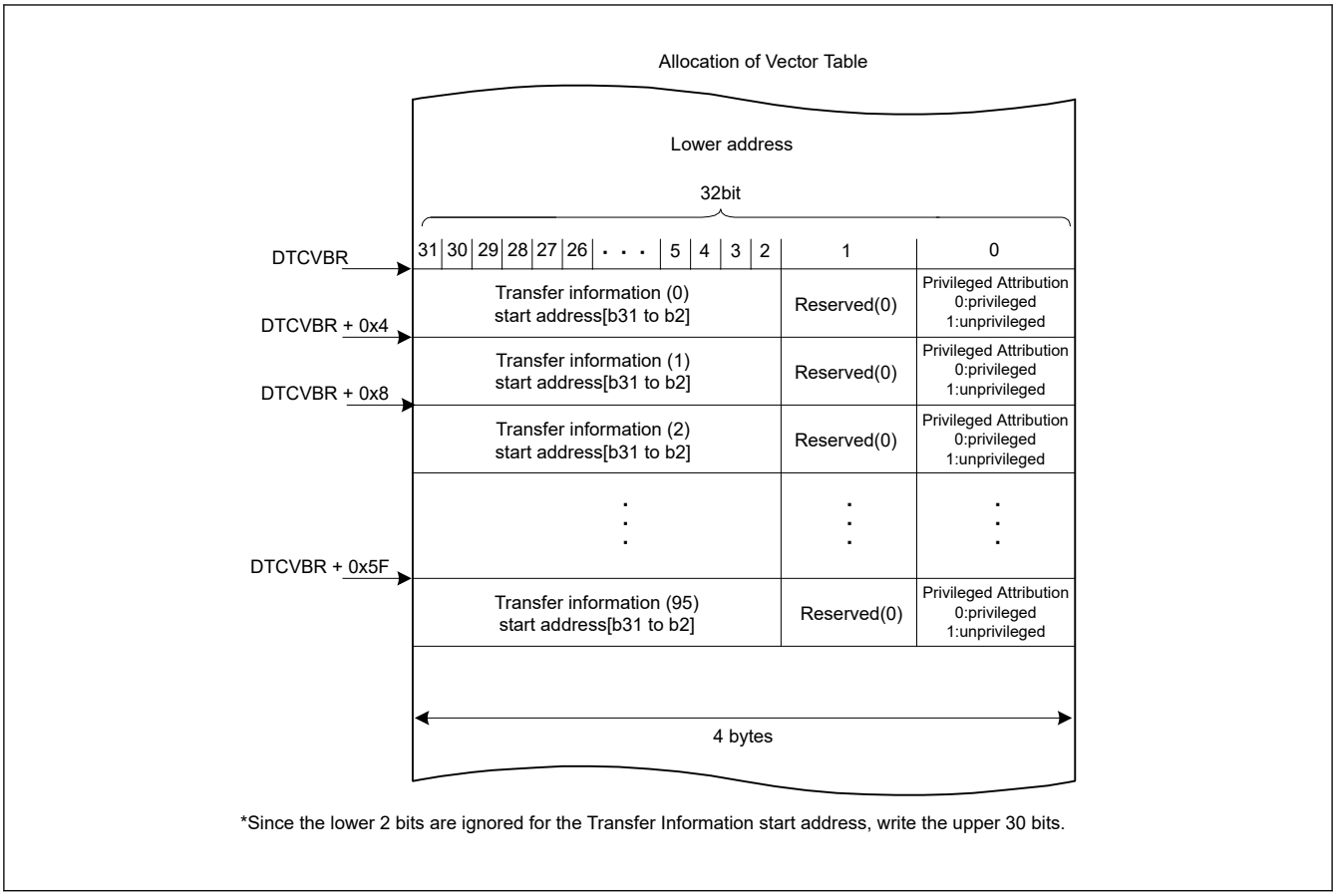


Figure 17.3 Bit arrangement in the vector table

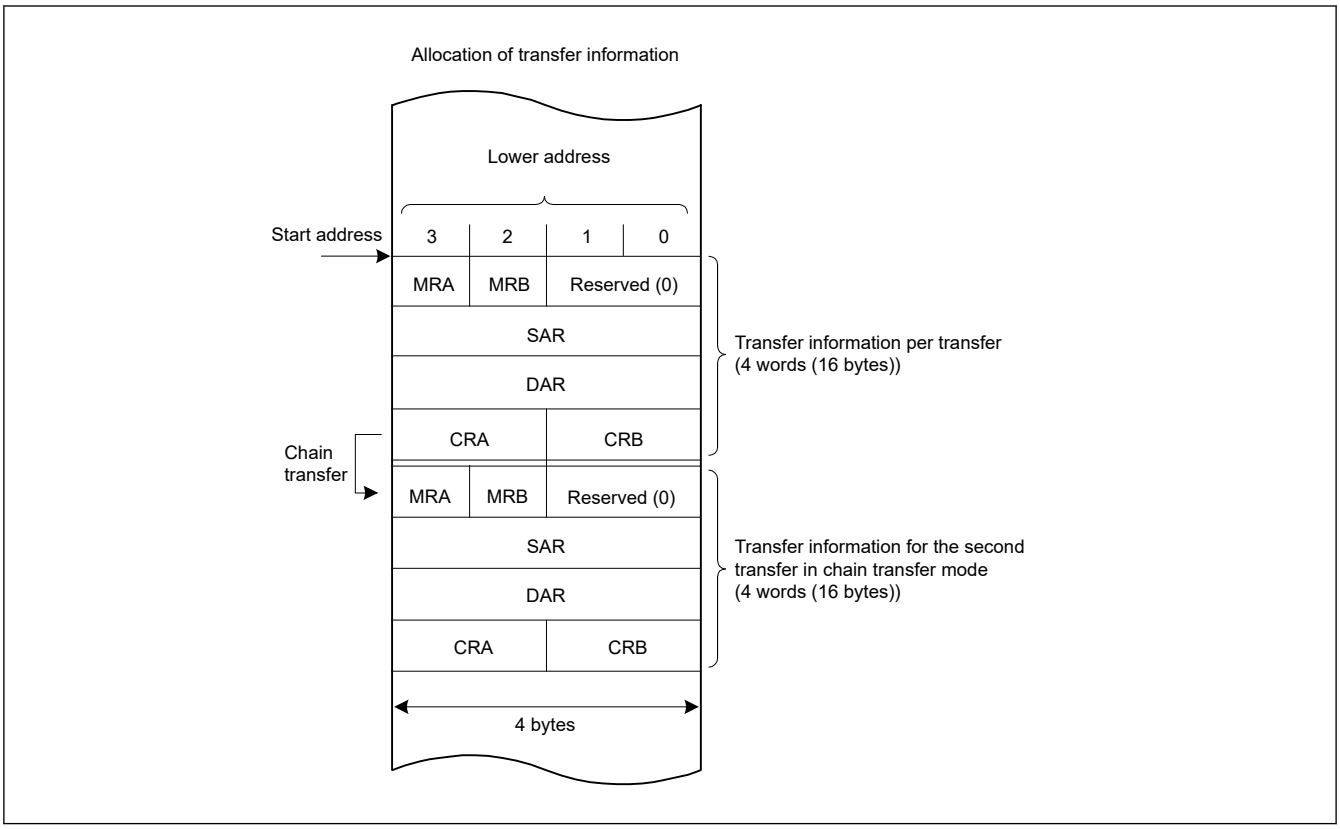


Figure 17.4 Allocation of transfer information in the SRAM area

17.4 Operation

The DTC transfers data according to the transfer information. Storage of the transfer information in the SRAM area is required before a DTC operation. When the DTC is activated, it reads the DTC vector associated with the vector number. The DTC reads the transfer information from the transfer information store address referenced by the DTC vector and transfers the data. After the data transfer, the DTC writes back the transfer information. Storing the transfer information in the SRAM area allows data transfer of any number of channels.

The transfer modes include:

- Normal transfer mode
- Repeat transfer mode
- Block transfer mode.

The DTC specifies a transfer source address in the SAR register and a transfer destination address in the DAR register. The values of these registers are incremented, decremented, or address-fixed independently after the data transfer.

Table 17.2 describes the DTC transfer modes.

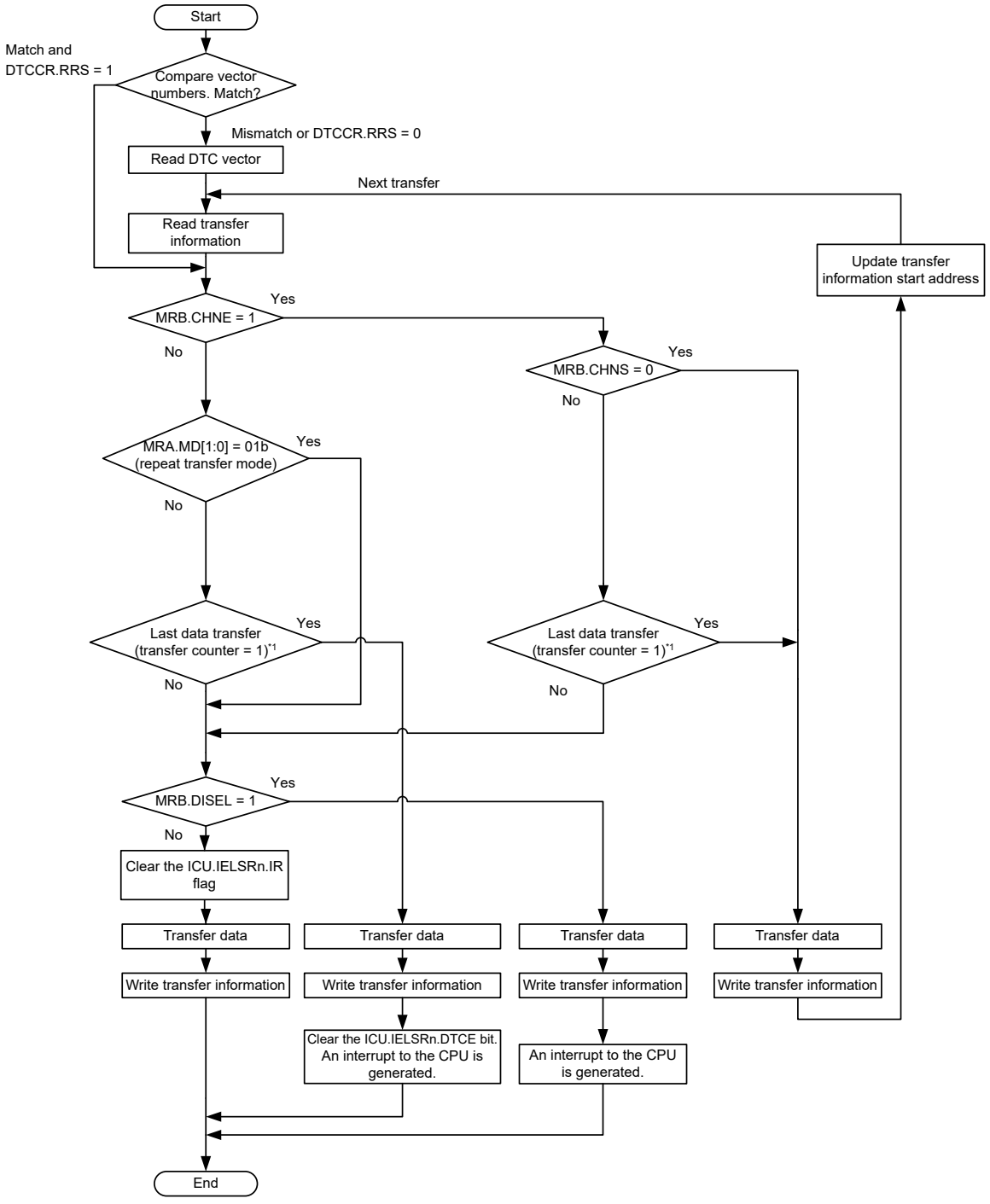
Table 17.2 DTC transfer modes

Transfer mode	Data size transferred on single transfer request	Increment or decrement of memory address	Settable transfer count
Normal transfer mode	1 byte (8 bit), 1 halfword (16 bit), 1 word (32 bit)	Incremented or decremented by 1, 2, or 4 or address-fixed	1 to 65536
Repeat transfer mode*1	1 byte (8 bit), 1 halfword (16 bit), 1 word (32 bit)	Incremented or decremented by 1, 2, or 4 or address-fixed	1 to 256*3
Block transfer mode*2	Block size specified in CRAH (1 to 256 bytes, 1 to 256 halfwords (2 to 512 bytes), or 1 to 256 words (4 to 1024 bytes))	Incremented or decremented by 1, 2, or 4 or address-fixed	1 to 65536

Note 1. Set the transfer source or transfer destination as the repeat area.
Note 2. Set the transfer source or transfer destination as the block area.
Note 3. After a data transfer of the specified count, the initial state is restored and operation restarts.

Setting the MRB.CHNE bit to 1 allows multiple transfers or chain transfer on a single activation source. It also enables a chain transfer when the specified data transfer is complete.

Figure 17.5 shows the operation flow of the DTC. Table 17.3 lists the chain transfer conditions. The combination of control information for the second and subsequent transfers are omitted in this table.



Note 1. Counter value before starting data transfer.

Figure 17.5 DTC operation flow

Table 17.3 Chain transfer conditions

First transfer				Second transfer*3				
CHNE bit	CHNS bit	DISEL bit	Transfer counter*1 *2	CHNE bit	CHNS bit	DISEL bit	Transfer counter*1 *2	DTC transfer
0	—	0	Other than (1 → 0)	—	—	—	—	Ends after the first transfer
0	—	0	(1 → 0)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU
0	—	1	—	—	—	—	—	
1	0	—	—	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	0	Other than (1 → *)	—	—	—	—	Ends after the first transfer
1	1	—	(1 → *)	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	1	Other than (1 → *)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU

Note 1. The transfer counter used depends on the transfer modes as follows:
Normal transfer mode — CRA register
Repeat transfer mode — CRAL register
Block transfer mode — CRB register

Note 2. On completion of a data transfer, the counters operate as follows:
1 → 0 in normal and block transfer modes
1 → CRAH in repeat transfer mode
(1 → *) in the table indicates both of these two operations, depending on the mode.

Note 3. Chain transfer can be selected for the second or subsequent transfers. The conditions for the combination of the second transfer and CHNE = 1 is omitted.

17.4.1 Transfer Information Read Skip Function

Reading of vector addresses and transfer information can be skipped by setting the DTCCR.RRS bit. When a DTC activation request is generated, the current DTC vector number is compared with the DTC vector number in the previous activation process. When these vector numbers match and the RRS bit is set to 1, the DTC data transfer is performed without reading the vector address and transfer information. However, when the previous transfer is a chain transfer, the vector address and transfer information are read. Additionally, when the transfer counter (CRA register) becomes 0 during the previous normal transfer, and when the transfer counter (CRB register) becomes 0 during the previous block transfer, transfer information is read regardless of the RRS bit. Figure 17.13 shows an example when reading the transfer information is skipped.

To update the vector table and transfer information, set the RRS bit to 0, update the vector table and transfer information, then set the RRS bit to 1. The stored vector number is discarded by setting the RRS bit to 0. The updated DTC vector table and transfer information are read in the next activation process.

17.4.2 Transfer Information Write-Back Skip Function

When the MRA.SM[1:0] bits or the MRB.DM[1:0] bits are set to address fixed, a part of the transfer information is not written back. Table 17.4 lists the transfer information write-back skip conditions and the associated registers. The CRA and CRB registers are written back, and the write-back of the MRA and MRB registers is skipped.

Table 17.4 Transfer information write-back skip conditions and applicable registers

MRA.SM[1:0] bits		MRB.DM[1:0] bits			
b3	b2	b3	b2	SAR register	DAR register
0	0	0	0	Skip	Skip
0	0	0	1		
0	1	0	0		
0	1	0	1		
0	0	1	0	Skip	Write-back
0	0	1	1		
0	1	1	0		
0	1	1	1		
1	0	0	0	Write-back	Skip
1	0	0	1		
1	1	0	0		
1	1	0	1		
1	0	1	0	Write-back	Write-back
1	0	1	1		
1	1	1	0		
1	1	1	1		

17.4.3 Normal Transfer Mode

The normal transfer mode allows a 1-byte (8 bit), 1-halfword (16 bit), 1-word (32 bit) data transfer on a single activation source. The transfer count can be set from 1 to 65536. Transfer source and destination addresses can be independently set to increment, decrement, or fixed. This mode enables an interrupt request to the CPU to be generated at the end of a specified-count transfer.

Table 17.5 lists register functions in normal transfer mode, and Figure 17.6 shows the memory map of normal transfer mode.

Table 17.5 Register functions in normal transfer mode

Register	Description	Value written back by writing transfer information
SAR	Transfer source address	Increment, decrement, or fixed*1
DAR	Transfer destination address	Increment, decrement, fixed*1
CRA	Transfer counter A	CRA - 1
CRB	Transfer counter B	Not updated

Note 1. Write-back operation is skipped in address-fixed mode.

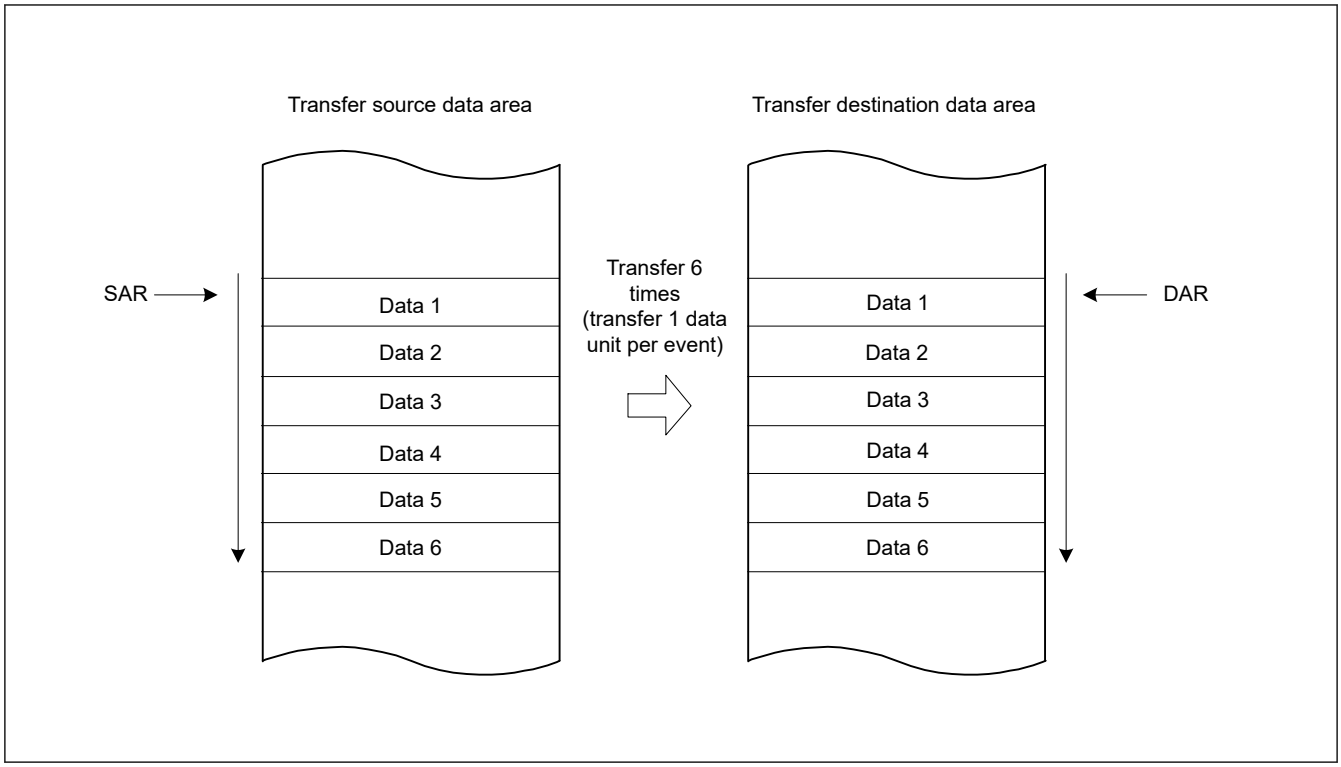


Figure 17.6 Memory map of normal transfer mode (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRA = 0x0006)

17.4.4 Repeat Transfer Mode

The repeat transfer mode allows a 1-byte (8-bit), 1-halfword (16-bit), or 1-word (32-bit) data transfer on a single activation source. Transfer source or transfer destination for the repeat area must be specified in the MRB.DTS bit. The transfer count can be set from 1 to 256. When the specified transfer count is complete, the initial value of the address register specified in the repeat area is restored, the initial value of the transfer counter is restored, and transfer is repeated. The other address register is incremented or decremented continuously or remains unchanged.

When the transfer counter CRAL decrements to 0x00 in repeat transfer mode, the CRAL value is updated to the value set in the CRAH register. As a result, the transfer counter does not clear to 0x00, which disables interrupt requests to the CPU when the MRB.DISEL bit is set to 0. An interrupt request to the CPU is generated when the specified data transfer completes.

Table 17.6 lists the register functions in repeat transfer mode, and Figure 17.7 shows the memory map of repeat transfer mode.

Table 17.6 Register functions in repeat transfer mode

Register	Description	Value written back by writing transfer information	
		When CRAL is not 1	When CRAL is 1
SAR	Transfer source address	Increment, decrement, fixed*1	• When the MRB.DTS bit is 0 Increment, decrement, or fixed*1 • When the MRB.DTS bit is 1 SAR register initial value
DAR	Transfer destination address	Increment, decrement, or fixed*1	• When the MRB.DTS bit is 0 DAR register initial value • When the MRB.DTS bit is 1 Increment, decrement, or fixed*1
CRAH	Retains transfer counter	CRAH	CRAH
CRAL	Transfer counter A	CRAL - 1	CRAH
CRB	Transfer counter B	Not updated	Not updated

Note 1. Write-back is skipped in address-fixed mode.

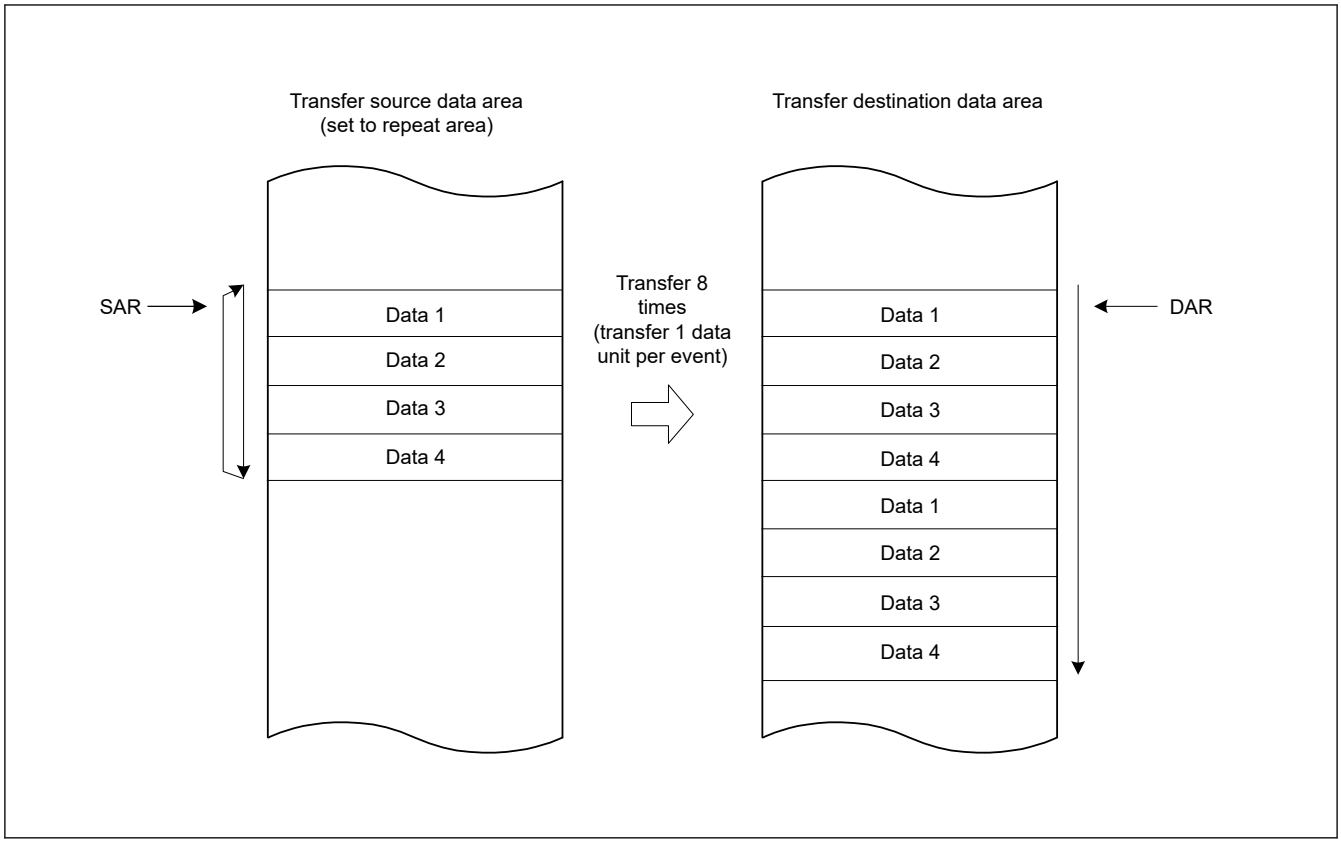


Figure 17.7 Memory map of repeat transfer mode when transfer source is a repeat area (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRAH = 0x04)

17.4.5 Block Transfer Mode

The block transfer mode allows single-block data transfer on a single activation source. Transfer source or transfer destination for the block area must be specified in the MRB.DTS bit. The block size can be set from 1 to 256 bytes, 1 to 256 halfwords (2 to 512 bytes), or 1 to 256 words (4 to 1024 bytes). When transfer of the specified block completes, the initial values of the block size counter CRAL and the address register (the SAR register when the MRB.DTS = 1 or the DAR register when the DTS = 0) specified in the block area are restored. The other address register is incremented or decremented continuously or remains unchanged.

The transfer count (block count) can be set from 1 to 65536. This mode enables an interrupt request to the CPU to be generated at the end of the specified-count block transfer.

Table 17.7 lists the register functions in block transfer mode, and Figure 17.8 shows the memory map for block transfer mode.

Table 17.7 Register functions in block transfer mode

Register	Description	Value written back by writing transfer information
SAR	Transfer source address	• When MRB.DTS bit is 0 Increment, decrement, or fixed*1 • When MRB.DTS bit is 1 SAR register initial value.
DAR	Transfer destination address	• When MRB.DTS bit is 0 DAR register initial value • When MRB.DTS bit is 1 Increment, decrement, or fixed*1.
CRAH	Holds block size	CRAH
CRAL	Block size counter	CRAH
CRB	Block transfer counter	CRB - 1

Note 1. Write-back is skipped in address-fixed mode.

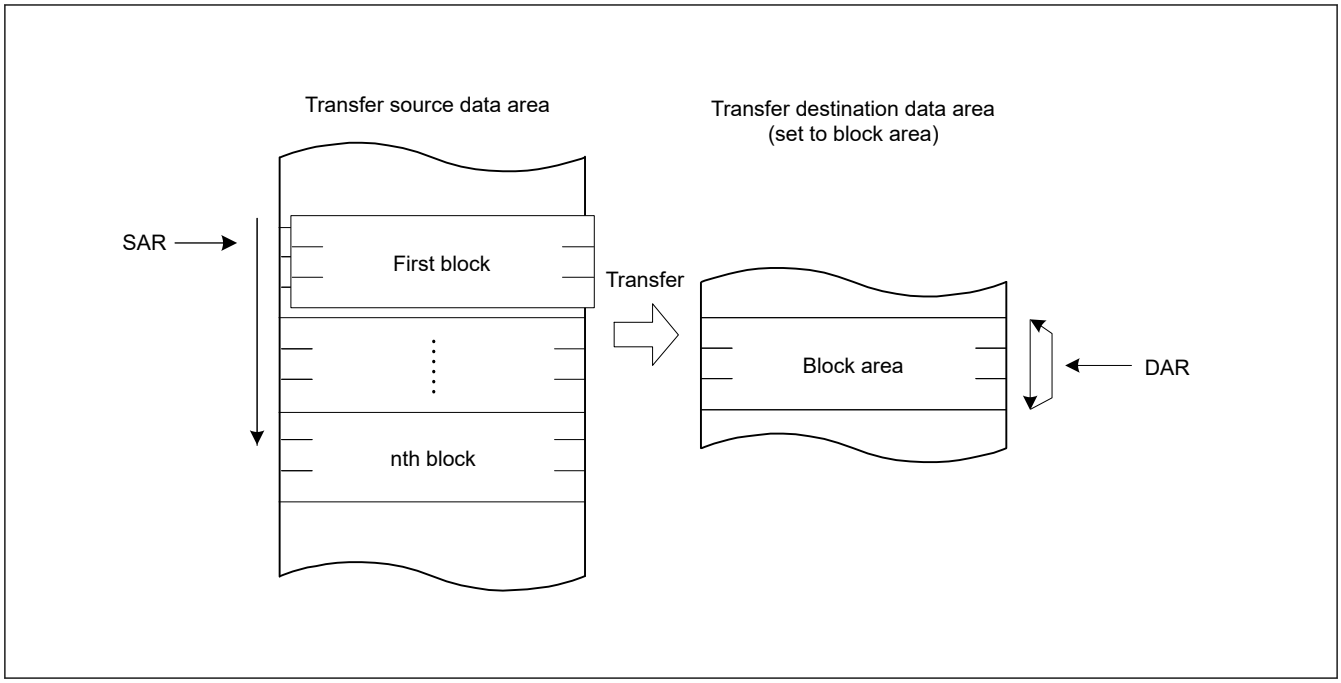


Figure 17.8 Memory map of block transfer mode

17.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 allows chain transfer to be performed continuously on a single activation source. If the MRB.CHNE is set to 1 and CHNS to 0, an interrupt request to the CPU is not generated on completion of the specified number of rounds of transfer or by setting the MRB.DISEL bit to 1. An interrupt request is sent to the CPU each time DTC data transfer is performed. Data transfer has no effect on the ICU.IELSRn.IR flag of the activation source.

The SAR, DAR, CRA, CRB, MRA, and MRB registers can be set independently of each other to define the data transfer. Figure 17.9 shows a chain transfer operation.

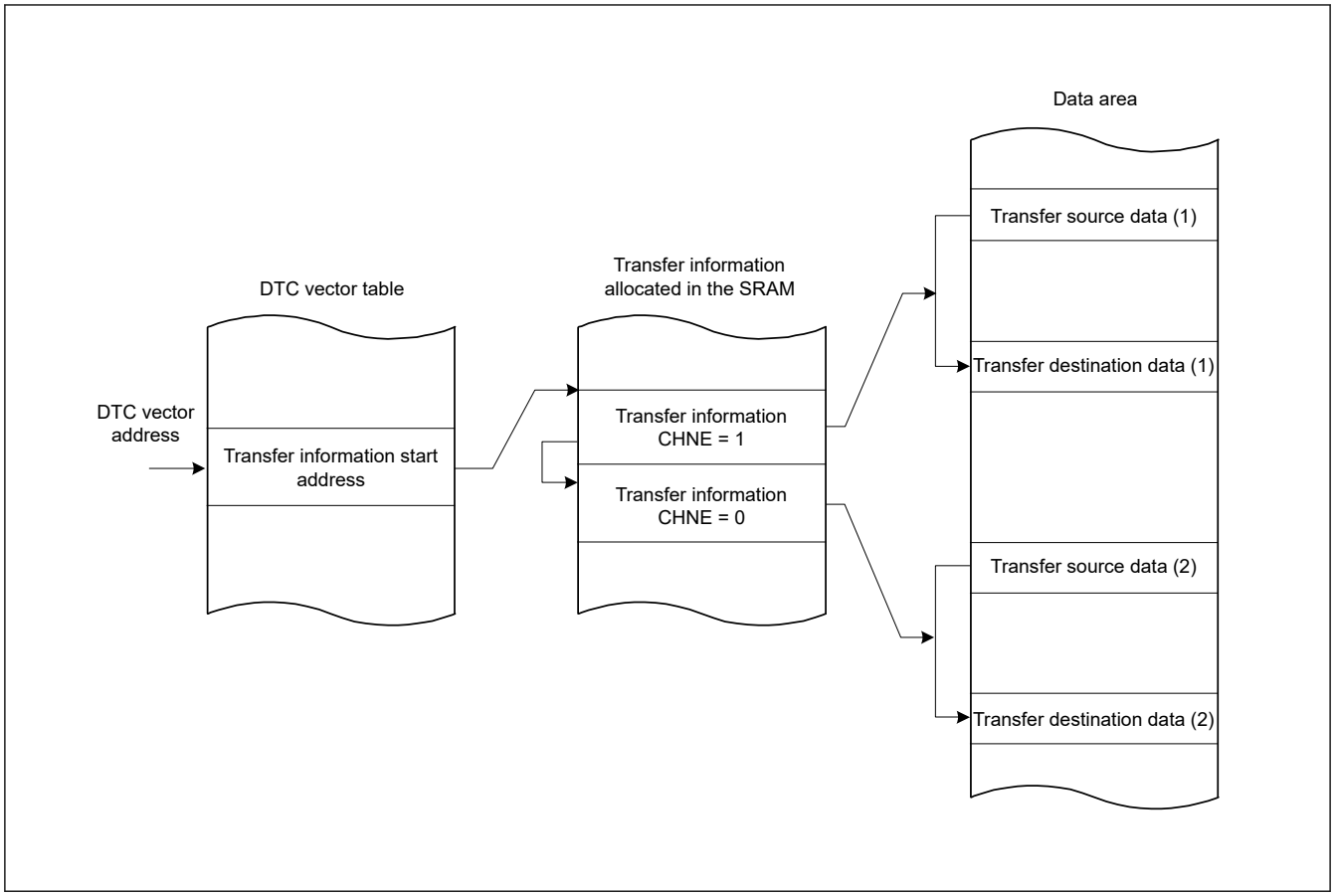


Figure 17.9 Chain transfer operation

Writing 1 to the MRB.CHNE and CHNS bits enables chain transfer to be performed only after completion of the specified data transfer. In repeat transfer mode, chain transfer is performed after completion of the specified data transfer. For details on chain transfer conditions, see Table 17.3.

17.4.7 Operation Timing

Figure 17.10 to Figure 17.13 are timing diagrams that show the minimum number of execution cycles.

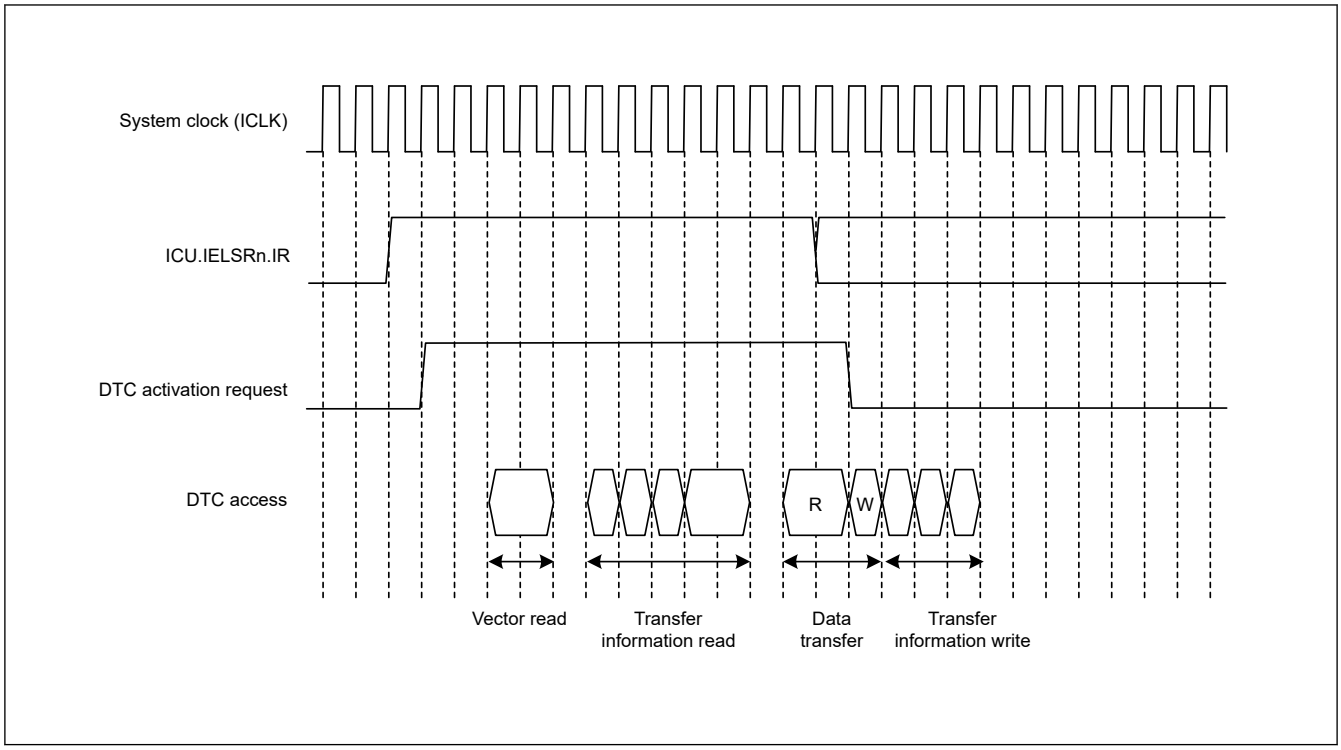


Figure 17.10 Example 1 of DTC operation timing in normal transfer and repeat transfer modes

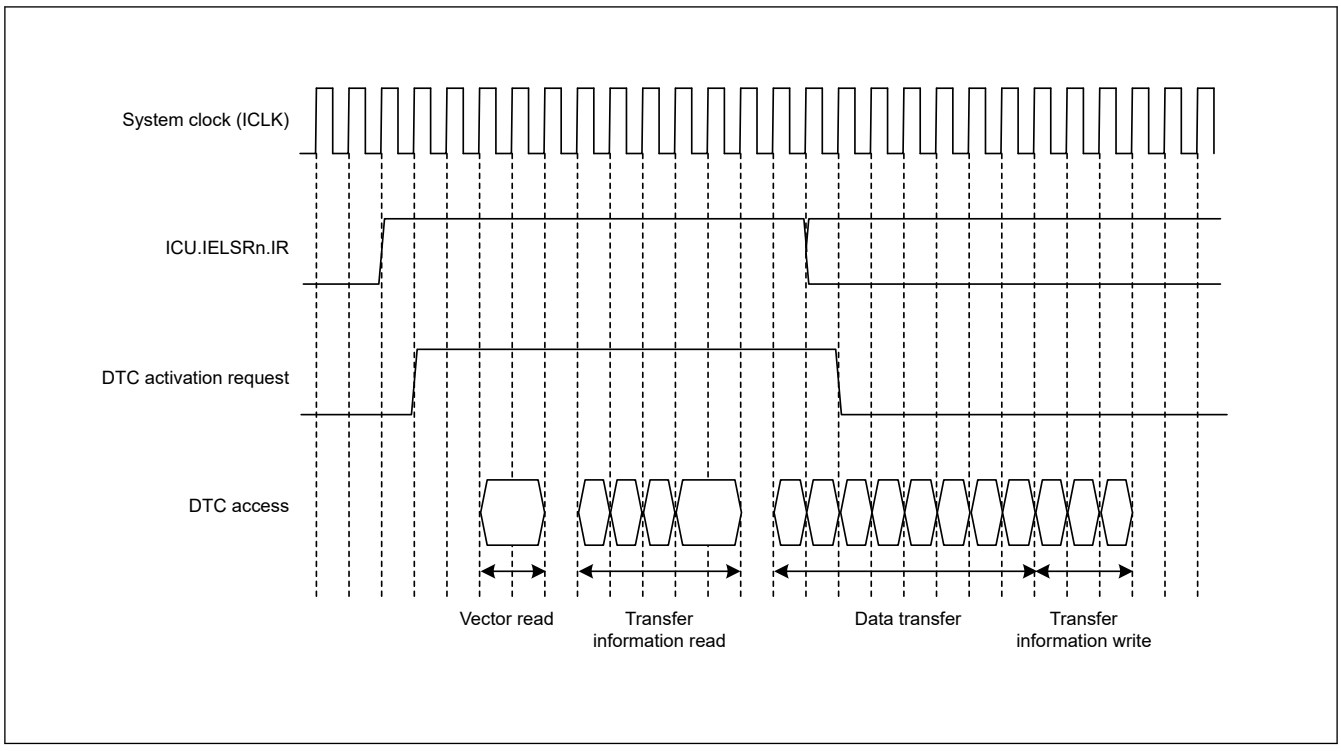
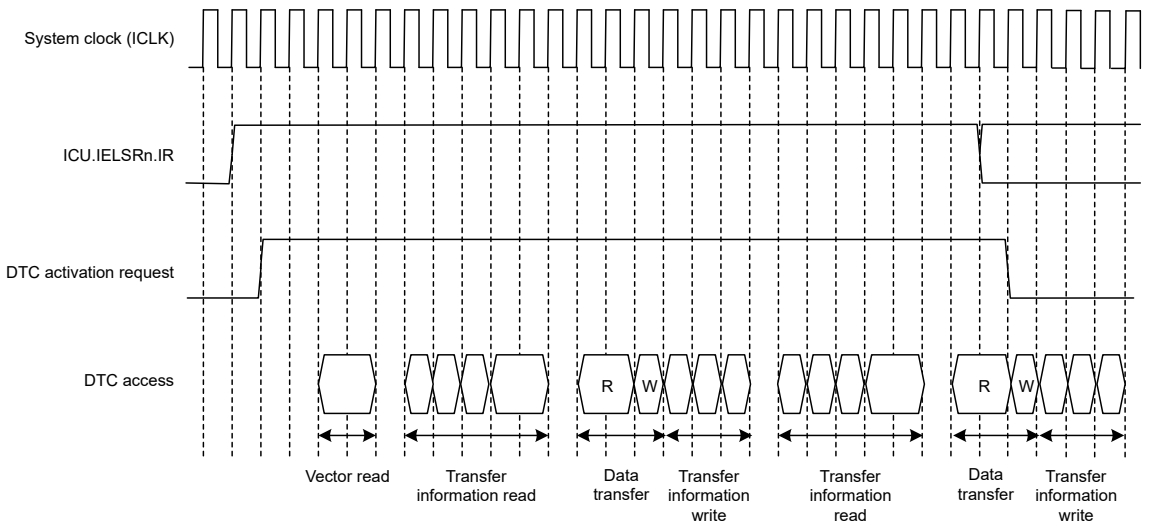
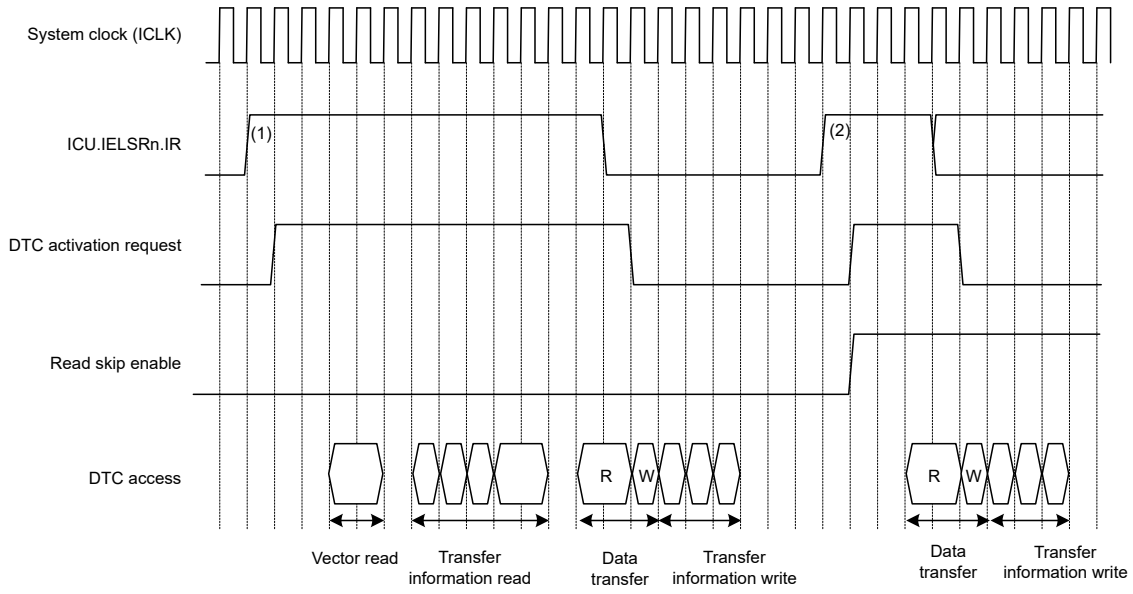


Figure 17.11 Example 2 of DTC operation timing in block transfer mode when the block size = 4



Note: n = Vector number

Figure 17.12 Example 3 of DTC operation timing for chain transfer



Note: n = Vector number
Note: When activation sources (vector numbers) of (1) and (2) are the same and the RRS = 1, the transfer information read for request (2) is skipped.

Figure 17.13 Example of operation when a transfer information read is skipped with the vector, transfer information, and transfer destination data on the SRAM, and the transfer source data on the peripheral module

17.4.8 Vector Security

The security attribute of transfer access of DTC vector n and security attribute of access to the IELSRn (n = 0 to 95) register of ICU are controlled by SAIELSRn bit of ICUSARx (x = G, H or I) registers in CPSCU. For details on the CPSCU.ICUSARx registers, see section 13, Interrupt Controller Unit (ICU).

When the CPSCU.ICUSARx.IELSRnSA bit is 0

- IELSRn register have security attribute
- The vector number of DTC activated by the interrupt request set in IELS behaves as a Secure master.

When the CPSCU.ICUSARx.IELSRnSA bit is 1

- IELSRn register have non-security attribute
- The vector number of DTC activated by the interrupt request set in IELS behaves as a Non-secure master.

For areas accessible to Secure or Non-secure masters, See section 14, Buses, and section 43, Security Features .

17.4.9 Vector privilege

The privileged attribute of transfer access of DTC vector n is controlled bit0 in vector table.

If bit0 corresponding to each vector number in the vector table is set to 0, DTC transfer is privileged access.

If bit0 is set to 1, DTC transfers are unprivileged access.

Refer to for vector table and transfer information.

17.5 DTC Setting Procedure

Before using the DTC, set the DTC Vector Base Register (DTCVBR). Set the ICU.IELSRn.IELS[8:0] bits to 0 to disable the interrupt in the NVIC and follow the procedure in Table 17.8 to set the DTC.

Table 17.8 DTC setting procedure

No.	Step Name	Description
1	Set the DTCCR*1.RRS bit to 0	Set the DTCCR*1.RRS bit to 0 to reset the transfer information read skip flag. After that, the transfer information read is not skipped while the DTC is activated. Be sure to specify this setting when the transfer information is updated.
2	Set transfer information (MRA, MRB, SAR, DAR, CRA, and CRB)	Allocate transfer information (MRA, MRB, SAR, DAR, CRA, and CRB) in the data area. To set transfer information, see section 17.2. Register Descriptions. To allocate transfer information, see section 17.3.1. Allocating Transfer Information and DTC Vector Table.
3	Set transfer information start addresses in the DTC vector table	Set the transfer information start addresses in the DTC vector table. To set the DTC vector table, see section 17.3.1. Allocating Transfer Information and DTC Vector Table.
4	Set the DTCCR*1.RRS bit to 1	Set the DTCCR*1.RRS bit to 1 to enable skipping of the second and subsequent transfer information read cycles for continuous DTC activation from the same interrupt source. The RRS bit can be set to 1, but if this is set during DTC transfer, it becomes valid from the next transfer.
5	Set the ICU.IELSRn.DTCE bit to 1. Set the ICU.IELSRn.IELS[8:0] as interrupt source. The interrupt should be enabled in the NVIC.	Set the ICU.IELSRn.DTCE bit to 1. Set ICU.IELSRn.IELS[8:0] as interrupt sources that trigger DTC. The interrupt must be enabled in the NVIC. See section 13.3.2. Event Number in section 13, Interrupt Controller Unit (ICU).
6	Set the enable bit for an activation source interrupt	Set the enable bit for the activation source interrupts to 1. When a source interrupt is generated, the DTC is activated. To set the interrupt source enable bit, see the settings for the modules that are to be the activation sources.
7	Set the DTCST.DTCST bit to 1	Set the DTC Module Start bit (DTCST.DTCST) to 1.

Note: The DTCST.DTCST bit can be set even if the setting for each activation source is not completed.
Note: When DTCSAR.DTCSTSA = 0 is set, DTCST.DTCST = 1 must be set in the Secure state to use DTC in the Non-secure state.
Note 1. When used in Secure state, access DTCCR_SEC instead of DTCCR.

17.6 Examples of DTC Usage

17.6.1 Normal Transfer

This section provides an example of DTC usage and its application when receiving 128 bytes of data from an SCI.

(1) Transfer information settings

In the MRA register, select a fixed source address (MRA.SM[1:0] = 00b), normal transfer mode (MRA.MD[1:0] = 00b), and byte-sized transfer (MRA.SZ[1:0] = 00b). In the MRB register, specify incrementation of the destination address (MRB.DM[1:0] = 10b) and single data transfer by a single interrupt (MRB.CHNE = 0 and MRB.DISEL = 0). The MRB.DTS bit can be set to any value. Set the RDR register address of the SCI in the SAR register, the start address of the SRAM area for data storage in the DAR register, and 128 (0x0080) in the CRA register. The CRB register can be set to any value.

(2) DTC vector table settings

The start address of the transfer information for the RXI interrupt is set in the vector table for the DTC.

(3) ICU settings and DTC module activation

Set the ICU.IELSRn.DTCE bit to 1 and set ICU.IELSRn.IELS[8:0] as the SCI interrupt. The interrupt must be enabled in the NVIC. Set the DTCST.DTCST bit to 1.

(4) SCI settings

Enable the SCIn_RXI (n = 0 to 4, 9) interrupt by setting the SCR.RIE bit in the SCI to 1. If a reception error occurs during the SCI receive operation, reception stops. To manage this, use settings that allow the CPU to accept receive error interrupts.

(5) DTC transfer

Each time a reception of 1 byte by the SCI is complete, an SCIn_RXI interrupt is generated to activate the DTC. The DTC transfers the received byte from the RDR of the SCI to the SRAM, after which the DAR register is incremented and the CRA register is decremented.

(6) Interrupt handling

After 128 rounds of data transfer are complete and the value in the CRA register becomes 0, an SCIn_RXI interrupt request is generated for the CPU. Complete the process in the handling routine for this interrupt.

17.6.2 Chain transfer

This section provides an example of chain transfer by the DTC and describes its use in the output of pulses by the General PWM Timer (GPT). You can use chain transfer to transfer PWM timer compare data and change the period of the PWM timer for the GPT.

For the first of the chain transfers, normal transfer mode is specified for transfer to the GPT32n.GTCCRC register (n = 0 to 7) or GPT16m.GTCCRC register (m = 8 to 13). For the second transfer, normal transfer mode is specified for transfer to the GPT32n.GTCCRE register (n = 0 to 7) or GPT16m.GTCCRE register (m = 8 to 13). For the third transfer of the chained transfer, normal transfer mode for transfer to the GPT32n.GTPBR register (n = 0 to 7) or GPT16m.GTPBR register (m = 8 to 13) is specified. This is because clearing of the activation source and generation of an interrupt on completion of the specified number of transfers are restricted to the third of the chain transfers, that is, transfer while MRB.CHNE = 0.

The following example shows how to use the counter overflow interrupt with the GPT320.GTPR register as an activating source for the DTC.

(1) First transfer information setting

Set up transfer to the GPT320.GTCCRC register.

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up chain transfer (MRB.CHNE = 1 and MRB.CHNS = 0).

4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the GPT320.GTCCRC register.
6. Set the CRAH and CRAL registers to the size of the data table. The CRB register can be set to any value.

(2) Second transfer information setting

Set up for transfer to the GPT320.GTCCRE register.

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up chain transfer (MRB.CHNE = 1, MRB.CHNS = 0).
4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the GPT320.GTCCRE register.
6. Set the CRAH and CRAL registers to the size of the data table. The CRB register can be set to any value.

(3) Third transfer information set

Set up transfer to the GPT320.GTPBR register.

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up single data transfer per interrupt (MRB.CHNE = 0, MRB.DISEL = 0). The MRB.DTS bit can be set to any value.
4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the GPT320.GTPBR register.
6. Set the CRA register to the size of the data table. The CRB register can be set to any value.

(4) Transfer information assignment

Place the transfer information for use in the transfer to the GPT320.GTPBR immediately after the transfer control information for use in the GPT320.GTCCRC and GPT320.GTCCRE registers.

(5) DTC vector table

In the DTC vector table, set the address where the transfer control information for use in transfer to the GPT320.GTCCRC and GPT320.GTCCRE registers starts.

(6) ICU setting and DTC module activation

1. Set the ICU.IELSRn.DTCE bit associated with the GPT320 counter overflow interrupt.
2. Set the ICU.IELSRn.IELS[8:0] bits and specify the GPT320 counter overflow.
3. Set the DTCST.DTCST bit to 1.

(7) GPT settings

1. Set the GPT320.GTIOR register so that the GTCCRA and GTCCRB registers operate as output compare registers.
2. Set the default PWM timer compare values in the GPT320.GTCCRA and GPT320.GTCCRB registers and the next PWM timer compare values in the GPT320.GTCCRC and GPT320.GTCCRE registers.
3. Set the default PWM timer period values in the GPT320.GTPR register and the next PWM timer period values in the GPT320.GTPBR register.
4. Set 1 to the output bit in PmnPFS.PDR, and set 00011b to the Peripheral Select bits in PmnPFS.PSEL[4:0].

(8) GPT activation

Set the GPT320.GTSTR.CSTRT bits to 1 to start the GPT320.GTCNT counter.

(9) DTC transfer

Each time a GPT320 counter overflow is generated with the GPT320.GTPR register, the next PWM timer compare values are transferred to the GPT320.GTCCRC and GPT320.GTCCRE registers. The setting for the next PWM timer period is transferred to the GPT320.GTPBR register.

(10) Interrupt handling

After the specified rounds of data transfer are complete, for example when the value in the CRA register for GPT transfer becomes 0, a GPT320 counter overflow interrupt request is issued for the CPU. Complete the process for this interrupt in the handling routine.

17.6.3 Chain Transfer when Counter = 0

The second data transfer is performed only when the transfer counter is set to 0 in the first data transfer, and the first data transfer information is repeatedly changed in the second transfer. Chain transfer enables transfers to be repeated 256 times or more.

The following procedure shows an example of configuring a 1-KB input buffer, where the input buffer is set so that its lower address starts with 0x00. Figure 17.14 shows a chain transfer when the counter = 0.

1. Set the normal transfer mode to input data for the first data transfer. Set the following:
   (a) Transfer source address = fixed.
   (b) CRA register = 0x0200 (512) times.
   (c) MRB.CHNE bit = 1 (chain transfer is enabled).
   (d) MRB.CHNS bit = 1 (chain transfer is performed only when the transfer counter is 0).
   (e) MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
2. Prepare the upper 8-bit address of the start address at every 512 times of the transfer destination address for the first data transfer in different area such as the flash. For example, when setting the input buffer to 0x8000 to 0x83FF, prepare 0x82 and 0x80.
3. For the second data transfer:
   (a) Set the repeat transfer mode (with transfer source and destination address = fixed.) to reset the transfer counter of the first data transfer.
   (b) Specify the CRA register in the first transfer information area for the transfer destination.
   (c) Set the MRB.CHNE bit = 1 (chain transfer is enabled).
   (d) Set the MRB.CHNS bit = 0 (select continuous chain transfer).
   (e) Set the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
   (f) CRA register = 0x0101 (The transfer count is 1).
4. For the third data transfer:
   (a) Set the repeat transfer mode (with the source as the repeat area) to reset the transfer destination address of the first data transfer.
   (b) Specify the upper 8 bits of the DAR register in the first transfer information area for the transfer destination.
   (c) Set the MRB.CHNE bit = 0 (chain transfer is disabled).
   (d) Set the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
   (e) When setting the input buffer to 0x8000 to 0x83FF, also set the transfer counter to 2.
5. The first data transfer is performed by an interrupt 512 times. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the transfer counter of the first data transfer to 0x0200. The lower 8 bits of the transfer destination address and the transfer counter of the first data transfer becomes 0x0200.
6. The second data transfer is performed by an interrupt 1 times. When the transfer counter of the first data transfer becomes 0, the third data transfer starts. Set the upper 8 bits of the transfer destination address of the first data transfer to

0x82. The lower 8 bits of the transfer destination address becomes 0x00 and the transfer counter of the first data transfer becomes 0x0200.

7. In succession, the first data transfer is performed by an interrupt 512 times as specified for the first data transfer. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the transfer counter of the first data transfer to 0x0200. The lower 8 bits of the transfer destination address and the transfer counter of the first data transfer becomes 0x0200.
8. The second data transfer is performed by an interrupt 1 times. When the transfer counter of the first data transfer becomes 0, the third data transfer starts. Set the upper 8 bits of the transfer destination address of the first data transfer to 0x80. The lower 8 bits of the transfer destination address becomes 0x00 and the transfer counter of the first data transfer becomes 0x0200.
9. Steps 5 to 8 are repeated indefinitely. Because the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

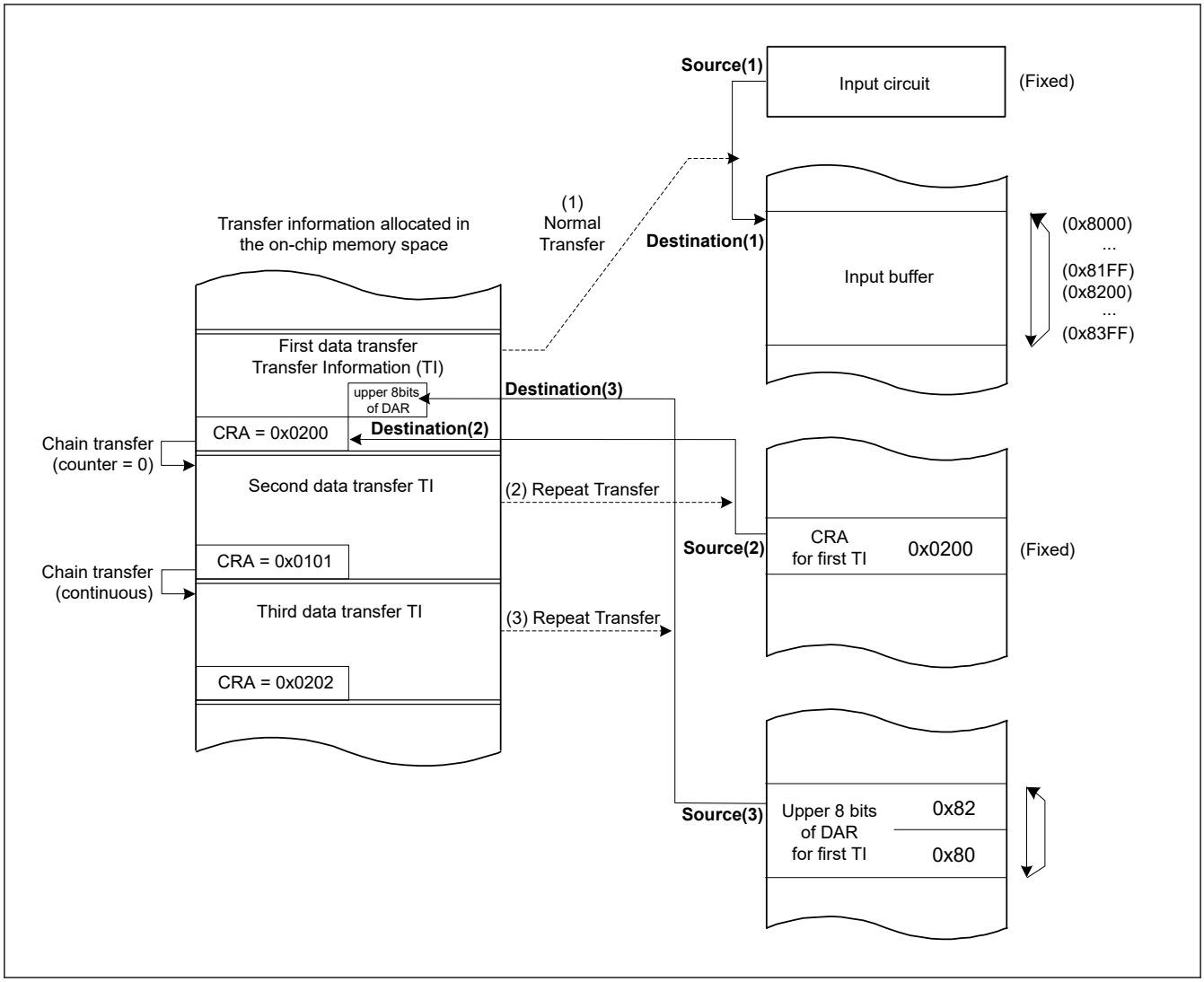


Figure 17.14 Chain transfer when counter = 0

17.7 Processing on DTC Transfer Error

If the access error occurs during DTC transfer, the DTC immediately stops access during transfer. DTC is stopped only for the vector number that caused the error. Therefore, inform the vector number that caused the error to the ICU and cleared the corresponding ICU setting (IELSR). After that, if there is a request other than the vector number which caused the error, it will be re-arbitration as it is.

The error response is informed to ICU when the transfer error occurs. ICU clears the ICU.IELSRn of the corresponding vector number which caused the transfer error. Write back to SRAM is not performed.

17.8 Interrupt

17.8.1 Interrupt Request of Transfer End

When the DTC completes data transfer of the specified count or when data transfer with MRB.DISEL set to 1 is complete, a DTC activation source generates an interrupt to the CPU. Two types of interrupt are available: interrupts triggered by a DTC activation (per channel) and an interrupt triggered by the event signal DTC0_COMPLETE (common to all channels). Interrupts to the CPU are controlled according to the settings in the NVIC and the ICU.IELSRn.IELS[8:0] bits. See section 13, Interrupt Controller Unit (ICU). The DTC prioritizes activation sources by granting the smaller interrupt vector numbers higher priority. The priority of interrupts to the CPU is determined by the NVIC priority.

When the transfer destination is the external bus, an interrupt request is generated upon completion of data write to the write buffer not to the actual transfer destination.

17.8.2 Interrupt Request of Transfer Error

The error response detection interrupt request (DMA0_TRANSERR) is generated from the DMAC/DTC when the transfer error is detected during DTC transfer.

The types of interrupts that occur when the DTC transfer error occurs are listed in Table 17.9. The Table 17.9 also shows error information stored when a transfer error occurs.

Table 17.9 Interrupt and error information due to DTC transfer error cause

Transfer error factor	NMI/RESET*1 Request	Interrupt Request	Bus Error Status	Error Address Error R/W	Error Channel Information
MSAU Error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. MSERRSTAT	BUS.BMSA4ERRADD BUS.BMSA4ERRRW	DTC.DTEVR
Master MPU Error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. MMERRSTAT	BUS.BUS4ERRADD BUS.BUS4ERRRW	DTC.DTEVR
Illegal Access Error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. ILERRSTAT	BUS.BUS4ERRADD BUS.BUS4ERRRW	DTC.DTEVR
Slave Bus Error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. SLERRSTAT	BUS.BUS4ERRADD BUS.BUS4ERRRW	DTC.DTEVR
TrustZone Filter error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. STERRSTAT	BUS.BUS4ERRADD BUS.BUS4ERRRW	DTC.DTEVR

Note 1. If ICU.NMIER.BUSEN is enabled with DMA0_TRANSERR set in ICU.IELSR, NMI and an interrupt will occur due to a transfer error caused by DMA. it is able to be generated only NMI by not setting DMA0_TRANSERR to ICU.IELSR.

17.9 Event Link

The DTC can produce an event link request on completion of one transfer request. When the destination for transfer is an external bus, however, the event link request will be issued after completion of writing to the write buffer rather than after completion of writing to the actual destination for transfer.

17.10 Low Power Consumption Function

Before transitioning to the module-stop state, Software Standby mode or Deep Software Standby mode, set the DTCST.DTCST bit to 0, and then perform the operations described in the following sections.

(1) Module-stop function

Writing 1 to the MSTPCRA.MSTPA22 bit enables the module-stop function of the DTC. If a DTC transfer is in progress when 1 is written to the MSTPCRA.MSTPA22 bit, the transition to the module-stop state proceeds after the DTC transfer ends. While the MSTPCRA.MSTPA22 bit is 1, accessing the DTC registers is prohibited. Writing 0 to the MSTPCRA.MSTPA22 bit releases the DTC from the module-stop state.

(2) Software Standby mode and Deep Software Standby mode

Use the settings described in section 10.7.3.1. Transition to Software Standby Mode or section 10.7.4.1. Transition to Deep Software Standby Mode.

If DTC transfer operations are in progress when the WFI instruction is executed, the transition to Software Standby mode or Deep Software Standby mode follows the completion of the DTC transfer.

(3) Notes on Low Power Consumption Function

For the WFI instruction and the register setting procedure, see section 10, Low Power Modes.

To perform a DTC transfer after returning from a low power mode.

To use a request that is generated in Software Standby mode as an interrupt request to the CPU but not as a DTC activation request, specify the CPU as the interrupt request destination as described in section 13.5.5. Selecting Interrupt Request Destinations, then execute the WFI instruction.

17.11 Usage Notes

17.11.1 Transfer Information Start Address

You must set multiples of 4 for the transfer information start addresses in the vector table. Otherwise, such addresses are accessed with their lowest 2 bits regarded as 00b.

17.11.2 Restriction for OSPI

There is a restriction to using OSPI, see section 37.4.5. Restriction in 8D-8D-8D profile 1.0 format.

18. Event Link Controller (ELC)

18.1 Overview

The Event Link Controller (ELC) uses the event requests generated by various peripheral modules as source signals to connect them to different modules, allowing direct link between the modules without CPU intervention.

Table 18.1 lists the ELC specifications, and Figure 18.1 shows a block diagram.

Table 18.1 ELC Specifications

Item	Description
Event link function	246 types of event signals can be directly connected to modules. The ELC generates the ELC event signal, and events that activate the DTC.
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security and Privilege attribution can be set for each registers

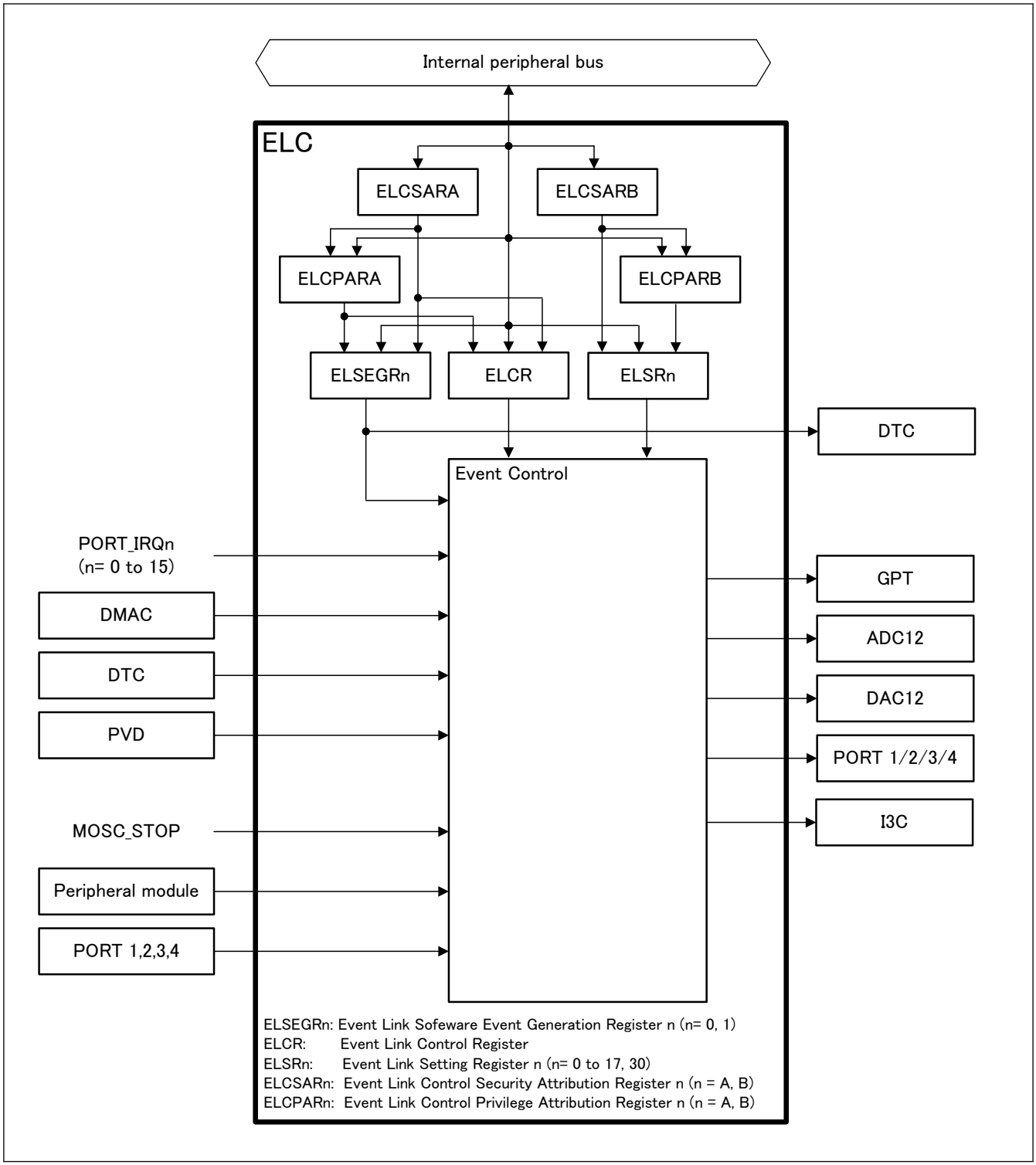


Figure 18.1 ELC block diagram

18.2 Register Descriptions

18.2.1 ELCR : Event Link Controller Register

Base address: ELC = 0x4020_1000
ELC_NS = 0x5020_1000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ELCON	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	ELCON	All Event Link Enable 0: ELC function is disabled. 1: ELC function is enabled.	R/W

Note: S-TYPE-3, P-TYPE-3

The ELCR register controls the ELC operation.

18.2.2 ELSEGRn : Event Link Software Event Generation Register n (n = 0, 1)

Base address: ELC = 0x4020_1000
ELC_NS = 0x5020_1000

Offset address: 0x04 + 0x04 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	WI	WE	—	—	—	—	—	SEG
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SEG	Software Event Generation 0: Normal operation 1: Software event is generated.	W
5:1	—	These bits are read as 0. The write value should be 0.	R/W
6	WE	SEG Bit Write Enable 0: Write to SEG bit disabled. 1: Write to SEG bit enabled.	R/W
7	WI	ELSEGR Register Write Disable 0: Write to ELSEGR register enabled. 1: Write to ELSEGR register disabled.	W

Note: S-TYPE-3, P-TYPE-3

SEG bit (Software Event Generation)

When 1 is written to the SEG bit while the WE bit is 1, a software event is generated. This bit is read as 0. Even when 1 is written to this bit, data is not stored. The WE bit must be set to 1 before writing to this bit.

A software event can trigger a linked DTC event.

WE bit (SEG Bit Write Enable)

The SEG bit can only be written to when the WE bit is 1. Clear the WI bit to 0 before writing to this bit.

[Setting condition]

- If 1 is written to this bit while the WI bit is 0, this bit becomes 1.

[Clearing condition]

- If 0 is written to this bit while the WI bit is 0, this bit becomes 0.

WI bit (ELSEGR Register Write Disable)

The ELSEGR register can only be written to when the write value to the WI bit is 0. This bit is read as 1. Before setting the WE or SEG bit, the WI bit must be set to 0.

18.2.3 ELSRn : Event Link Setting Register n (n = 0 to 17, 30)

Base address: ELC = 0x4020_1000
ELC_NS = 0x5020_1000

Offset address: 0x20 + 0x04 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ELS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	ELS[8:0]	Event Link Select 0x000: Event output disabled for the associated peripheral module 0x001: Number setting for the event signal to be linked ⋮ 0x1BA: Number setting for the event signal to be linked Others: Settings prohibited	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: P-TYPE-3

Note: This register has different secure access permission depending on the setting of Trusted Event Route Control Register (TEVTRCR).
If the security attribution is configured as Secure,

- Secure access is allowed.
- Non-secure write access is ignored and Non-secure read access is read as 0, TrustZone access error is generated.

If the security attribution is configured as Non-secure and the trusted event route is disabled,

- Secure write access is ignored and Secure read access is read as 0, TrustZone access error is generated.
- Non-secure access is allowed.

If the security attribution is configured as Non-secure and the trusted event route is enabled,

- Secure access is allowed.
- Non-secure write access is ignored and Non-secure read access is allowed.
- TrustZone access error is not generated.

The register specifies an event signal to be linked to each peripheral module. Table 18.2 shows the association between the register and the peripheral modules. Table 18.3 shows the association between the event signal names set in the register and the signal numbers.

Table 18.2 Association between the registers and peripheral functions (1 of 2)

Register name	Peripheral function (module)	Event name
ELSR0	GPT (A)	ELC_GPTA
ELSR1	GPT (B)	ELC_GPTB
ELSR2	GPT (C)	ELC_GPTC
ELSR3	GPT (D)	ELC_GPTD
ELSR4	GPT (E)	ELC_GPTE
ELSR5	GPT (F)	ELC_GPTF
ELSR6	GPT (G)	ELC_GPTG
ELSR7	GPT (H)	ELC_GPTH

Table 18.2 Association between the registers and peripheral functions (2 of 2)

Register name	Peripheral function (module)	Event name
ELSR8	ADC12A0	ELC_AD00
ELSR9	ADC12B0	ELC_AD01
ELSR10	ADC12A1	ELC_AD10
ELSR11	ADC12B1	ELC_AD11
ELSR12	DAC12 channel 0	ELC_DA0
ELSR13	DAC12 channel 1	ELC_DA1
ELSR14	PORT1	ELC_PORT1
ELSR15	PORT2	ELC_PORT2
ELSR16	PORT3	ELC_PORT3
ELSR17	PORT4	ELC_PORT4
ELSR30	I3C	ELC_I3C

Table 18.3 Association between event signal names set in .ELS[8:0] bits and signal numbers (1 of 7)

Event number	Interrupt request source	Name	Description
0x001	Port	PORT_IRQ0*1	External pin interrupt 0
0x002		PORT_IRQ1*1	External pin interrupt 1
0x003		PORT_IRQ2*1	External pin interrupt 2
0x004		PORT_IRQ3*1	External pin interrupt 3
0x005		PORT_IRQ4*1	External pin interrupt 4
0x006		PORT_IRQ5*1	External pin interrupt 5
0x007		PORT_IRQ6*1	External pin interrupt 6
0x008		PORT_IRQ7*1	External pin interrupt 7
0x009		PORT_IRQ8*1	External pin interrupt 8
0x00A		PORT_IRQ9*1	External pin interrupt 9
0x00B		PORT_IRQ10*1	External pin interrupt 10
0x00C		PORT_IRQ11*1	External pin interrupt 11
0x00D		PORT_IRQ12*1	External pin interrupt 12
0x00E		PORT_IRQ13*1	External pin interrupt 13
0x00F		PORT_IRQ14*1	External pin interrupt 14
0x010		PORT_IRQ15*1	External pin interrupt 15
0x011	DMAC0	DMAC00_INT	DMAC0 transfer end 0
0x012		DMAC01_INT	DMAC0 transfer end 1
0x013		DMAC02_INT	DMAC0 transfer end 2
0x014		DMAC03_INT	DMAC0 transfer end 3
0x015		DMAC04_INT	DMAC0 transfer end 4
0x016		DMAC05_INT	DMAC0 transfer end 5
0x017		DMAC06_INT	DMAC0 transfer end 6
0x018		DMAC07_INT	DMAC0 transfer end 7
0x021	DTC0	DTC0_DTCEND	DTC0 transfer end
0x038	PVD	PVD_PVD1	Voltage monitor 1 interrupt
0x039		PVD_PVD2	Voltage monitor 2 interrupt

Table 18.3 Association between event signal names set in .ELS[8:0] bits and signal numbers (2 of 7)

Event number	Interrupt request source	Name	Description
0x03E	MOSC	MOSC_STOP	Mail Clock oscillation stop
0x040	ULPT0	ULPT0_ULPTI	Underflow
0x041		ULPT0_ULPTCMAI	Compare match A
0x042		ULPT0_ULPTCMBI	Compare match B
0x043	ULPT1	ULPT1_ULPTI	Underflow
0x044		ULPT1_ULPTCMAI	Compare match A
0x045		ULPT1_ULPTCMBI	Compare match B
0x046	AGT0	AGT0_AGTI	AGT interrupt
0x047		AGT0_AGTCMAI	Compare match A
0x048		AGT0_AGTCMBI	Compare match B
0x049	AGT1	AGT1_AGTI	AGT interrupt
0x04A		AGT1_AGTCMAI	Compare match A
0x04B		AGT1_AGTCMBI	Compare match B
0x052	IWDT	IWDT_NMIUNDF	IWDT underflow
0x053	WDT	WDT0_NMIUNDF	WDT underflow
0x056	RTC	RTC_PRD	Periodic interrupt
0x05C	IIC0	IIC0_RXI	Receive data full
0x05D		IIC0_TXI	Transmit data empty
0x05E		IIC0_TEI	Transmit end
0x05F		IIC0_EEI	Transfer error
0x061	IIC1	IIC1_RXI	Receive data full
0x062		IIC1_TXI	Transmit data empty
0x063		IIC1_TEI	Transmit end
0x064		IIC1_EEI	Transfer error
0x07B	ACMPHS	ACMP_HS0*1	Comparator interrupt 0
0x07C		ACMP_HS1*1	Comparator interrupt 1
0x083	ELC	ELC_SWEVT0	Software event 0
0x084		ELC_SWEVT1	Software event 1
0x088	I/O Port	IOPORT_GROUP1	Port 1 event
0x089		IOPORT_GROUP2	Port 2 event
0x08A		IOPORT_GROUP3	Port 3 event
0x08B		IOPORT_GROUP4	Port 4 event
0x0A0	GPT	GPT_UVWEDGE	UVW edge event

Table 18.3 Association between event signal names set in .ELS[8:0] bits and signal numbers (3 of 7)

Event number	Interrupt request source	Name	Description
0x0A1	GPT0	GPT0_CCMPA	Compare match A
0x0A2		GPT0_CCMPB	Compare match B
0x0A3		GPT0_CMPC	Compare match C
0x0A4		GPT0_CMPD	Compare match D
0x0A5		GPT0_CMPE	Compare match E
0x0A6		GPT0_CMPF	Compare match F
0x0A7		GPT0_OVF	Overflow
0x0A8		GPT0_UDF	Underflow
0x0A9		GPT0_PC	Cycle count function end
0x0AA	GPT1	GPT1_CCMPA	Compare match A
0x0AB		GPT1_CCMPB	Compare match B
0x0AC		GPT1_CMPC	Compare match C
0x0AD		GPT1_CMPD	Compare match D
0x0AE		GPT1_CMPE	Compare match E
0x0AF		GPT1_CMPF	Compare match F
0x0B0		GPT1_OVF	Overflow
0x0B1		GPT1_UDF	Underflow
0x0B2		GPT1_PC	Cycle count function end
0x0B3	GPT2	GPT2_CCMPA	Compare match A
0x0B4		GPT2_CCMPB	Compare match B
0x0B5		GPT2_CMPC	Compare match C
0x0B6		GPT2_CMPD	Compare match D
0x0B7		GPT2_CMPE	Compare match E
0x0B8		GPT2_CMPF	Compare match F
0x0B9		GPT2_OVF	Overflow
0x0BA		GPT2_UDF	Underflow
0x0BB		GPT2_PC	Cycle count function end
0x0BC	GPT3	GPT3_CCMPA	Compare match A
0x0BD		GPT3_CCMPB	Compare match B
0x0BE		GPT3_CMPC	Compare match C
0x0BF		GPT3_CMPD	Compare match D
0x0C0		GPT3_CMPE	Compare match E
0x0C1		GPT3_CMPF	Compare match F
0x0C2		GPT3_OVF	Overflow
0x0C3		GPT3_UDF	Underflow
0x0C4		GPT3_PC	Cycle count function end

Table 18.3 Association between event signal names set in .ELS[8:0] bits and signal numbers (4 of 7)

Event number	Interrupt request source	Name	Description
0x0C5	GPT4	GPT4_CCMPA	Compare match A
0x0C6		GPT4_CCMPB	Compare match B
0x0C7		GPT4_CMPC	Compare match C
0x0C8		GPT4_CMPD	Compare match D
0x0C9		GPT4_CMPE	Compare match E
0x0CA		GPT4_CMPF	Compare match F
0x0CB		GPT4_OVF	Overflow
0x0CC		GPT4_UDF	Underflow
0x0CE	GPT5	GPT5_CCMPA	Compare match A
0x0CF		GPT5_CCMPB	Compare match B
0x0D0		GPT5_CMPC	Compare match C
0x0D1		GPT5_CMPD	Compare match D
0x0D2		GPT5_CMPE	Compare match E
0x0D3		GPT5_CMPF	Compare match F
0x0D4		GPT5_OVF	Overflow
0x0D5		GPT5_UDF	Underflow
0x0D7	GPT6	GPT6_CCMPA	Compare match A
0x0D8		GPT6_CCMPB	Compare match B
0x0D9		GPT6_CMPC	Compare match C
0x0DA		GPT6_CMPD	Compare match D
0x0DB		GPT6_CMPE	Compare match E
0x0DC		GPT6_CMPF	Compare match F
0x0DD		GPT6_OVF	Overflow
0x0DE		GPT6_UDF	Underflow
0x0E0	GPT7	GPT7_CCMPA	Compare match A
0x0E1		GPT7_CCMPB	Compare match B
0x0E2		GPT7_CMPC	Compare match C
0x0E3		GPT7_CMPD	Compare match D
0x0E4		GPT7_CMPE	Compare match E
0x0E5		GPT7_CMPF	Compare match F
0x0E6		GPT7_OVF	Overflow
0x0E7		GPT7_UDF	Underflow
0x0E9	GPT8	GPT8_CCMPA	Compare match A
0x0EA		GPT8_CCMPB	Compare match B
0x0EB		GPT8_CMPC	Compare match C
0x0EC		GPT8_CMPD	Compare match D
0x0ED		GPT8_CMPE	Compare match E
0x0EE		GPT8_CMPF	Compare match F
0x0EF		GPT8_OVF	Overflow
0x0F0		GPT8_UDF	Underflow
0x0F1		GPT8_PC	Cycle count function end

Table 18.3 Association between event signal names set in .ELS[8:0] bits and signal numbers (5 of 7)

Event number	Interrupt request source	Name	Description
0x0F2	GPT9	GPT9_CCMPA	Compare match A
0x0F3		GPT9_CCMPB	Compare match B
0x0F4		GPT9_CMPC	Compare match C
0x0F5		GPT9_CMPD	Compare match D
0x0F6		GPT9_CMPE	Compare match E
0x0F7		GPT9_CMPF	Compare match F
0x0F8		GPT9_OVF	Overflow
0x0F9		GPT9_UDF	Underflow
0x0FA		GPT9_PC	Cycle count function end
0x0FB	GPT10	GPT10_CCMPA	Compare match A
0x0FC		GPT10_CCMPB	Compare match B
0x0FD		GPT10_CMPC	Compare match C
0x0FE		GPT10_CMPD	Compare match D
0x0FF		GPT10_CMPE	Compare match E
0x100		GPT10_CMPF	Compare match F
0x101		GPT10_OVF	Overflow
0x102		GPT10_UDF	Underflow
0x103		GPT10_PC	Cycle count function end
0x104	GPT11	GPT11_CCMPA	Compare match A
0x105		GPT11_CCMPB	Compare match B
0x106		GPT11_CMPC	Compare match C
0x107		GPT11_CMPD	Compare match D
0x108		GPT11_CMPE	Compare match E
0x109		GPT11_CMPF	Compare match F
0x10A		GPT11_OVF	Overflow
0x10B		GPT11_UDF	Underflow
0x10D	GPT12	GPT12_CCMPA	Compare match A
0x10E		GPT12_CCMPB	Compare match B
0x10F		GPT12_CMPC	Compare match C
0x110		GPT12_CMPD	Compare match D
0x111		GPT12_CMPE	Compare match E
0x112		GPT12_CMPF	Compare match F
0x113		GPT12_OVF	Overflow
0x114		GPT12_UDF	Underflow
0x116	GPT13	GPT13_CCMPA	Compare match A
0x117		GPT13_CCMPB	Compare match B
0x118		GPT13_CMPC	Compare match C
0x119		GPT13_CMPD	Compare match D
0x11A		GPT13_CMPE	Compare match E
0x11B		GPT13_CMPF	Compare match F
0x11C		GPT13_OVF	Overflow
0x11D		GPT13_UDF	Underflow

Table 18.3 Association between event signal names set in .ELS[8:0] bits and signal numbers (6 of 7)

Event number	Interrupt request source	Name	Description
0x124	SCI0	SCI0_RXI*2	Receive data full
0x125		SCI0_TXI*2	Transmit data empty
0x126		SCI0_TEI*2	Transmit end
0x127		SCI0_ERI	Receive error
0x128		SCI0_AED	Effective edge detection
0x12A		SCI0_AM	Address match event
0x12B	SCI1	SCI1_RXI*2	Receive data full
0x12C		SCI1_TXI*2	Transmit data empty
0x12D		SCI1_TEI*2	Transmit end
0x12E		SCI1_ERI	Receive error
0x12F		SCI1_AED	Effective edge detection
0x131		SCI1_AM	Address match event
0x132	SCI2	SCI2_RXI*2	Receive data full
0x133		SCI2_TXI*2	Transmit data empty
0x134		SCI2_TEI*2	Transmit end
0x135		SCI2_ERI	Receive error
0x138		SCI2_AM	Address match event
0x139	SCI3	SCI3_RXI*2	Receive data full
0x13A		SCI3_TXI*2	Transmit data empty
0x13B		SCI3_TEI*2	Transmit end
0x13C		SCI3_ERI	Receive error
0x13F		SCI3_AM	Address match event
0x140	SCI4	SCI4_RXI*2	Receive data full
0x141		SCI4_TXI*2	Transmit data empty
0x142		SCI4_TEI*2	Transmit end
0x143		SCI4_ERI	Receive error
0x146		SCI4_AM	Address match event
0x163	SCI9	SCI9_RXI*2	Receive data full
0x164		SCI9_TXI*2	Transmit data empty
0x165		SCI9_TEI*2	Transmit end
0x166		SCI9_ERI	Receive error
0x169		SCI9_AM	Address match event
0x178	SPI0	SPI0_SPRI	Receive buffer full
0x179		SPI0_SPTI	Transmit buffer empty
0x17A		SPI0_SPII	Idle
0x17B		SPI0_SPEI	Error
0x17C		SPI0_SPCEND	Communication complete event

Table 18.3 Association between event signal names set in .ELS[8:0] bits and signal numbers (7 of 7)

Event number	Interrupt request source	Name	Description
0x17D	SPI1	SPI1_SPRI	Receive buffer full
0x17E		SPI1_SPTI	Transmit buffer empty
0x17F		SPI1_SPII	Idle
0x180		SPI1_SPEI	Error
0x181		SPI1_SPCEND	Transmission complete event
0x19D	I3C	I3C_RESP	Normal Response queue full
0x19E		I3C_CMD	Normal Command queue empty
0x19F		I3C_IBI	Normal IBI Queue Empty/Full
0x1A0		I3C_RX	Normal Rx Data buffer full
0x1A1		I3C_TX	Normal Tx Data buffer empty
0x1A2		I3C_RCV	Normal Receive Status queue full
0x1A3		I3C_HRESP	High priority response queue full
0x1A4		I3C_HCMD	High priority command queue empty
0x1A5		I3C_HRX	High priority Rx Data buffer full
0x1A6		I3C_HTX	High priority Tx Data buffer empty
0x1A7		I3C_TEND	Transmit end
0x1A8		I3C_EEI	Transfer error or event occurrence
0x1A9		I3C_STEV	Synchronous Timing
0x1AA		I3C_MREFOVF	MREF Counter Overflow
0x1AB		I3C_MREFCPT	MREF Capture
0x1AC		I3C_AMEV	Additional Master-initiated bus Event
0x1AE	ADC12	ADC120_ADI	A/D scan end interrupt
0x1AF		ADC120_GBADI	A/D scan end interrupt for group B
0x1B2		ADC120_WCMPM	Compare match
0x1B3		ADC120_WCMPUM	Compare mismatch
0x1B4		ADC121_ADI	A/D scan end interrupt
0x1B5		ADC121_GBADI	A/D scan end interrupt for group B
0x1B8		ADC121_WCMPM	Compare match
0x1B9		ADC121_WCMPUM	Compare mismatch
0x1BA	DOC	DOC_DOPCI	Data operation circuit interrupt

Note 1. Only pulse (edge detection) is supported.
Note 2. This event is not supported in FIFO mode.

18.2.4 ELCSARA : Event Link Controller Security Attribution Register A

Base address: ELC = 0x4020_1000
ELC_NS = 0x5020_1000

Offset address: 0xE0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	ELSEGR1	ELSEGR0	ELCR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ELCR	Event Link Controller Register Security Attribution Target register: ELCR 0: Secure 1: Non-secure	R/W
1	ELSEGR0	Event Link Software Event Generation Register 0 Security Attribution 0: Secure 1: Non-secure	R/W
2	ELSEGR1	Event Link Software Event Generation Register 1 Security Attribution 0: Secure 1: Non-secure	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1
Note: This register is write-protected by PRCR_S register.

This register specifies the security attribution for the ELCR, ELSEGR0 and ELSEGR1 registers.

18.2.5 ELCSARB : Event Link Controller Security Attribution Register B

Base address: ELC = 0x4020_1000
ELC_NS = 0x5020_1000

Offset address: 0xE4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	ELSR30	—	—	—	—	—	—	—	—	—	—	—	—	ELSR[17:16]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ELSR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
17:0	ELSR[17:0]	Event Link Setting Register n Security Attribution Target register: ELSRn (n = 0 to 17) 0: Secure 1: Non-secure	R/W
29:18	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
30	ELSR30	Event Link Setting Register 30 Security Attribution Target register: ELSR30 0: Secure 1: Non-secure	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1
Note: This register is write-protected by PRCR_S register.

This register specifies the security attribution for the Register ELSRn (n = 0 to 17, 30).

18.2.6 ELCPARA : Event Link Controller Privilege Attribution Register A

Base address: ELC = 0x4020_1000
ELC_NS = 0x5020_1000

Offset address: 0xF0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	ELSEGR1	ELSEGR0	ELCR
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	ELCR	Event Link Controller Register Privilege Attribution Target register: ELCR 0: Privileged 1: Unprivileged	R/W
1	ELSEGR0	Event Link Software Event Generation Register 0 Privilege Attribution Target register: ELSEGR0 0: Privileged 1: Unprivileged	R/W
2	ELSEGR1	Event Link Software Event Generation Register 1 Privilege Attribution Target register: ELSEGR1 0: Privileged 1: Unprivileged	R/W
31:3	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-2, P-TYPE-1
Note: This register is write-protected by PRCR_S and PRCR_NS register.

This register specifies the privilege attribution for the Register ELCR, ELSEGR0 and ELSEGR1 registers.

18.2.7 ELCPARB : Event Link Controller Privilege Attribution Register B

Base address: ELC = 0x4020_1000
ELC_NS = 0x5020_1000

Offset address: 0xF4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	ELSR30	—	—	—	—	—	—	—	—	—	—	—	—	ELSR[17:16]	
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ELSR[15:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
17:0	ELSR[17:0]	Event Link Setting Register n Privilege Attribution (n = 0 to 17) Target register: ELSRn (n = 0 to 17) 0: Privileged 1: Unprivileged	R/W
29:18	—	These bits are read as 1. The write value should be 1.	R/W
30	ELSR30	Event Link Setting Register 30 Privilege Attribution Target register: ELSR30 0: Privileged 1: Unprivileged	R/W
31	—	This bit is read as 1. The write value should be 1.	R/W

Note: S-TYPE-2, P-TYPE-1
Note: This register is write-protected by PRCR_S and PRCR_NS register.

This register specifies the privilege attribution for the Register ELSRn (n = 0 to 17, 30) registers.

18.3 Operation

18.3.1 Relation between Interrupt Handling and Event Linking

Event number for an event link is the same as that for the associated interrupt source. For information on generating event signals, see the explanation in the chapter for each event source module.

18.3.2 Linking Events

When an event occurs and that event is already set as a trigger in the Event Link Setting Register (ELSRn), the associated module is activated. The operation of the module must be set up in advance. Table 18.4 lists the operations of modules when an event occurs.

Table 18.4 Module operations when event occurs (1 of 2)

Module	Operations When Event is Input
GPT	• Start counting • Stop counting • Clear counting • Up counting • Down counting • Input capture
DAC12	Start D/A conversion

Table 18.4 Module operations when event occurs (2 of 2)

Module	Operations When Event is Input
I/O Ports	• Change pin output based on the EORR (reset) or EOSR (set) • Latch pin state to EIDR • The following ports can be used for the ELC: PORT 1 PORT 2 PORT 3 PORT 4
I3C	Start operation
ADC12	Start A/D conversion
DTC	Start DTC data transfer

18.3.3 Example of Procedure for Linking Events

To link events:

1. Set the operation of the module for which an event is to be linked.
2. Set the appropriate ELSRn.ELS[8:0] bits for the module to be linked.
3. Set the ELCR.ELCON bit to 1 to enable linkage of all events.
4. Configure the module from which an event is output and activate the module. The link between the two modules is now active.
5. To stop event linkage of modules individually, set 0 to the ELSRn.ELS[8:0] bit associated with the modules. To stop linkage of all the events, set the ELCR.ELCON bit to 0.

If event link output from the RTC is to be used, set the ELC after the RTC settings, for example, for initialization and time setting. Unintended events may be generated if RTC settings are made after the ELC settings.

If event link output from the PVD is to be used, set the ELC after setting the PVD. To disable the PVD, do so after setting 0x00 to the associated ELSRn register.

18.4 Usage Notes

18.4.1 Linking DMAC/DTC Transfer End Signals as Events

When linking the DMAC/DTC transfer end signals as events, do not set the same peripheral module as the DMAC/DTC transfer destination and event link destination. If set, the peripheral module might be started before DMAC/DTC transfer to the peripheral module is complete.

18.4.2 Setting Clocks

To link events, you must enable the ELC and the related modules. The modules cannot operate if the related modules are in the module-stop state or in low power mode in which the module is stopped (CPU Deep Sleep mode, Software Standby mode or Deep Software Standby mode). For more information, seeTable 18.3 and section 10, Low Power Modes.

18.4.3 Module-Stop Function Setting

The Module Stop Control Register C (MSTPCRC) can enable or disable ELC operation. The ELC is initially stopped after reset. Releasing the module-stop state enables access to the registers. The ELCON bit must be set to 0 before disabling ELC operation using the Module Stop Control Register. For more information, see Table 18.3 and section 10, Low Power Modes.

18.4.4 ELC Delay Time

In Figure 18.2, module A accesses module B through the ELC. There is a delay time in the ELC between module A and module B. Table 18.5 shows the ELC delay time.

Table 18.6 shows the clock domain of each module A. Table 18.7 shows the clock domain of each module B.

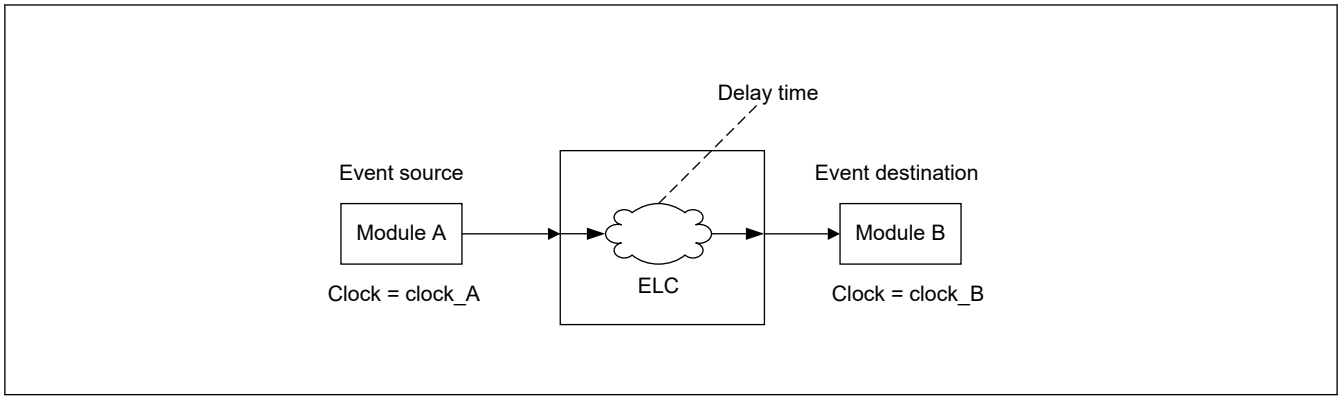


Figure 18.2 ELC delay time

Table 18.5 ELC delay time

Clock domain*1	Clock frequency	ELC delay time
clock_A = clock_B	clock_A = clock_B	0 cycle
clock_A ≠ clock_B	clock_A = clock_B	1 cycle to 2 cycles
	clock_A > clock_B	1 cycle to 2 cycles of clock_B
	clock_A < clock_B	1 cycle to 2 cycles of clock_A

Note 1. When PORT_IRQ, PVD, MOSC, DMAC, DTC, I/O Ports is selected as event source module, the ELC delay time is 2 cycles to 4 cycles of PCLKB.

Table 18.6 Clock domain of each module A

Module A	Clock domain of clock_A
PORT_IRQ, PVD, MOSC	ASYNC
DMAC, DTC, I/O Ports	ICLK
ULPT, AGT, IWDT, CWDT, IIC, ACMPHS, ELC, RTC	PCLKB
GPT	PCLKD
SCI, SPI, I3C, ADC12, DOC	PCLKA

Table 18.7 Clock domain of each module B

Module B	Clock domain of clock_B
ADC12, DAC12, I3C	PCLKA
DTC, I/O Ports	ICLK
GPT	PCLKD

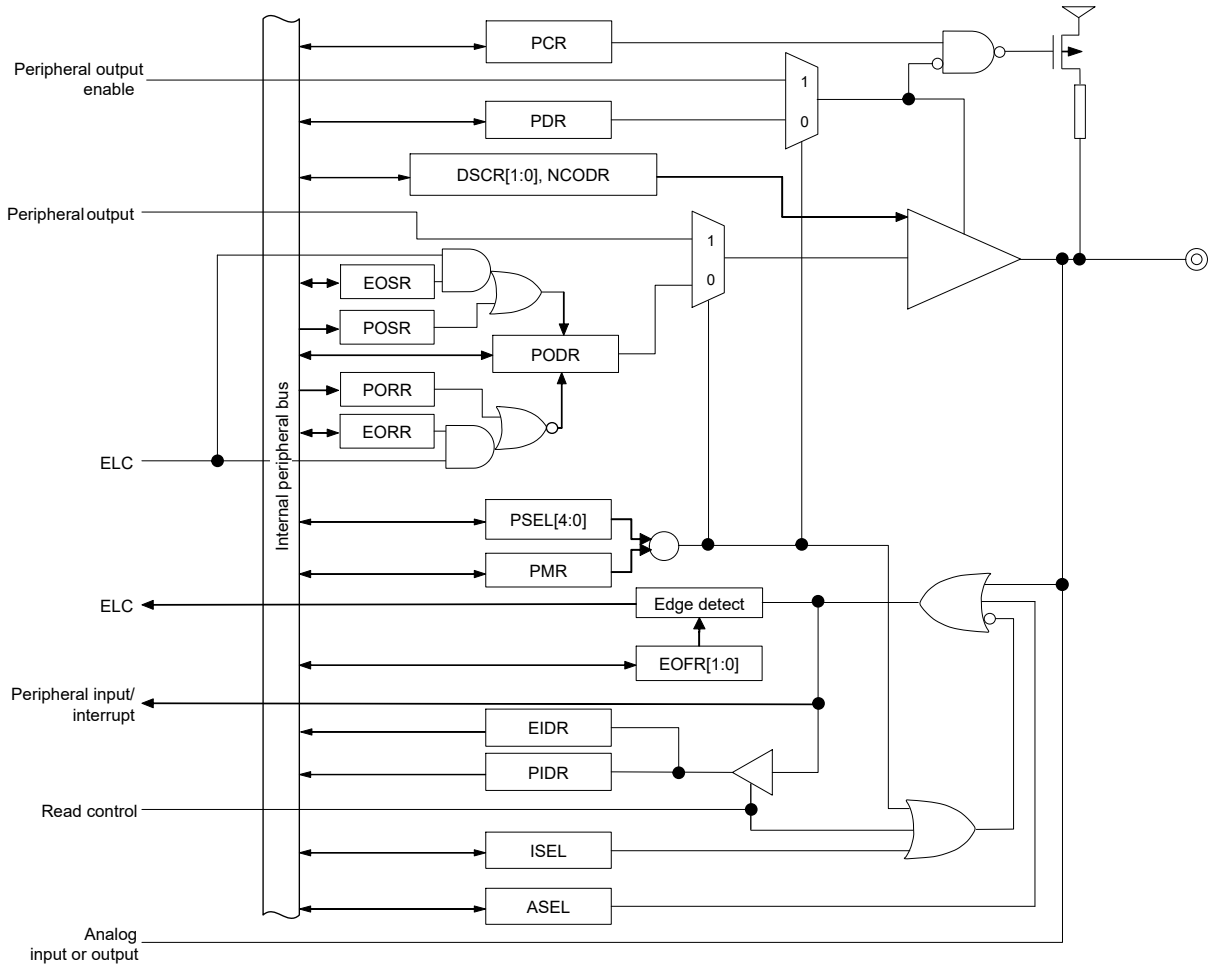
19. I/O Ports

19.1 Overview

The I/O port pins operate as general I/O port pins, I/O pins for peripheral modules, interrupt input pins, analog I/O, port group function for the ELC, or bus control pins.

All pins except P209 (as TDO of JTAG ports) operate as input pins immediately after a reset, and pin functions are switched by register settings. The I/O ports and peripheral modules for each pin are specified in the associated registers.

Figure 19.1 shows a connection diagram for the I/O port registers. The configuration of the I/O ports differs for different packages. Table 19.1 lists the I/O port specifications by package, and Table 19.2 lists the port functions.



Note: This figure shows a basic port configuration. The configuration differs depending on the ports.

Figure 19.1 Connection diagram for I/O port registers

Table 19.1 I/O port specifications (1 of 3)

Port	Package		Package	
	224 pins	Number of pins	224 pins (without MIPI)	Number of pins
PORT0	P000 to P011, P014, P015	14	P000 to P011, P014, P015	14
PORT1	P100 to P107, P112 to P115	12	P100 to P107, P112 to P115	12
PORT2	P200, P201, P206 to P213	10	P200 to P213	14

Table 19.1 I/O port specifications (2 of 3)

Port	Package 224 pins	Number of pins	Package 224 pins (without MIPI)	Number of pins
PORT3	P300 to P312	13	P300 to P315	16
PORT4	P400 to P415	16	P400 to P415	16
PORT5	P500 to P515	16	P500 to P515	16
PORT6	P600 to P607, P609 to P615	15	P600 to P607, P609 to P615	15
PORT7	P700 to P715	16	P700 to P715	16
PORT8	P800 to P815	16	P800 to P815	16
PORT9	P902 to P915	14	P900 to P915	16
PORTA	PA00 to PA15	16	PA00 to PA15	16
PORTB	PB00 to PB07	8	PB00 to PB07	8

Table 19.1 I/O port specifications (3 of 3)

Port	Package 176 pins	Number of pins	Package 176 pins (without MIPI)	Number of pins
PORT0	P000 to P010, P014, P015	13	P000 to P010, P014, P015	13
PORT1	P100 to P107, P112 to P115	12	P100 to P107, P112 to P115	12
PORT2	P200, P201, P206, P208 to P213	9	P200 to P206, P208 to P213	13
PORT3	P300 to P312	13	P300 to P315	16
PORT4	P400 to P415	16	P400 to P415	16
PORT5	P500 to P502, P511 to P513	6	P500 to P502, P511 to P513	6
PORT6	P600 to P607, P609 to P615	15	P600 to P607, P609 to P615	15
PORT7	P700 to P710	11	P700 to P710	11
PORT8	P800 to P806, P808 to P812. P814, P815	14	P800 to P806, P808 to P812. P814, P815	14
PORT9	P905 to P909	5	P900, P901, P905 to P909	7
PORTA	PA00, PA01, PA08 to PA10	5	PA00, PA01, PA08 to PA10	5
PORTB	PB00, PB01	2	PB00, PB01	2

Table 19.2 I/O port functions (1 of 2)

Port	Port name	Input pull-up	Open- drain output	Drive capability switching	5Vtolerant	I/O	Power supply
PORT0	P000 to P011, P014, P015	✓	✓	Low	—	Input / Output	AVCC0
PORT1	P100 to P103	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC2
	P104 to P107	✓	✓	Low, middle, high	—	Input / Output	VCC2
	P112 to P115	✓	✓	Low, middle, high	—	Input / Output	VCC
PORT2	P200	✓	—	—	—	Input	VCC
	P201	✓	✓	Low	—	Input / Output	VCC
	P202 to P204, P207 to P213	✓	✓	Low, middle, high	—	Input / Output	VCC
	P205, P206	✓	✓	Low, middle, high	✓	Input / Output	VCC

Table 19.2 I/O port functions (2 of 2)

Port	Port name	Input pull-up	Open- drain output	Drive capability switching	5Vtolerant	I/O	Power supply
PORT3	P300 to P303, P309 to P315	✓	✓	Low, middle, high	—	Input / Output	VCC
	P304 to P308	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC
PORT4	P400, P401, P405	✓	✓	Low, middle, high	—	Input / Output	VCC
	P402 to P404, P406 to P415	✓	✓	Low, middle, high	✓	Input / Output	VCC
PORT5	P500 to P502	✓	✓	Low, middle, high	—	Input / Output	VCC2
	P503 to P510, P513 to P515	✓	✓	Low, middle, high	—	Input / Output	VCC
	P511, P512	✓	✓	Low, middle, high	✓	Input / Output	VCC
PORT6	P600	✓	✓	Low, middle, high	—	Input / Output	VCC2
	P601 to P607, P609 to P615	✓	✓	Low, middle, high	—	Input / Output	VCC
PORT7	P700 to P708	✓	✓	Low, middle, high	—	Input / Output	VCC
	P709 to P715	✓	✓	Low, middle, high	✓	Input / Output	VCC
PORT8	P800 to P804, P808 to P810	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC2
	P805 to P807, P813 to P815	✓	✓	Low, middle, high	—	Input / Output	VCC
	P811, P812	✓	✓	Low, middle, high	—	Input / Output	VCC2
PORT9	P900 to P915	✓	✓	Low, middle, high	—	Input / Output	VCC
PORTA	PA00 to PA08, PA10 to PA15	✓	✓	Low, middle, high	—	Input / Output	VCC
	PA09	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC
PORTB	PB00, PB02 to PB07	✓	✓	Low, middle, high	—	Input / Output	VCC
	PB01	✓	✓	Low, middle, high	✓	Input / Output	VCC

Note: ✓: Available
—: Setting prohibited

19.2 Register Descriptions

19.2.1 PCNTR1/PODR/PDR : Port Control Register 1

Base address: PORTm = 0x4040_0000 + 0x0020 × m (m = 0 to 9, A, B)
PORTm_NS = 0x5040_0000 + 0x0020 × m (m = 0 to 9, A, B)

Offset address: 0x000 (PCNTR1/PDR)
0x002 (PODR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PODR15	PODR14	PODR13	PODR12	PODR11	PODR10	PODR09	PODR08	PODR07	PODR06	PODR05	PODR04	PODR03	PODR02	PODR01	PODR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PDR15	PDR14	PDR13	PDR12	PDR11	PDR10	PDR09	PDR08	PDR07	PDR06	PDR05	PDR04	PDR03	PDR02	PDR01	PDR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PDR15 to PDR00	Pmn Direction 0: Input (functions as an input pin) 1: Output (functions as an output pin)	R/W
31:16	PODR15 to PODR00	Pmn Output Data 0: Low output 1: High output	R/W

Note: S-TYPE-4, P-TYPE-5
Note: m = 0 to 9, A, B, n = 00 to 15

The Port Control Register 1 (PCNTR1/PODR/PDR) is a 32-bit or 16-bit read/write register that controls port direction and port output data. The PCNTR1 specifies the port direction and output data, and is accessed in 32-bit units. The PDRn (bits [15:0] in PCNTR1) and PODRn (bits [31:16] in PCNTR1) respectively, are accessed in 16-bit units.

PDRn bits (Pmn Direction)

The PDRn bits select the input or output direction for individual pins on the associated port when the pins are configured as general I/O pins. Each pin on port m is associated with a PORTm.PCNTR1.PDRn bit. The I/O direction can be specified in 1-bit unit. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0. In the case of input only ports, PDRn bits are reserved. See section 19.1. Overview. The PDRn bit in the PORTm.PCNTR1 register serves the same function as the PDR bit in the PFS.PmnPFS register.

PODRn bits (Pmn Output Data)

The PODRn bits hold data to be output from the general I/O pins. Bits of non-existent port m are reserved. Reserved bits are read as 0. The write value should be 0. In the case of input only ports, PODRn bits are reserved. See section 19.1. Overview. The PODRn bit in the PORTm.PCNTR1 register serves the same function as the PODR bit in the PFS.PmnPFS register.

19.2.2 PCNTR2/EIDR/PIDR : Port Control Register 2

Base address: PORTm = 0x4040_0000 + 0x0020 × m (m = 0 to 9, A, B)
PORTm_NS = 0x5040_0000 + 0x0020 × m (m = 0 to 9, A, B)

Offset address: 0x004 (PCNTR2/PIDR)
0x006 (EIDR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EIDR15	EIDR14	EIDR13	EIDR12	EIDR11	EIDR10	EIDR09	EIDR08	EIDR07	EIDR06	EIDR05	EIDR04	EIDR03	EIDR02	EIDR01	EIDR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PIDR15	PIDR14	PIDR13	PIDR12	PIDR11	PIDR10	PIDR09	PIDR08	PIDR07	PIDR06	PIDR05	PIDR04	PIDR03	PIDR02	PIDR01	PIDR00
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
15:0	PIDR15 to PIDR00	Pmn State 0: Low level 1: High level	R
31:16	EIDR15 to EIDR00 *2	Port Event Input Data*1 When an ELC_PORTx signal occurs 0: Low input 1: High input	R

Note: S-TYPE-4, P-TYPE-5
Note: m = 0 to 9, A, B, n = 00 to 15
Note 1. x = 1, 2, 3 or 4 for EIDR only
Note 2. Supported by ports 1, 2, 3 or 4.

The Port Control Register 2 (PCNTR2/EIDR/PIDR) allows read access to the Pmn state and the port event input data using 32-bit or 16-bit access.

The PCNTR2 represents the Pmn state and the port event input data, and is accessed in 32-bit units.

The PIDRn (bits [15:0] in PCNTR2) and EIDRn (bits [31:16] in PCNTR2) respectively, are accessed in 16-bit units. Bits associated with non-existent pins are reserved. Reserved bits are read as undefined.

PIDRn bits (Pmn State)

The PIDRn bits reflect the individual pin states of the port, regardless of the values set in PmnPFS.PMR and PORTm.PCNTR1.PDRn. The PIDRn bit in the PORTm.PCNTR2 register serves the same function as the PIDR bit in the PFS.PmnPFS register.

A pin state cannot be reflected in PIDRn when one of the following functions is enabled:

- RTC Time Capture input and tamper detection (RTCIC)
- CS area controller (CSC)
- Analog function (ASEL = 1)
- Trace interface (TCLK, TDATAn (n = 0 to 3), SWO)

EIDRn bits (Port Event Input Data)

The EIDRn bits latch a pin state when an ELC_PORTx signal occurs. Pin states can only be input to EIDRn when PmnPFS.PMR and PORTm.PCNTR1.PDRn are 0. When the PmnPFS.ASEL bit is set to 1, the associated pin state is not reflected in EIDRn.

19.2.3 PCNTR3/PORR/POSR : Port Control Register 3

Base address: PORTm = 0x4040_0000 + 0x0020 × m (m = 0 to 9, A, B)
PORTm_NS = 0x5040_0000 + 0x0020 × m (m = 0 to 9, A, B)

Offset address: 0x008 (PCNTR3/POSR)
0x00A (PORR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PORR 15	PORR 14	PORR 13	PORR 12	PORR 11	PORR 10	PORR 09	PORR 08	PORR 07	PORR 06	PORR 05	PORR 04	PORR 03	PORR 02	PORR 01	PORR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	POSR 15	POSR 14	POSR 13	POSR 12	POSR 11	POSR 10	POSR 09	POSR 08	POSR 07	POSR 06	POSR 05	POSR 04	POSR 03	POSR 02	POSR 01	POSR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	POSR15 to POSR00	Pmn Output Set 0: No effect on output 1: High output	W
31:16	PORR15 to PORR00	Pmn Output Reset 0: No effect on output 1: Low output	W

Note: S-TYPE-4, P-TYPE-5
Note: m = 0 to 9, A, B, n = 00 to 15

The Port Control Register 3 (PCNTR3/PORR/POSR) is a 32-bit or 16-bit write register that controls the setting or resetting of the port output data.

The PCNTR3 controls the setting or resetting of the port output data, and is accessed in 32-bit units.

The POSRn (bits [15:0] in PCNTR3) and the PORRn (bits [31:16] in PCNTR3) respectively, are accessed in 16-bit units.

POSRn bits (Pmn Output Set)

POSR changes PODR when set by a software write. For example, for P100, when PORT1.PCNTR3.POSR00 = 1, PORT1.PCNTR1.PODR00 outputs 1. Bits associated with non-existent pins are reserved. The write value should always be 0. In the case of input only ports, POSRn bits are reserved. See section 19.1. Overview.

PORRn bits (Pmn Output Reset)

PORR changes PODR when reset by a software write. For example, for P100, when PORT1.PCNTR3.PORR00 = 1, PORT1.PCNTR1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value should always be 0. In the case of input only ports, PORRn bits are reserved. See section 19.1. Overview.

Note: When EORRn or EOSRn is set, writing is prohibited to PODRn, PORRn, and POSRn.
Note: PORRn and POSRn should not be set at the same time.

19.2.4 PCNTR4/EORR/EOSR : Port Control Register 4

Base address: PORTm = 0x4040_0000 + 0x0020 × m (m = 1 to 4)
PORTm_NS = 0x5040_0000 + 0x0020 × m (m = 1 to 4)

Offset address: 0x00C (PCNTR4/EOSR)
0x00E (EORR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EORR 15	EORR 14	EORR 13	EORR 12	EORR 11	EORR 10	EORR 09	EORR 08	EORR 07	EORR 06	EORR 05	EORR 04	EORR 03	EORR 02	EORR 01	EORR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EOSR 15	EOSR 14	EOSR 13	EOSR 12	EOSR 11	EOSR 10	EOSR 09	EOSR 08	EOSR 07	EOSR 06	EOSR 05	EOSR 04	EOSR 03	EOSR 02	EOSR 01	EOSR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	EOSR15 to EOSR00	Pmn Event Output Set When an ELC_PORTx signal occurs 0: No effect on output 1: High output	R/W
31:16	EORR15 to EORR0	Pmn Event Output Reset When an ELC_PORTx signal occurs 0: No effect on output 1: Low output	R/W

Note: S-TYPE-4, P-TYPE-5
Note: m = 1 to 4, n = 00 to 15, x = 1 to 4

The Port Control Register 4 (PCNTR4/EORR/EOSR) is a 32-bit or 16-bit read/write register that controls the setting or resetting of the port output data by an event input from the ELC.

The PCNTR4 controls the setting or resetting of the port output data by an event input from the ELC, and is accessed in 32-bit units.

The EOSRn (bits [15:0] in PCNTR4) and EORRn (bits [31:16] in PCNTR4) respectively, are accessed in 16-bit units.

EOSRn bits (Pmn Event Output Set)

EOSR changes PODR when set because an ELC_PORTx signal occurs. For example, for P100 if PORT1.PCNTR4.EOSR00 is set to 1 when the ELC_PORTx occurs, PORT1.PCNTR1.PODR00 outputs 1. Bits associated with non-existent pins are reserved. The write value should always be 0. For input only ports, EOSRn bits are reserved. See section 19.1. Overview.

EORRn bits (Pmn Event Output Reset)

EORR changes PODR when reset because an ELC_PORTx signal occurs. For example, for P100 if PORT1.PCNTR4.EORR00 = 1 when the ELC_PORTx occurs, PORT1.PCNTR1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value should always be 0. For input only ports, EORRn bits are reserved. See section 19.1. Overview.

Note: When EORRn or EOSRn is set, writing is prohibited to PODRn, PORRn, and POSRn.

Note: EORRn and EOSRn should not be set at the same time.

19.2.5 PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0 to 9, A, B, n = 00 to 15)

Base address: PFS = 0x4040_0800
PFS_NS = 0x5040_0800

Offset address: 0x000 + 0x040 × m + 0x004 × n (PmnPFS/PmnPFS_HA/PmnPFS_BY)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	PSEL[4:0]					—	—	—	—	—	—	—	PMR
Value after reset:	0	0	0	0*1	0	0	0*1	0*1	0	0	0	0	0	0	0	0*1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ASEL	ISEL	EOFR[1:0]		DSCR[1:0]		—	—	—	NCODR	—	PCR	—	PDR	PIDR	PODR
Value after reset:	0	0	0	0	0	0*1	0	0	0	0	0	0*1	0	0	x	0

Bit	Symbol	Function	R/W
0	PODR	Port Output Data 0: Low output 1: High output	R/W
1	PIDR	Pmn State 0: Low level 1: High level	R
2	PDR	Port Direction 0: Input (functions as an input pin) 1: Output (functions as an output pin)	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	PCR	Pull-up Control 0: Disable input pull-up 1: Enable input pull-up	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	NCODR	N-Channel Open-Drain Control 0: CMOS output 1: NMOS open-drain output	R/W
9:7	—	These bits are read as 0. The write value should be 0.	R/W
11:10	DSCR[1:0]	Port Drive Capability 0 0: Low drive 0 1: Middle drive 1 0: High speed high drive 1 1: High drive	R/W
13:12	EOFR[1:0]	Event on Falling/Event on Rising*2 0 0: Don't care 0 1: Detect rising edge 1 0: Detect falling edge 1 1: Detect both edges	R/W
14	ISEL	IRQ Input Enable 0: Not used as an IRQn input pin 1: Used as an IRQn input pin	R/W
15	ASEL	Analog Input Enable 0: Not used as an analog pin 1: Used as an analog pin	R/W
16	PMR	Port Mode Control 0: Used as a general I/O pin 1: Used as an I/O port for peripheral functions	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
28:24	PSEL[4:0]	Peripheral Select These bits select the peripheral function. For individual pin functions, see the associated tables in this chapter.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note 1. The initial value of P201, P208, P209. P210, P211, P814 and P815 is not 0x0000_0000. P201 is 0x0000_0010, P208 is 0x0001_0010, P209 is 0x0001_0400, P210 is 0x0001_0410, P211 is 0x0001_0010, P814 is 0x1301_0000 and P815 is 0x1301_0000.
Note 2. Supported by PORTn (n = 1 to 4).

Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) is a 32-bit, 16-bit, or 8-bit read/write control register that selects the port mn pin function. PmnPFS is accessed in 32-bit units. PmnPFS_HA (PmnPFS [15:0] bits) is accessed in 16-bit units. PmnPFS_BY (PmnPFS[7:0] bits) is accessed in 8-bit units.

The available Port mn pin depends on the product. For details, see Table 19.1

PODR bit (Port Output Data), PIDR bit (Port State), PDR bit (Port Direction)

The PDR, PIDR, and PODR bits serve the same function as the PCNTR. When these bits are read, the PCNTR value is read.

PCR bit (Pull-up Control)

The PCR bit enables or disables an input pull-up resistor on the individual port pins. When a pin is in the input state with the associated bit in PmnPFS.PCR set to 1, the pull-up resistor connected to the pin is enabled. When a pin is set as a general port output pin, or a peripheral function output pin, the pull-up resistor for the pin is disabled regardless of the PCR setting. The pull-up resistor is also disabled in the reset state. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

NCODR bit (N-Channel Open-Drain Control)

The NCODR bit specifies the output type for the port pins. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

DSCR[1:0] bits (Port Drive Capability)

The DSCR[1:0] bits switches the capability of the port. If the capability of a pin is fixed, the associated bit is a read/write bit, but the capability cannot be changed. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

EOFR[1:0] bits (Event on Falling/Event on Rising)

The EOFR[1:0] bits select the edge detection method for the port group input signal. These bits support rising, falling, or both edge detections. When the EOFR[1:0] bits are set to 01b, 10b, or 11b, the input enable of the I/O cell is asserted. Following that, the event pulse is input from the external pin, and the GPIO outputs the event pulse to the ELC. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

ISEL bit (IRQ Input Enable)

The ISEL bit specifies IRQ input pins. This setting can be used in combination with the peripheral functions, although an IRQn (external pin interrupt) of the same number must only be enabled for one pin. The ISEL bit for an unspecified IRQn is reserved.

ASEL bit (Analog Input Enable)

The ASEL bit specifies analog pins. When a pin is set as an analog pin by this bit:

1. Specify it as a general I/O port in the Port Mode Control bit (PmnPFS.PMR).
2. Disable the pull-up resistor in the Pull-up Control bit (PmnPFS.PCR).
3. Specify the input in the Port Direction bit (PmnPFS.PDR). The pin state cannot be read at this point. The PmnPFS register is protected by the Write-Protect Register for Non-secure(PWPR_NS) . Release write-protect before modifying the register.

The ASEL bit for an unspecified analog I/O pin is reserved.

PMR bit (Port Mode Control)

The PMR bit specifies the port pin function. Bits associated with non-existent pins are reserved. The write value should be 0.

PSEL[4:0] bits (Peripheral Select)

The PSEL[4:0] bits assign the peripheral function.

19.2.6 PWPR_NS : Write-Protect Register for Non-secure

Base address: PFS_NS = 0x5040_0800

Offset address: 0x50C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	B0WI	PFSWE	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	PFSWE	PmnPFS Register Write Enable 0: Writing to the PmnPFS register is disabled 1: Writing to the PmnPFS register is enabled	R/W
7	B0WI	PFSWE Bit Write Disable 0: Writing to the PFSWE bit is enabled 1: Writing to the PFSWE bit is disabled	R/W

Note: S-TYPE-7, P-TYPE-2
Note: The access to this register is not controlled by any security attribution register.

PFSWE bit (PmnPFS Register Write Enable)

Writing to the PmnPFS register is enabled only when the PFSWE bit is set to 1. You must first write 0 to the B0WI bit before setting PFSWE to 1.

B0WI bit (PFSWE Bit Write Disable)

Writing to the PFSWE bit is enabled only when the B0WI bit is set to 0.

19.2.7 PFENET : Ethernet Control Register

Base address: PFS = 0x4040_0800
PFS_NS = 0x5040_0800

Offset address: 0x500

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PHYMODE0	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	PHYMODE0	Ethernet Mode Setting ch0 0: RMII mode (ETHERC channel 0) 1: MII mode (ETHERC channel 0)	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: The access to PHYMODE0 bit is controlled by PSARB.PSARB15 bit and PPARB.PPARB15 bit.

PHYMODE0 bit (Ethernet Mode Setting ch0)

The PHYMODE0 bit specifies the PHY mode of ETHERC channel 0. Select the same mode as that specified in the pin function select bits (PmnPFS.PSEL[4:0]). When the signals for the RMII mode are specified in the PmnPFS.PSEL[4:0] bits, set the PHYMODE bit to 0 (RMII mode). When the signals for the MII mode are specified in the PmnPFS.PSEL[4:0] bits, set the PHYMODE bit to 1 (MII mode).

19.2.8 PWPR_S : Write-Protect Register for Secure

Base address: PFS = 0x4040_0800

Offset address: 0x514

Bit position:	7	6	5	4	3	2	1	0
Bit field:	B0WI	PFSWE	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	PFSWE	PmnPFS Register Write Enable 0: Disable writes to the PmnPFS register 1: Enable writes to the PmnPFS register	R/W
7	B0WI	PFSWE Bit Write Disable 0: Enable writes the PFSWE bit 1: Disable writes to the PFSWE bit	R/W

Note: S-TYPE-6, P-TYPE-2

PFSWE bit (PmnPFS Register Write Enable)

Writing to the PmnPFS register of the IO port pin set as secure by the PmSAR register is enabled only when the PFSWE bit is set to 1. You must first write 0 to the B0WI bit before setting PFSWE to 1.

B0WI bit (PFSWE Bit Write Disable)

Writing to the PFSWE bit is enabled only when the B0WI bit is set to 0.

19.2.9 PmSAR : Port Security Attribution register (m = 0 to 9, A, B)

Base address: PFS = 0x4040_0800
PFS_NS = 0x5040_0800

Offset address: 0x530 + 0x004 × m

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PMNSA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PMNSA[15:0]	Pmn Security Attribution Target I/O port pin : Pmn 0: Secure 1: Non-secure	R/W

Note: S-TYPE-1, P-TYPE-1
Note: This register is write-protected by PRCR register.
Note: m = 0 to 9, A, B, n = 00 to 15

Port Security Attribution Register is a 16-bit register that setting the Security Attribution of the each port, the registers are accessed only in 16-bit units.

PMNSA[15:0] bits (Pmn Security Attribution)

The PmnSA bit specifies the Security Attribution of Pmn.

19.2.10 LVOCR : Low Voltage Operation Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xAB0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	LVO1E	LVO0E
Value after reset:	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	LVO0E	Low Voltage Operation 0 Enable 0: Disable 1: Enable	R/W
1	LVO1E	Low Voltage Operation 1 Enable 0: Disable 1: Enable	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Low voltage operation control register controls the IO when VCC/VCC2 is lower than 2.7V.

LVO0E bit (Low Voltage Operation 0 Enable)

When VCC is lower than 2.7V, set the LVO0E bit to 1.

LVO1E bit (Low Voltage Operation 1 Enable)

When VCC2 is lower than 2.7V, set the LVO1E bit to 1.

19.3 Operation

19.3.1 General I/O Ports

All pins except P208 to P211, P814 and P815 operate as general I/O ports after reset. General I/O ports are organized as 16 bits per port and can be accessed by port with the Port Control Registers (PCNTRn, where n = 1 to 4), or by individual pins with the Port mn Pin Function Select register. For details on these registers, see section 19.2. Register Descriptions.

Each port has the following bits:

- Port Security Attribution register (PmSAR)(m = 0 to 9, A, B), which indicates the security attribution.
- Port Direction bit (PDRn), which selects input or output direction
- Port Output Data bit (PODRn), which holds data for output
- Port Input Data bit (PIDRn), which indicates the pin states
- Event Input Data bit (EIDRn), which indicates the pin state when an ELC_PORTn (n = 1, 2, 3 or 4) signal occurs
- Port Output Set bit (POSRn), which indicates the output value when a software write occurs
- Port Output Reset bit (PORRn), which indicates the output value when a software write occurs
- Event Output Set bit (EOSRn), which indicates the output value when an ELC_PORTn (n = 1, 2, 3 or 4) signal occurs
- Event Output Reset bit (EORRn), which indicates the output value when an ELC_PORTn (n = 1, 2, 3 or 4) signal occurs.

19.3.2 Port Function Select

The following port functions are available for configuring each pin:

- Security function: Security attribution for each pins
- I/O configuration: CMOS output or NMOS open-drain output, pull-up control, and drive capability
- General I/O port: Port direction, output data setting, and read input data
- Alternate function: Configured function mapping to the pin.

Each pin is associated with a Port mn Pin Function Select register (PmnPFS), which includes the associated PODR, PIDR, and PDR bits. In addition, the PmnPFS register includes the following:

- PCR: Pull-up resistor control bit that turns the input pull-up MOS on or off
- NCODR: N-channel open-drain control bit that selects the output type for each pin
- DSCR[1:0]: Drive capability control bit that selects the drive capability
- EOFR[1:0]: For selecting the edge of the event that input from the port group
- ISEL: IRQ input enable bit to specify an IRQ input pin
- ASEL: Analog input enable bit to specify an analog pin
- PMR: Port mode bit to specify the pin function of each port
- PSEL[4:0]: Port function select bits to select the associated peripheral function.

These configurations can be made by a single-register access to the Port mn Pin Function Select register. For details, see section 19.2.5. PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0 to 9, A, B, n = 00 to 15).

19.3.3 Port Group Function for ELC

In the MCU, Port 1 are assigned for the ELC port group function.

19.3.3.1 Behavior When ELC_PORTn (n = 1, 2, 3 or 4) is Input from ELC

The MCU supports the two functions described in this section when an ELC_PORTn (n = 1, 2, 3 or 4) signal comes from the ELC.

(1) Input to EIDR

For the GPI function (PDR = 0 and PMR = 0 in the PmnPFS register), when an ELC_PORTn (n = 1, 2, 3 or 4) signal comes from the ELC, the input enable of the I/O cell is asserted, and data from the external pins is read into the EIDR bit. See Figure 19.2

For the GPO function (PDR = 1) or the peripheral mode (PMR = 1), 0 is input into the EIDR bit from the external pins.

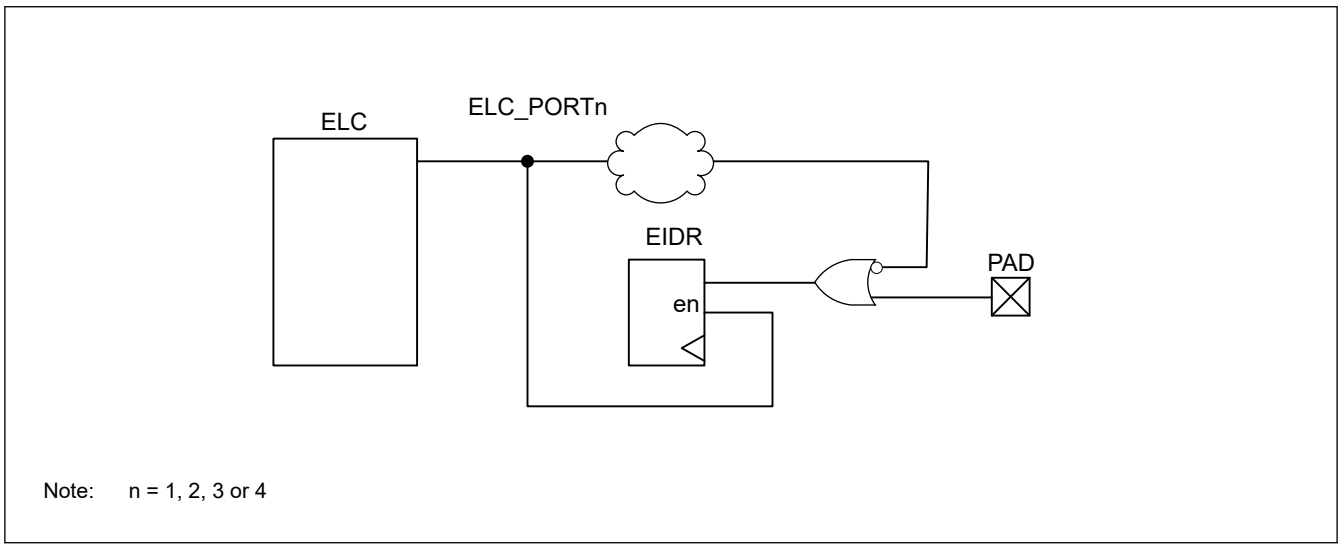


Figure 19.2 Event ports input data

(2) Output from PODR by EOSR and EORR

When an ELC_PORTn (n = 1, 2, 3 or 4) signal occurs, the data is output from the PODR to the external pin based on the settings in the EOSR and EORR registers.

- If EOSR is set to 1, when an ELC_PORTn (n = 1, 2, 3 or 4) signal occurs, the PODR register outputs 1 to the external pin. Otherwise, when EOSR = 0, the PODR value is retained.
- If EORR is set to 1, when ELC_PORTn (n = 1, 2, 3 or 4) signal occurs, the PODR register outputs 0 to the external pin. Otherwise, when EORR = 0, the PODR value is retained.

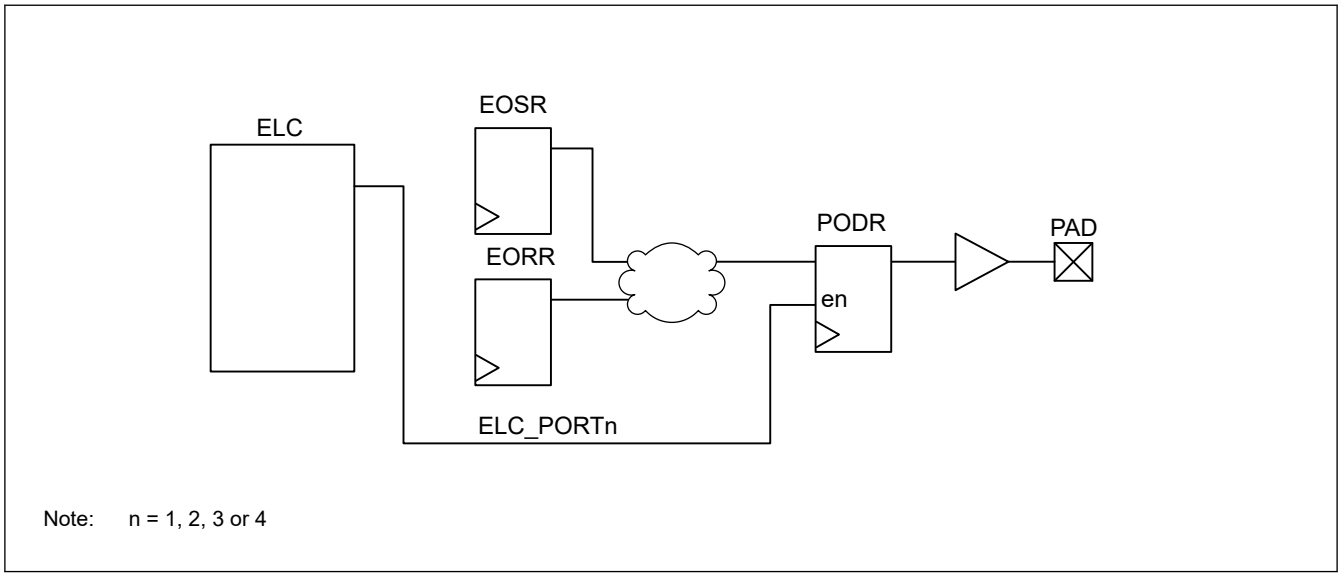


Figure 19.3 Event ports output data

19.3.3.2 Behavior When an Event Pulse is Output to ELC

To output the event pulse from the external pins to the ELC, set the EOFR[1:0] bits in the PmnPFS register. For details, see section 19.2.5. PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0 to 9, A, B, n = 00 to 15). When the EOFR[1:0] bits are set, the input enable of the I/O cell is asserted.

Data from the external pin is the input. For example, for Port 1, when the data is input from P100 to P115, the data of those 16 pins is organized by OR logic. This data is formed into a one-shot pulse that goes to the ELC. The operation of Port n (n = 2 to 4) is also the same as Port 1. See Figure 19.4.

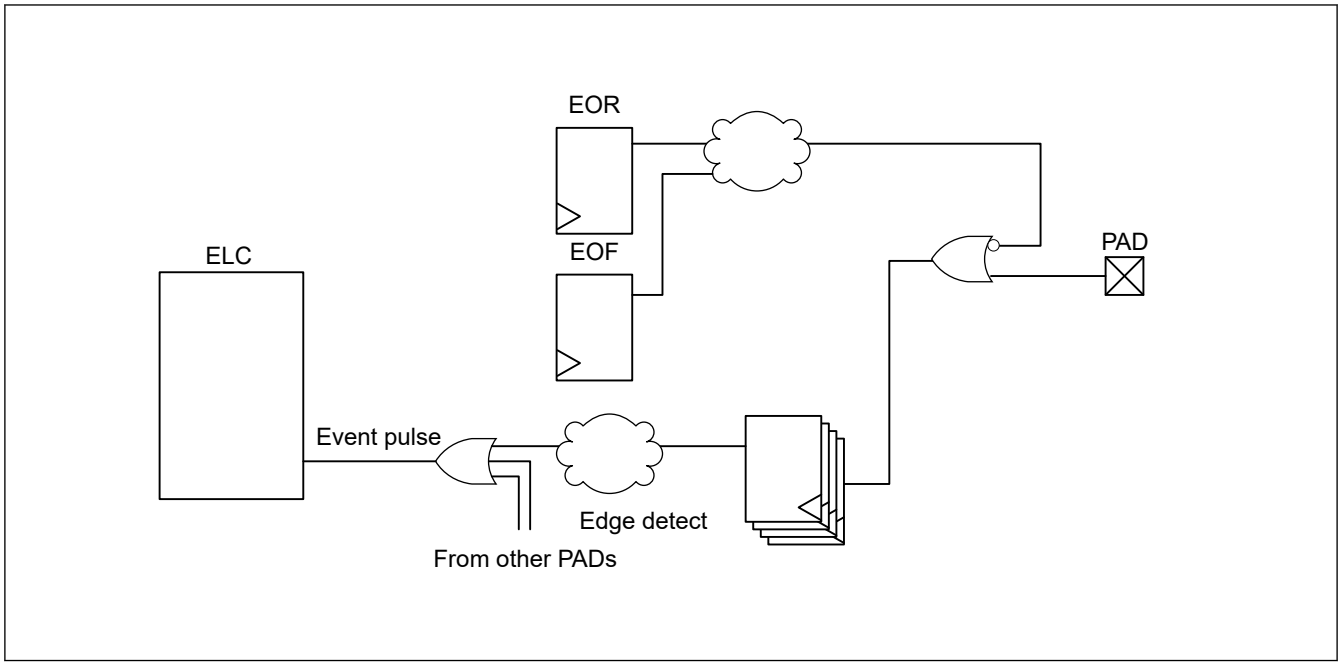


Figure 19.4 Generation of event pulse

19.4 Handling of Unused Pins

Table 19.3 shows how to handle unused pins.

Table 19.3 Handling of unused pins (1 of 2)

Pin name	Description
MD	Use as a mode selection pin
RES	Connect to VCC through a resistor
USB_DP, USB_DM	When both P814PFS.PMR and P815PFS.PMR bits are set to 1, keep these pins open. When P814PFS.PMR or P815PFS.PMR bit is set to 0, configure it in the same way as Other ports.
P200/NMI	Connect to VCC through a resistor
EXTAL	When the main clock oscillator is not used, set the MOSCCR.MOSTP bit to 1 (general port P212). When this pin is not used as port P212, configure it in the same way as ports 1 to B.
XTAL	When the main clock oscillator is not used, set the MOSCCR.MOSTP bit to 1 (general port P213). When the external clock is input to the EXTAL pin, the XTAL pin functions as P213. When this pin is not used as port P213, configure it in the same way as ports 1 to B.
XCIN	Connect to VSS through a resistor
XCOUT	Keep pin open
P000 to P015	• If the direction is set to input (PCNTR1.PDRn = 0), connect the associated pin to AVCC0 through a resistor or to AVSS0 through a resistor *1 • If the direction is set to output (PCNTR1.PDRn = 1), keep pin open *1
P100 to P107, P500 to P502, P600, P800 to P804, P808 to P812	• If the direction is set to input (PCNTR1.PDRn = 0), connect the associated pin to VCC2 through a resistor or to VSS through a resistor *1 • If the direction is set to output (PCNTR1.PDRn = 1), keep pin open *1
Other ports	• If the direction is set to input (PCNTR1.PDRn = 0), connect the associated pin to VCC through a resistor or to VSS through a resistor *1 *2 • If the direction is set to output (PCNTR1.PDRn = 1), keep pin open *1 *3
USBHS_DP, USBHS_DM, USBHS_RREF	• Preconditions: – AVCC_USBHS = VCC_USBHS: Connect to VCC – VSS1_USBHS = VSS2_USBHS: Connect to VSS – Set the module-stop state for USBHS (MSTPCRB.MSTPB12 = 1) • Processing details: – USBHS_DP, USBHS_DM, and USBHS_RREF: Open

Table 19.3 Handling of unused pins (2 of 2)

Pin name	Description
VREFH0, VREFH	Connect to AVCC0
VREFL0, VREFL	Connect to AVSS0
MIPI_DL0_N, MIPI_DL1_N, MIPI_DL0_P, MIPI_DL1_P, MIPI_CL_N, MIPI_CL_P	• Preconditions: – VCC18_MIPI: Open – AVCC_MIPI: Connect to VCC – VSS_MIPI: Connect to VSS – Set the module-stop state for MIPI DSI (MSTPCRC.MSTPC10 = 1) • Processing details: – MIPI_DL0_N, MIPI_DL1_N, MIPI_DL0_P, MIPI_DL1_P, MIPI_CL_N and MIPI_CL_P: Open
VLO	Keep pin open*4
VBATT	Connect to VCC

Note 1. Clear the PmnPFS.PMR, PmnPFS.ISEL, PmnPFS.PCR, and PmnPFS.ASEL bits to 0.
Note 2. P208, P210 and P211 are recommended for pull up VCC through a resistor, because these pins are input pull-up enabled from the initial value (PmnPFS.PCR = 1).
Note 3. P209 is recommended for setting the direction to output (PCNTR1.PDRn = 1), because this pin is output from the initial value.
Note 4. For more information, see section 59, Internal Voltage Regulator

19.5 Usage Notes

19.5.1 Procedure for Specifying the Pin Functions

To specify the I/O pin functions:

1. Clear the B0WI bit in the PWPR_NS register. This enables writing to the PFSWE bit in the PWPR_NS register.*1
2. Set 1 to the PFSWE bit in the PWPR_NS register. This enables writing to the PmnPFS register.*1
3. Clear the Port Mode Control bit in the PMR to 0 for the target pin to select the general I/O port.
4. Specify the I/O function for the pin through the PSEL[4:0] bits settings in the PmnPFS register.
5. Set the PMR bit to 1 as required to switch to the selected I/O function for the pin.
6. Clear the PFSWE bit in the PWPR_NS register. This disables writing to the PmnPFS register.*1
7. Set 1 to the B0WI bit in the PWPR_NS register. This disables writing to the PFSWE bit in the PWPR_NS register.*1

Note 1. When the security attribution of Pmn is set to 0, set the PWPR_S register to write to the PmnPFS register.

19.5.2 Procedure for Using Port Group Input

To use the port group input (port n (n = 1 to 4)):

1. Set the ELSRx.ELS[8:0] bits to all 0 to ignore unexpected pulses. For more information, see section 18, Event Link Controller (ELC).
2. Set the EOFR[1:0] bits of the PmnPFS register to specify the rising, falling, or both edge detections.
3. Execute a dummy read or wait for a short time, for example 100 ns. Ignoring of unexpected pulses depends on the initial value of the external pin.
4. Set the ELSRx.ELS[8:0] bits to enable the event signals.

19.5.3 Port Output Data Register (PODR) Summary

This register outputs data as follows:

1. Outputs 0 if PCNTR4.EORR is set to 1 when ELC_PORTn (n = 1, 2, 3 or 4) signal occurs.
2. Outputs 1 if PCNTR4.EOSR is set to 1 when ELC_PORTn (n = 1, 2, 3 or 4) signal occurs.
3. Outputs 0 if PCNTR3.PORR is set to 1.
4. Outputs 1 if PCNTR3.POSR is set to 1.
5. Outputs 0 or 1 because PCNTR1.PODRn is set.

6. Outputs 0 or 1 because PmnPFS.PODRn is set.

Numbers in this list correspond to the priority for writing to the PODRn. For example, if 1. and 3. from the list occur at the same time, the higher priority event 1. is executed.

19.5.4 Notes on Using Analog Functions

To use an analog function, set the Port Mode Control bit (PMR) and the Port Direction bit (PDRn) to 0 so that the pin acts as a general input port. Next, set the Analog Input Enable bit (ASEL) in the Port mn Pin Function Select Register (PmnPFS.ASEL) to 1.

19.5.5 I/O Buffer Specification

The P402, P403, and P404 can be used as the RTC input, tamper detection and other peripheral functions. Table 19.4 lists the P402, P403, P404 specifications.

Table 19.4 P402, P403, P404 specifications

I/O port	RTC and tamper detection			Other peripheral	
	RTC and tamper detection input enable register	RTC	Tamper detection	Other peripheral enable register	CAC, CEU GPT, CAN, SCI, SDHI, SSIE, ETHERC (MII), ETHERC (RMII), and interrupt
P402	VBTICTLR.VCH0INEN	RTCIC0	RTCIC0	P402PFS.PSEL and PMR	For details, see section 19.6. Peripheral Select Settings for Each Product.
P403	VBTICTLR.VCH1INEN	RTCIC1	RTCIC1	P403PFS.PSEL and PMR	
P404	VBTICTLR .VCH2INEN	RTCIC2	RTCIC2	P404PFS.PSEL and PMR	

These RTC inputs and tamper detection are controlled by the VBTICTLR register and this register has the highest priority for selecting the functions.

P402, P403, and P404 can be used as IRQn-DS (n = 4, 14, 15) whether RTC inputs and tamper detection are selected or not. When using these interrupts, set the interrupt procedure after setting the VBTICTLR register. See section 11.2.8. VBTICTLR : VBATT Input Control Register.

See Figure 19.5.

The VBTICTLR register is only initialized by VBATT_POR reset. Therefore, when not using the RTC inputs and tamper detection, the associated bit of VBTICTLR register must be set to 0 after except VBATT_POR reset.

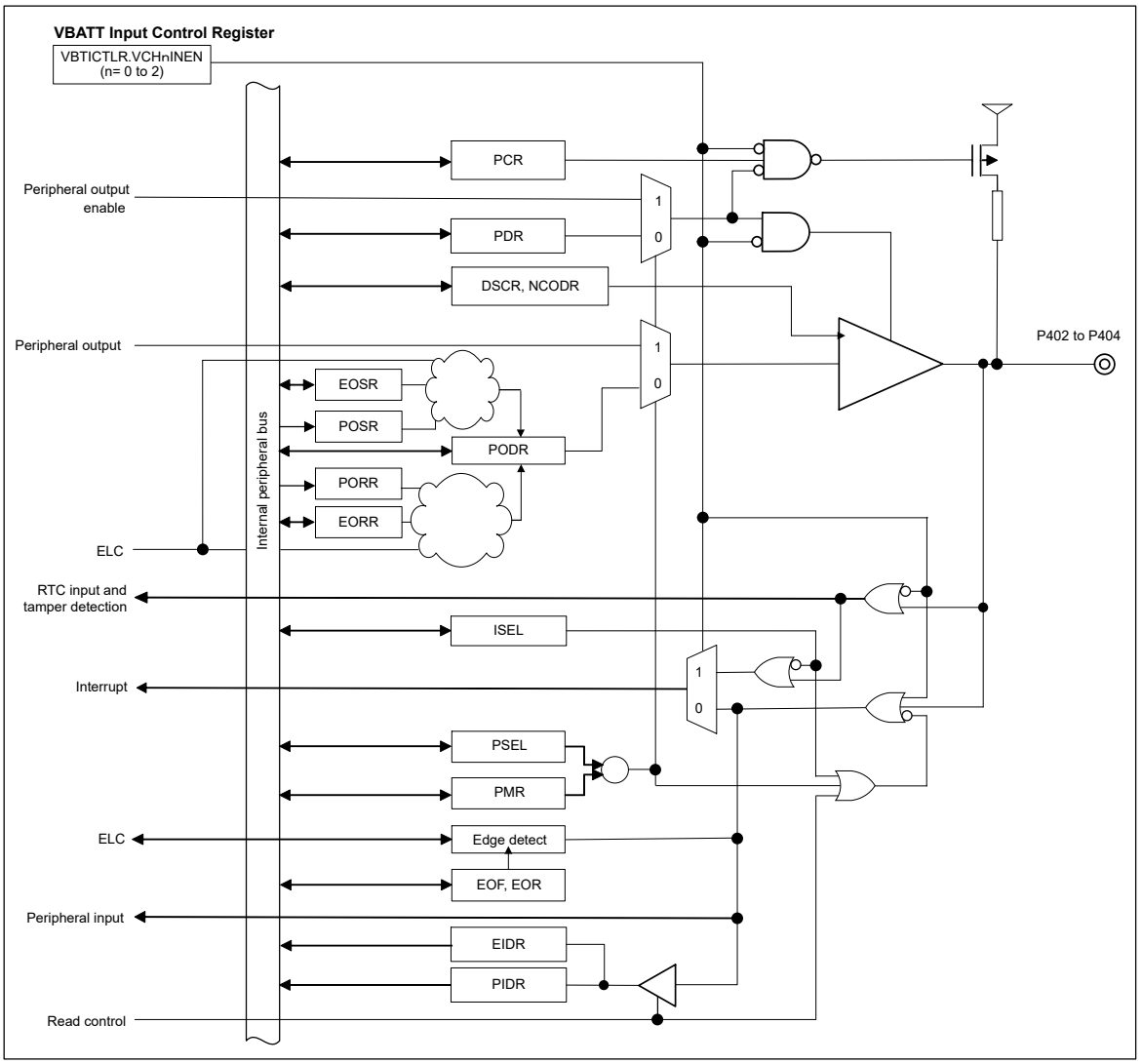


Figure 19.5 P402, P403, P404 diagram

19.5.6 Selecting Pins USB_DP and USB_DM

Pins USB_DP and USB_DM are shared with pins P814 and P815, respectively. Pins USB_DP and P814 can be set with the PFS.P814PFS.PMR bit, and pins USB_DM and P815 can be set with the PFS.P815PFS.PMR bit. Table 19.5 shows setting values of bits PFS.P814PFS.PMR and PFS.P815PFS.PMR with each selected pin.

Table 19.5 Selecting the USB/PORT pins

PMR Bits Settings		Pins Selected	
P814PFS.PMR bit	P815PFS.PMR bit	P814/USB_DP Pin	P815/USB_DM Pin
0	0	P814	P815
0	1	P814	P815
1	0	P814	P815
1	1	USB_DP	USB_DM

Note: When using P814/USB_DP and P815/USB_DM as GPIO pins (P814 and P815), use the USB related registers with their initial values.

Note: When using P814/USB_DP and P815/USB_DM as USB pins (USB_DP and USB_DM), use the GPIO related registers for P814 and P815 with their initial values.
Note: When using P814/USB_DP and P815/USB_DM as GPIO pins or USB pins, set these pins only once after a reset.

19.5.7 Pull-up/Pull-down Setting for P814 and P815 at USBFS/GPIO Function

When the GPIO function of P814 and P815 is used, the Pull-up/Pull-down function of USBFS register affects to P814 and P815. Therefore, before using GPIO function, the pull-up and pull-down control of USBFS needs to be disabled. The control register is SYSCFG.DPRPU and SYSCFG.DRPD bits.

19.6 Peripheral Select Settings for Each Product

This section describes the pin function select configuration using the PmnPFS register. Some pin names have added _A, _B, or _C suffixes. When assigning IIC, SPI, SSIE, ETHERC and SDHI functionality, select the functional pins having the same suffix. The other pins can be selected regardless of the suffix. Assigning the same function to two or more pins simultaneously is prohibited.

1. In Pmn pin function select register(PmnPFS), the PSEL bits have to be set when the PMR bit of the target pin is 0. If the PSEL bits are set when the PMR bit is 1, the unexpected edges may be input at the input function or the unexpected pulses may be output to the external pin at the output function.
2. Only the allowed values (functions) should be specified in the PSEL bits of PmnPFS register. If a value which is not allowed for the register is specified, the correct operation is not guaranteed.
3. The single function should not be assigned to the multiple pins by PmnPFS register. When the GPT1, GPT5, SCI3, IIC0 or SPI0 are configured as secure and these pin function is being assigned to the pin which security attribution is set as secure by the PmSAR register, the write access to the PSEL bits for setting same function as secure pin in other pins is ignored when the security attribution of that pin is non-secure. For example, if the PSARE.PSARE30 bit is 0 (GPT1 is secure) and the P209PFS.PSEL bits is 00011b (pin function is GTIOC1A) and the P2SAR.109SA bit is 0 (P209 is secure), the write 00011b to the P405PFS.PSEL bits is ignored when the P4SAR.405SA bit is 1 (P405 is non-secure).
4. The PORT0, PORT5 and PORT8 have the analog functions such as A/D converter. When these pins are used as an analog function, for avoiding the loss of resolution, the PMR bit should be set to 0 and PDR bit should be set to 0. After that, ASEL bit should be set to 1

Table 19.6 Register settings for input/output pin function (PORT0)

PSEL[4:0] settings	Function	pin													
		P000	P001	P002	P003	P004	P005	P006	P007	P008	P009	P010	P011	P014	P015
00000b (value after reset)	Hi-Z/ JTAG/SWD	Hi-Z													
ASEL bit		AN100/ IVCMP2	AN101/ IVREF0	AN102/ IVCMP3	AN104/ IVREF1	AN000/ IVCMP2	AN001	AN002/ IVCMP3	AN004	AN008	AN006	AN005/ IVCMP0	AN106	AN007/ DA0	AN105/ DA1
ISEL bit		IRQ6-DS	IRQ7-DS	IRQ8-DS	—	IRQ9-DS	IRQ10-DS	IRQ11-DS	—	IRQ12-DS	IRQ13-DS	IRQ14	—	—	IRQ13
DSCR[1:0] bits	Drive capability control*1	L	L	L	L	L	L	L	L	L	L	L	L	L	L
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
176 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	✓	✓
176 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	✓	✓

✓: Available
—: Setting prohibited
Note 1. The drive strength of this port cannot be controlled by PmnPFS.DSCR[1:0] bits.

Table 19.7 Register settings for input/output pin function (PORT1) (1 of 2)

PSEL [4:0]settings	Function	pin											
		P100	P101	P102	P103	P104	P105	P106	P107	P112	P113	P114	P115
00000b(value after reset)	Hi-Z/ JTAG/SW D	Hi-Z											

Table 19.7 Register settings for input/output pin function (PORT1) (2 of 2)

PSEL [4:0]settings	Function	pin											
		P100	P101	P102	P103	P104	P105	P106	P107	P112	P113	P114	P115
00001b	AGT	AGTIO0	AGTEE0	AGTO0	—	—	—	AGTOB0	AGTOA0	—	—	—	—
00010b	GPT*2	GTETRGA	GTETRGB	GTOWLO	GTOWUP	GTETRGB	—	GTOWLO	GTOWUP	—	—	—	—
00011b	GPT*2	GTIOC8B	GTIOC8A	GTIOC2B	GTIOC2A	GTIOC1B	GTIOC1A	GTIOC8B	GTIOC8A	GTIOC3B	GTIOC2A	GTIOC2B	GTIOC5A
00100b	SCI*1	—	—	—	—	—	—	—	—	TXD0_A/ MOSI0_A/ SDA0_A	RXD0_A/ MISO0_A/ SCL0_A	CTS0_RTS0_A/ SS0_A	CTS0_A
00101b	SCI*1	SCK9_A	RXD9_A/ MISO9_A/ SCL9_A/	TXD9_A/ MOSI9_A/ SDA9_A/	CTS9_RTS9_A/ SS9_A	CTS9_A	—	—	—	—	—	—	—
00110b	SPI*1	MISOB_A	MOSIB_A	RSPCKB_A	SSLB0_A	SSLB1_A	SSLB2_A	SSLB3_A	—	SSLA2_B	SSLA1_B	SSLA0_B	MOSIA_B
01010b	CAC/ ADC12	—	—	ADTRG 0	—	—	—	—	—	—	—	—	—
01011b	BUS	—	—	—	—	—	—	—	—	A00/ BC0/ DQM1	CS1/ CKE	CS0/ WE	SDCS
01101b	SCI	—	—	—	—	—	—	—	—	—	—	DE0	—
01110b	SCI	DE9	—	—	DE9	—	—	—	—	—	—	—	—
10000b	CANFD	—	—	CRX0	CTX0	—	—	—	—	—	—	—	—
10010b	SSIE*1	—	—	—	—	—	—	—	—	SSIBCK0_B	SSILRCK0/ SSIFS0_B	SSIRXD0_B	SSITXD0_B
10110b	ETHERC (MII)	—	—	—	—	—	—	—	—	ET0_CRS	ET0_EXOUT	ET0_LINKSTA	ET0_WOL
10111b	ETHERC (RMII)	—	—	—	—	—	—	—	—	RMII0_CRS_DV_A	ET0_EXOUT	ET0_LINKSTA	ET0_WOL
11001b	GLCDC	—	—	—	—	—	—	—	—	LCD_DATA10_A	LCD_DATA09_A	LCD_DATA08_A	LCD_DATA07_A
11100b	OSPI	OM_SIO0	OM_SIO3	OM_SIO4	OM_SIO2	OM_CS1	OM_ECSINT1	OM_RESET	OM_CS0	—	—	—	—
11110b	ULPT	—	—	—	—	—	ULPTO1-DS	ULPTEE1-DS	—	ULPTOB0-DS	ULPTOA0-DS	—	—
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		IRQ2	IRQ1	—	—	IRQ1	IRQ0	—	—	—	—	—	—
DSCR[1:0] bits	Drive capability control	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
176 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
176 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance “_A”, “_B”, or “_C” to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Table 19.8 Register settings for input/output pin function (PORT2) (1 of 2)

PSEL [4:0]settings	Function	Pin													
		P200*4	P201	P202	P203	P204	P205	P206	P207	P208	P209	P210	P211	P212	P213
00000b(value after reset)	Hi-Z/ JTAG/SWD	Hi-Z								TDI	TDO	TMS/ SWDIO	TCK/ SWCLK	Hi-z	
00001b	AGT	—	—	—	—	AGTIO1	AGTO1	—	—	—	—	—	—	AGTEE1	—
00010b	GPT*2	—	—	—	—	GTIW	GTIV	GTIU	—	GTOVLO	GTOVUP	GTOULO	GTOUUP	GTETRGD	GTETRGC
00011b	GPT*2	—	—	GTIOC5B	GTIOC5A	GTIOC4B	GTIOC4A	—	—	GTIOC1B	GTIOC1A	GTIOC0B	GTIOC0A	GTIOC0B	GTIOC0A

Table 19.8 Register settings for input/output pin function (PORT2) (2 of 2)

PSEL [4:0]settings	Function	Pin													
		P200*4	P201	P202	P203	P204	P205	P206	P207	P208	P209	P210	P211	P212	P213
00100b	SCI*1	—	—	—	—	SCK4_A	TXD4_A/ MOSI4_ A/ SDA4_A	RXD4_A/ MISO4_ A/ SCL4_A	—	—	—	—	—	—	—
00101b	SCI*1	—	—	—	—	—	—	—	—	RXD9_B/ MISO9_ B/ SCL9_B	TXD9_B/ MOSI9_ B/ SDA9_B	CTS9_RTS9_B/ SS9_B	SCK9_B	RXD1_C / MISO1_ C/ SCL1_C	TXD1_C/ MOSI1_ C/ SDA1_C
00110b	SPI*1	—	—	MOSIA_ A	RSPCKA _A	SSLA0_ A	SSLA1_ A	SSLA2_ A	—	—	—	—	—	—	—
00111b	IIC/I3C*1	—	—	—	—	—	SCL1_B	SDA1_B	—	—	—	—	—	—	—
01001b	CLKOUT/ ACMPHS/RTC	—	—	—	—	—	CLKOUT	—	—	VCOUT	CLKOUT	—	—	—	—
01010b	CAC/ ADC12	—	—	—	—	CACREF	—	—	—	—	—	—	—	—	ADTRG1
01011b	BUS	—	—	—	—	—	—	CS7	—	—	—	—	—	—	—
01101b	SCI	—	—	—	—	DE4	—	—	—	—	—	—	—	—	—
01110b	SCI	—	—	—	—	—	—	—	—	—	—	DE9	DE9	—	—
10000b	CANFD	—	—	CRX0	CTX0	—	—	—	—	CRX1	CTX1	—	—	—	—
10010b	SSIE*1	—	—	—	—	SSIBCK1_A	SSILRCK1/ SSIFS1_A	SSIDATA1_A	—	—	—	—	—	—	—
10011b	USBFS	—	—	—	—	USB_OVRCURB	USB_OVRCURA	USB_VBUSEN	—	—	—	—	—	—	—
10101b	SDHI*1	—	—	SD0DAT6_A	SD0DAT5_A	SD0DAT4_A	SD0DAT3_A	SD0DAT2_A	—	—	—	—	—	—	—
11001b	GLCDC	—	—	—	—	—	—	—	LCD_DATA9_B	—	—	—	—	—	—
11110b	ULPT	—	—	ULPTOB1	ULPTOA1	—	—	—	—	—	—	—	—	—	ULPTEE0
11111b	MIPI	—	—	—	—	—	—	DSI_TE	—	—	—	—	—	—	—
Don't-care	Trace(Debug)/Clock	—	—	—	—	—	—	—	—	—	SWO	—	—	EXTAL	XTAL
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		—	—	IRQ3-DS	IRQ2-DS	—	IRQ1-DS	IRQ0-DS	—	IRQ3	—	—	—	IRQ3	IRQ2
DSCR[1:0] bits	Drive capability control	—	L*3	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
224 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
176 pins product		✓	✓	—	—	—	—	✓	—	✓	✓	✓	✓	✓	✓
176 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	—	✓	✓	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance "_A", "_B", or "_C", to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.
Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).
Note 3. The driver strength of this port can not be controlled by PmnPFS.DSCR[1:0] bits.
Note 4. When using NMI pin interrupt, Port related registers setting are not required.

Table 19.9 Register settings for input/output pin function (PORT3) (1 of 2)

PSEL [4:0]settings	Function	Pin														
		P300	P301	P302	P303	P304	P305	P306	P307	P308	P309	P310	P311	P312	P313	P314	P315
00000b(value after reset)	Hi-Z/ JTAG/SWD	Hi-z															

Table 19.9 Register settings for input/output pin function (PORT3) (2 of 2)

PSEL [4:0]settings	Function	Pin															
		P300	P301	P302	P303	P304	P305	P306	P307	P308	P309	P310	P311	P312	P313	P314	P315
00001b	AGT	—	AGTIO0	—	—	—	—	—	—	—	—	AGTEE1	AGTOB1	AGTOA1	—	—	—
00010b	GPT*2	—	GTOULO	GTOUUP	—	GTOVLO	GTOVUP	GTIW	GTIV	GTIU	—	—	GTADSM1	GTADSM0	—	—	—
00011b	GPT*2	GTIOC3A	GTIOC4B	GTIOC4A	GTIOC7B	GTIOC7A	—	—	—	—	—	—	—	—	—	—	—
00100b	SCI*1	SCK0_A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
00101b	SCI*1	—	—	—	—	—	—	—	—	CTS9_B	RXD3_B/MISO3_B/SCL3_B	TXD3_B/MOSI3_B/SDA3_B	SCK3_B	CTS3_RTS3_B/SS3_B	CTS3_C	CTS3_RTS3_C/SS3_C	SCK3_C
00110b	SPI*1	SSLA3_B	—	—	—	—	—	—	—	—	—	—	—	—	MISOA_A	—	—
01010b	CAC/ADC12	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADTRG0	—
01011b	BUS	A01/DQM3	A02	A03	A04	A05	A06	A07	A08	A09	A10	A11	A12	A13	—	—	—
01101b	SCI	DE0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
01110b	SCI	—	—	—	—	—	—	—	—	—	—	—	DE3	DE3	—	DE3	DE3
10000b	CANFD	—	—	—	—	—	—	—	—	—	—	—	CRX0	CTX0	—	—	—
10101b	SDHI*1	—	SD0DAT3_B	SD0DAT2_B	SD0DAT1_B	SD0DAT0_B	SD0WP	SD0CD	SD0CMD_B	SD0CLK_B	—	—	—	—	SD0DAT7_A	—	—
10110b	ETHERC (MII)	ET0_RX_CLK	ET0_ERXD0	ET0_ERXD1	ET0_ETXD0	ET0_ETXD1	ET0_RX_ER	ET0_TX_EN	ET0_MDIO	ET0_MDC	ET0_ETXD3	ET0_ETXD2	ET0_TX_ER	ET0_TX_CLK	—	—	—
10111b	ETHERC (RMII)	RMII0_RX_ER_A	RMII0_RXD1_A	RMII0_RXD0_A	REF50CK0_A	RMII0_TXD0_A	RMII0_TXD1_A	RMII0_TXD_EN_A	ET0_MDIO	ET0_MDC	—	—	—	—	—	—	—
11001b	GLCDC	LCD_DATA11_A	LCD_DATA12_A	LCD_DATA13_A	LCD_DATA14_A	—	—	—	—	—	LCD_DATA15_A	LCD_DATA16_A	LCD_DATA17_A	LCD_DATA18_B	—	—	—
11110b	ULPT	ULPTEVI0-DS	ULPTEE0-DS	ULPTO0-DS	—	ULPTO1	ULPTEE1	ULPTEVI1	ULPTOA1	ULPTOB1	—	—	—	—	—	—	—
Don't-care	Trace(Debug)	—	—	—	—	TDATA3	TDATA2	TDATA1	TDATA0	TCLK	—	—	—	—	—	—	—
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		IRQ4	IRQ6	IRQ5	—	IRQ9	IRQ8	—	—	—	—	—	—	—	—	—	—
DSCR[1:0] bits	Drive capability control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—
224 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
176 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—
176 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance "_A", "_B", or "_C", to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Table 19.10 Register settings for input/output pin function (PORT4) (1 of 3)

PSEL [4:0]settings	Function	pin															
		P400	P401	P402	P403	P404	P405	P406	P407	P408	P409	P410	P411	P412	P413	P414	P415
00000b(value after reset)	Hi-Z/JTAG/SWD	Hi-Z															

Table 19.10 Register settings for input/output pin function (PORT4) (2 of 3)

PSEL [4:0]settings	Function	pin															
		P400	P401	P402	P403	P404	P405	P406	P407	P408	P409	P410	P411	P412	P413	P414	P415
00001b	AGT	AGTIO1	—	—	—	—	AGTIO1	—	AGTIO0	—	—	AGTOB1	AGTOA1	AGTEE1	—	—	—
00010b	GPT*3	—	GTETRGA	—	—	—	—	—	—	GTOWLO	GTOWUP	GTOVLO	GTOVUP	GTOULO	GTOUUP	GTADSM1	GTADSM0
00011b	GPT*3	GTIOC6A	GTIOC6B	—	GTIOC3A	GTIOC3B	GTIOC1A	GTIOC1B	GTIOC10B	GTIOC10A	—	GTIOC9B	GTIOC9A	—	—	GTIOC0B	GTIOC0A
00100b	SCI*2	—	—	—	—	—	SCK2_B	TXD2_B/ MOSI2_B/ SDA2_B	CTS4_RTS4_A/ SS4_A	CTS4_A	—	—	—	—	—	RXD4_B/ MISO4_B/ SCL4_B	TXD4_B/ MOSI4_B/ SDA4_B
00101b	SCI*2	TXD1_A/ MOSI1_A/ SDA1_A	RXD1_A/ MISO1_A/ SCL1_A	SCK1_A	CTS1_RTS1_A/ SS4_A	CTS1_A	—	—	—	RXD3_A/ MISO3_A/ SCL3_A	TXD3_A/ MOSI3_A/ SDA3_A	SCK3_A	CTS3_RTS3_A/ SS3_A	CTS3_A	—	—	—
00110b	SPI*2	—	—	—	—	—	—	SSLA3_C	SSLA3_A	—	—	MISOB_B	MOSIB_B	RSPCKB_B	SSLB0_B	SSLB1_B	SSLB2_B
00111b	IIC/I3C*2	I3C_SCL	I3C_SDA	—	—	—	—	—	SDA0_B	SCL0_B	SDA0_A	SCL0_A	—	—	—	—	—
01001b	CLKOUT/ ACMPHS/ RTC	—	—	—	—	—	—	—	RTCOUT	—	—	—	—	—	—	—	—
01010b	CAC/ ADC12	ADTRG1	—	CACREF	—	—	—	—	ADTRG0	—	—	—	—	—	—	—	—
01011b	BUS	—	—	—	—	—	—	—	CS6	A17	A18	A19	A20	A21	A22	A23	WAIT
01101b	SCI	—	—	—	—	—	DE2	—	DE4	—	—	—	—	—	—	—	—
01110b	SCI	—	—	DE1	DE1	—	—	—	—	—	—	DE3	DE3	—	—	—	—
01111b	CEU	VIO_D0	VIO_D1	—	—	VIO_D3	VIO_D2	VIO_D3	—	—	—	—	—	—	—	VIO_D13	VIO_D12
10000b	CANFD	—	CTX0	CRX0	—	—	—	—	—	—	—	—	—	—	—	CRX1	CTX1
10010b	SSIE*2	AUDIO_CLK	—	AUDIO_CLK	SSIBCK0_A	SSILRCK0/ SSIFS0_A	SSITXD0_A	SSIRXD0_A	—	—	—	—	—	—	—	—	—
10011b	USBFS	—	—	—	—	—	—	—	USB_VBUS	USB_VBUSEN	USB_OVRCURA-DS	USB_OVRCURB-DS	USB_ID	USB_EXICEN	—	—	—
10100b	USBHS	—	—	—	—	—	—	—	—	USBHS_VBUSEN	USBHS_OVRCURA	USBHS_OVRCURB	USBHS_ID	USBHS_EXICEN	—	—	—
10101b	SDHI*2	SD1CLK_B	SD1CMD_B	SD1DAT0_B	SD1DAT1_B	SD1DAT2_B	SD1DAT3_B	SD1CD	—	—	—	SD0DAT1_A	SD0DAT0_A	SD0CMD_A	SD0CLK_A	SD0WP	SD0CD
10110b	ETHERC (MII)	ET0_WOL	ET0_MDC	ET0_MDIO	ET0_LINKSTA	ET0_EXOUT	ET0_TX_EN	ET0_RX_ER	—	—	—	—	—	—	—	—	—
10111b	ETHERC (RMII)	ET0_WOL	ET0_MDC	ET0_MDIO	ET0_LINKSTA	ET0_EXOUT	RMII0_TXD_EN_B	RMII0_TXD1_B	—	—	—	—	—	—	—	—	—
11110b	ULPT	—	—	—	—	—	—	—	—	ULPTOB0	ULPTOA0	—	—	—	ULPTEE1	—	—
Don't-care	RTC/Clock	—	—	RTCIC0 *1	RTCIC1 *1	RTCIC2 *1	—	EXCIN	—	—	—	—	—	—	—	—	—
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		IRQ0	IRQ5-DS	IRQ4-DS	IRQ14-DS	IRQ15-DS	—	—	—	IRQ7	IRQ6	IRQ5	IRQ4	—	—	IRQ9	IRQ8
DSCR[1:0] bits	Drive capability control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
176 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 19.10 Register settings for input/output pin function (PORT4) (3 of 3)

PSEL [4:0]settings	Function	pin															
		P400	P401	P402	P403	P404	P405	P406	P407	P408	P409	P410	P411	P412	P413	P414	P415
176 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. To use this pin function, set the associated pin as a general input (set the PmnPFS.PDR and PmnPFS.PMR bits to 0).
Note 2. Recommend using pins that have a letter appended to their names, for instance "_A", "_B", or "_C", to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.
Note 3. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Table 19.11 Register settings for input/output pin function (PORT5)

PSEL [4:0]settings	Function	pin															
		P500	P501	P502	P503	P504	P505	P506	P507	P508	P509	P510	P511	P512	P513	P514	P515
00000b (value after reset)	Hi-Z/ JTAG/SWD	Hi-Z															
00011b	GPT*2	—	—	—	—	—	—	—	—	—	—	—	GTIOC0B	GTIOC0A	—	—	—
00111b	IIC/I3C	—	—	—	—	—	—	—	—	—	—	—	SDA1_A	SCL1_A	—	—	—
01010b	CAC/ ADC12	CACREF	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
01011b	BUS	—	—	—	—	—	D27/ DQ27	D28/ DQ28	D29/ DQ29	D30/ DQ30	D31/ DQ31	WR3/ BC3	—	—	—	—	—
01111b	CEU	—	—	—	—	—	—	—	—	—	—	—	—	—	VIO_FLD	—	—
10000b	CANFD	—	—	—	—	—	—	—	—	—	—	—	CRX1	CTX1	—	—	—
10011b	USBFS	USB_VBUSEN	USB_OVRCURA	USB_OVRCURB	—	—	—	—	—	—	—	—	—	—	—	—	—
10101b	SDHI*1	SD1DAT1_A	SD1DAT2_A	SD1DAT3_A	SD1CD	SD1WP	SD1DAT4_A	SD1DAT5_A	SD1DAT6_A	SD1DAT7_A	—	—	—	—	—	—	—
11001b	GLCDC	—	—	—	—	—	—	—	—	—	—	—	—	—	LCD_TCON2_B	LCD_EXTCLK_B	LCD_TCON3_B
11110b	ULPT	—	—	—	—	—	—	—	—	—	ULPTEVI1	ULPTEVI0	—	—	—	—	—
ASEL bit		AN121	AN120	AN019/ AN119	—	—	—	—	—	—	—	—	—	—	AN016/ AN116/ IVCMP0	—	—
ISEL bit		—	—	—	—	—	—	—	—	IRQ1	IRQ2	IRQ3	IRQ15	IRQ14	—	—	—
DSCR[1:0] bits	Drive capability control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
176 pins product		✓	✓	✓	—	—	—	—	—	—	—	—	✓	✓	✓	—	—
176 pins w/o MIPI product		✓	✓	✓	—	—	—	—	—	—	—	—	✓	✓	✓	—	—

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance "_A", "_B", or "_C", to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.
Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Table 19.12 Register settings for input/output pin function (PORT6)

PSEL [4:0]settings	Function	Pin														
		P600	P601	P602	P603	P604	P605	P606	P607	P609	P610	P611	P612	P613	P614	P615
00000b(value after reset)	Hi-Z/ JTAG/SW D	Hi-Z														
00001b	AGT	—	—	—	—	—	—	—	—	—	—	—	—	AGTO1	AGTO0	—
00010b	GPT*2	—	—	—	—	—	—	—	—	—	—	—	—	GTETRGA	GTETRGB	GTETRGC
00011b	GPT*2	GTIOC6B	GTIOC6A	GTIOC7B	GTIOC7A	GTIOC8B	GTIOC8A	—	—	GTIOC5B	GTIOC4A	GTIOC4B	—	—	—	—
00100b	SCI*1	—	SCK0_B	RXD0_B/ MISO0_B/ SCL0_B	TXD0_B/ MOSI0_B/ SDA0_B	CTS0_RTS0_B/ SS0_B	CTS0_B	—	—	TXD0_C/ MOSI0_C/ SDA0_C	RXD0_C/ MISO0_C/ SCL0_C	SCK0_C	CTS0_RTS0_C/ SS0_C	CTS0_C	—	—
00110b	SPI*1	—	—	—	—	—	—	—	—	MISOA_B	RSPCKA_B	MOSIA_B	SSLA0_B	—	—	—
01001b	CLKOUT/ ACMPHS/RTC	—	RTCOUT	—	—	—	—	—	—	—	—	CLKOUT	—	—	—	—
01010b	CAC/ ADC12	CACREF	—	—	—	—	—	—	—	—	—	CACREF	—	—	—	—
01011b	BUS	—	D00 [A00/ D00]/ DQ00	D01 [A01/ D01]/ DQ01	D02 [A02/ D02]/ DQ02	D03 [A03/ D03]/ DQ03	D04 [A04/ D04]/ DQ04	D05 [A05/ D05]/ DQ05	D06 [A06/ D06]/ DQ06	D08 [A08/ D08]/ DQ08	D09 [A09/ D09]/ DQ09	D10 [A10/ D10]/ DQ10	D11 [A11/ D11]/ DQ11	D12 [A12/ D12]/ DQ12	D13 [A13/ D13]/ DQ13	D14 [A14/ D14]/ DQ14
01101b	SCI	—	DE0	—	—	DE0	—	—	—	—	—	DE0	DE0	—	—	—
10000b	CANFD	—	—	—	—	—	—	—	—	CTX1	CRX1	—	—	—	—	—
10011b	USBFS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	USB_VBUSEN
10110b	ETHERC (MII)	—	—	—	—	—	—	—	—	ET0_RX_DV	ET0_COL	ET0_ERXD2	ET0_ERXD3	—	—	—
11001b	GLCDC	—	—	—	—	—	—	—	LCD_EXTCLK_A	LCD_DATA6_A	LCD_DATA5_A	LCD_DATA4_A	LCD_DATA3_A	LCD_DATA2_A	LCD_DATA1_A	LCD_DATA0_B
11100b	OSPI	OM_RSTO1	OM_WP1	—	—	—	—	—	—	—	—	—	—	—	—	—
11110b	ULPT	ULPTEVI1-DS	ULPTEVI0	ULPTEE0	ULPTO0	—	—	—	—	ULPTOA1-DS	ULPTOB1-DS	—	—	—	—	—
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—	IRQ7
DSCR[1:0] bits	Drive capability control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
176 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
176 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance “_A”, “_B”, or “_C”, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Table 19.13 Register settings for input/output pin function (PORT7) (1 of 2)

PSEL [4:0]settings	Function	Pin														
		P700	P701	P702	P703	P704	P705	P706	P707	P708	P709	P710	P711	P712	P713	P714	P715
00000b(value after reset)	Hi-Z/ JTAG/SW D	Hi-Z															
00001b	AGT	—	—	—	AGTO1	AGTO0	AGTIO0	AGTIO0	—	—	—	—	AGTEE0	AGTOB0	AGTOA0	—	—

Table 19.13 Register settings for input/output pin function (PORT7) (2 of 2)

PSEL [4:0]settings	Function	Pin															
		P700	P701	P702	P703	P704	P705	P706	P707	P708	P709	P710	P711	P712	P713	P714	P715
00010b	GPT*2	—	—	—	—	GTADSM0	GTADSM1	—	—	—	—	—	—	—	—	—	—
00011b	GPT*2	GTIOC5A	GTIOC5B	GTIOC6A	GTIOC6B	—	—	—	—	—	—	—	—	GTIOC2B	GTIOC2A	—	—
00100b	SCI*1	RXD2_B/MISO2_B/SCL2_B	CTS2_RTS2_B/SS2_B	CTS2_B	—	—	—	—	—	SCK4_B	CTS4_RTS4_B/SS4_B	CTS4_B	—	—	—	TXD4_C/MOSI4_C/SDA4_C	RXD4_C/MISO4_C/SCL4_C
00101b	SCI*1	—	—	—	—	—	CTS1_B	RXD1_B/MISO1_B/SCL1_B	TXD1_B/MOSI1_B/SDA1_B	—	—	—	—	—	—	—	—
00110b	SPI*1	MISOA_C	MOSIA_C	RSPCKA_C	SSLA0_C	SSLA1_C	SSLA2_C	—	—	SSLB3_B	—	—	—	—	—	—	—
01001b	CLKOUT/ACMPHS/RTC	—	—	—	VCOUT	—	—	—	—	—	—	—	—	—	—	—	—
01010b	CAC/ADC12	—	—	—	—	—	—	—	—	CACREF	—	—	—	—	—	—	—
01011b	BUS	—	—	—	—	—	—	—	—	WR1/BC1	CS4	CS5	—	—	—	—	—
01101b	SCI	—	DE2	—	—	—	—	—	—	DE4	DE4	—	—	—	—	—	—
01111b	CEU	VIO_D4	VIO_D5	VIO_D6	VIO_D7	VIO_D8	VIO_D9	VIO_D10	—	VIO_CLK	VIO_HD	VIO_VD	—	—	—	—	—
10000b	CANFD	—	—	—	—	CTX0	CRX0	—	—	—	—	—	—	—	—	—	—
10010b	SSIE*1	SSIDATA1_B	SSILRCK1/SSIFS1_A	SSIBCK1_B	—	—	—	—	—	AUDIO_CLK	—	—	—	—	—	—	—
10100b	USBHS	—	—	—	—	—	—	USBHS_OVRCURB-DS	USBHS_OVRCURA-DS	—	—	—	—	—	—	—	—
10101b	SDHI*1	SD1WP	SD1DAT4_B	SD1DAT5_B	SD1DAT6_B	SD1DAT7_B	—	—	—	—	—	—	—	—	—	—	—
10110b	ETHERC (MII)	ET0_ETXD1	ET0_ETXD0	ET0_ERXD1	ET0_ERXD0	ET0_RX_CLK	ET0_CRS	—	—	—	—	—	—	—	—	—	—
10111b	ETHERC (RMII)	RMII0_TXD0_B	REF50CK0_B	RMII0_RXD0_B	RMII0_RXD1_B	RMII0_RX_ER_B	RMII0_CRS_DV_B	—	—	—	—	—	—	—	—	—	—
11001b	GLCDC	—	—	—	—	—	—	—	LCD_DATA23_B	—	—	—	LCD_DATA10_B	LCD_DATA11_B	LCD_DATA12_B	LCD_DATA13_B	LCD_DATA14_B
11110b	ULPT	—	ULPTO1	ULPTO0	—	—	—	—	—	—	—	—	—	—	—	—	—
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		—	—	—	—	—	—	IRQ7	IRQ8	IRQ11	IRQ10	—	—	—	—	—	—
DSCR[1:0] bits	Drive capability control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
176 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—	—	—
176 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—	—	—

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance “_A”, “_B”, or “_C”, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Table 19.14 Register settings for input/output pin function (PORT8)

PSEL [4:0]settings	Function	Pin															
		P800	P801	P802	P803	P804	P805	P806	P807	P808	P809	P810	P811	P812	P813	P814	P815
00000b(value after reset)	Hi-Z/ JTAG/SWD	Hi-Z															
00001b	AGT	AGTOA0	AGTOB0	—	—	—	—	—	—	—	—	—	—	—	—	—	—
00010b	GPT*2	GTIU	GTIV	GTIW	GTETRGC	GTETRGD	—	—	—	—	—	—	—	—	—	—	—
00011b	GPT*2	GTIOC11A	GTIOC11B	GTIOC12A	GTIOC12B	GTIOC13A	—	—	—	GTIOC13B	—	—	—	—	—	GTIOC8B	GTIOC8A
00100b	SCI*1	CTS2_A	TXD2_A/ MOSI2_A/ SDA2_A	RXD2_A/ MISO2_A/ SCL2_A	SCK2_A	CTS2_RTS2_A/ SS2_A	—	—	—	—	—	—	—	—	—	—	—
01011b	BUS	—	—	—	—	—	—	—	—	—	—	—	—	—	D20/ DQ20	—	—
01101b	SCI	—	—	—	DE2	DE2	—	—	—	—	—	—	—	—	—	—	—
01111b	CEU	—	—	—	—	—	VIO_D15	VIO_D14	—	—	—	—	—	—	—	—	—
10000b	CANFD	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CRX0	CTX0
10011b	USBFS	—	—	—	—	—	—	—	—	—	—	—	USB_ID	USB_EXICEN	—	—	—
10101b	SDHI*1	—	—	—	—	—	—	—	—	—	—	SD1CLK_A	SD1CMD_A	SD1DAT0_A	—	—	—
11001b	GLCDC	—	—	—	—	—	LCD_TCON0_B	LCD_CLK_B	LCD_TCON1_B	—	—	—	—	—	—	—	—
11100b	OSPI	OM_SIO5	OM_DQS	OM_SIO6	OM_SIO1	OM_SIO7	—	—	—	OM_SCLK	OM_SCLKN	—	—	—	—	—	—
11110b	ULPT	—	—	—	—	—	—	—	—	—	—	ULPTOA0	ULPTOB0	—	—	—	—
Don't-care	USBFS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UBS_DP	UBS_DM
ASEL bit		—	—	—	—	—	AN017/ AN117	AN018/ AN118	—	—	—	—	—	AN122	—	—	—
ISEL bit		IRQ11	IRQ12	—	—	IRQ14	—	IRQ0	—	IRQ15	—	—	—	—	—	—	—
DSCR[1:0] bits	Drive capability control	L/M/H/ HH	L/M/H/ HH	L/M/H/ HH	L/M/H/ HH	L/M/H/ HH	L/M/H	L/M/H	L/M/H	L/M/H/ HH	L/M/H/ HH	L/M/H/ HH	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
176 pins product		✓	✓	✓	✓	✓	✓	✓	—	✓	✓	✓	✓	✓	—	✓	✓
176 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	—	✓	✓	✓	✓	✓	—	✓	✓

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance "_A", "_B", or "_C", to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Table 19.15 Register settings for input/output pin function (PORT9) (1 of 2)

PSEL [4:0]settings	Function	Pin															
		P900	P901	P902	P903	P904	P905	P906	P907	P908	P909	P910	P911	P912	P913	P914	P915
00000b(value after reset)	Hi-Z/ JTAG/SWD	Hi-Z															
00001b	AGT	—	AGTIO1	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 19.15 Register settings for input/output pin function (PORT9) (2 of 2)

PSEL [4:0]settings	Function	Pin															
		P900	P901	P902	P903	P904	P905	P906	P907	P908	P909	P910	P911	P912	P913	P914	P915
00011b	GPT*2	—	—	—	GTIOC 11A	GTIOC 11B	—	GTIOC 13B	GTIOC 13A	GTIOC 12B	GTIOC 12A	—	GTIOC 3B	GTIOC 3A	—	GTIOC 5B	GTIOC 5A
00101b	SCI*1	TXD3_C/ MOSI3_C/ SDA3_C	RXD3_C/ MISO3_C/ SCL3_C	—	—	—	CTS3_B	—	—	—	—	—	—	—	—	—	—
01001b	CLKOUT/ ACMPHS/ RTC	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKOUT	—	—
01011b	BUS	—	—	—	—	—	A14	A15	A16	CS2/ RAS	CS3/ CAS	—	—	—	—	—	—
10011b	USBFS	—	—	—	—	—	—	USB_ID	USB_EXICEN	—	—	—	—	—	—	—	—
10100b	USBHS	—	—	—	—	—	—	—	—	USBHS_ID	USBHS_EXICEN	—	—	—	—	—	—
11001b	GLCDC	—	—	LCD_DATA8_B	LCD_DATA7_B	LCD_DATA6_B	LCD_DATA19_A	LCD_DATA20_A	LCD_DATA21_A	LCD_DATA22_A	LCD_DATA23_A	LCD_DATA2_B	LCD_DATA3_B	LCD_DATA4_B	LCD_DATA5_B	LCD_DATA0_B	LCD_DATA1_B
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		—	—	—	—	—	IRQ8	IRQ9	IRQ10	IRQ11	—	—	—	—	—	—	—
DSCR[1:0] bits	Drive capability control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
176 pins product		—	—	—	—	—	✓	✓	✓	✓	✓	—	—	—	—	—	—
176 pins w/o MIPI product		✓	✓	—	—	—	✓	✓	✓	✓	✓	—	—	—	—	—	—

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance“_A”, “_B”, or “_C”, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Table 19.16 Register settings for input/output pin function (PORTA) (1 of 2)

PSEL [4:0]settings	Function	Pin															
		PA00	PA01	PA02	PA03	PA04	PA05	PA06	PA07	PA08	PA09	PA10	PA11	PA12	PA13	PA14	PA15
00000b(value after reset)	Hi-Z/ JTAG/SWD	Hi-Z															
00010b	GPT*2	—	—	—	—	—	—	—	—	GTETRGD	—	—	—	—	—	—	—
00011b	GPT*2	—	—	—	—	—	—	GTIOC 7B	GTIOC 7A	—	—	—	GTIOC 6A	GTIOC 6B	—	—	—
00100b	SCI*1	—	—	RXD2_C/ MISO2_C/ SCL2_C	TXD2_C/ MOSI2_C/ SDA2_C	SCK2_C	CTS2_RTS2_C/ SS2_C	CTS2_C	—	—	—	—	—	—	—	—	—
00101b	SCI*1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXD9_C/ MOSI9_C/ SDA9_C	RXD9_C/ MISO9_C/ SCL9_C
01011b	BUS	D07 [A07/ D07]/ DQ07	RD	D26/ DQ26	D25/ DQ25	D24/ DQ24	D23/ DQ23	D22/ DQ22	D21/ DQ21	D15 [A15/ D15]/ DQ15	EBCLK/ SDCLK	WR/ WR0/ DQM0	WR2/ BC2/ DQM2	D16/ DQ16	D17/ DQ17	D18/ DQ18	D19/ DQ19
01101b	SCI	—	—	—	—	DE2	DE2	—	—	—	—	—	—	—	—	—	—

Table 19.16 Register settings for input/output pin function (PORTA) (2 of 2)

PSEL [4:0]settings	Function	Pin															
		PA00	PA01	PA02	PA03	PA04	PA05	PA06	PA07	PA08	PA09	PA10	PA11	PA12	PA13	PA14	PA15
11001b	GLCDC	LCD_CLK_A	LCD_TCON0_A	—	—	—	—	—	—	LCD_TCON3_A	LCD_TCON2_A	LCD_TCON1_A	—	—	—	—	—
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		—	—	—	—	—	—	—	—	IRQ6	IRQ5	IRQ4	—	—	—	—	—
DSCR[1:0] bits	Drive capability control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H/HH	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
176 pins product		✓	✓	—	—	—	—	—	—	✓	✓	✓	—	—	—	—	—
176 pins w/o MIPI product		✓	✓	—	—	—	—	—	—	✓	✓	✓	—	—	—	—	—

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance "_A", "_B", or "_C", to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Table 19.17 Register settings for input/output pin function (PORTB)

PSEL[4:0]settings	Function	Pin							
		PB00	PB01	PB02	PB03	PB04	PB05	PB06	PB07
00000b(value after reset)	Hi-Z/ JTAG/SWD	Hi-Z							
00011b	GPT*2	—	—	—	—	—	—	GTIOC9A	GTIOC9B
00101b	SCI*1	SCK1_B	CTS1_RTS1_B/SS1_B	—	—	—	—	—	—
01011b	BUS	—	ALE	—	—	—	—	—	—
01110b	SCI	DE1	DE1	—	—	—	—	—	—
01111b	CEU	—	VIO_D11	—	—	—	—	—	—
10100b	USBHS	USBHS_VBUSEN	USBHS_VBUS	—	—	—	—	—	—
11001b	GLCDC	LCD_DATA22_B	LCD_DATA18_B	LCD_DATA21_B	LCD_DATA20_B	LCD_DATA19_B	LCD_DATA17_B	LCD_DATA16_B	LCD_DATA15_B
ASEL bit		—	—	—	—	—	—	—	—
ISEL bit		—	—	—	—	—	—	—	—
DSCR[1:0] bits	Drive capability control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open- drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	✓	✓	✓	✓	✓
224 pins w/o MIPI product		✓	✓	✓	✓	✓	✓	✓	✓
176 pins product		✓	✓	—	—	—	—	—	—
176 pins w/o MIPI product		✓	✓	—	—	—	—	—	—

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance "_A", "_B", or "_C", to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

20. Port Output Enable for GPT (POEG)

20.1 Overview

The Port Output Enable (POEG) function can place the General PWM Timer (GPT) output pins in the output disable state in one of the following ways:

- Input level detection of the GTETRGn (n = A to D) pins
- Output-disable request from the GPT
- Comparator interrupt request detection
- Oscillation stop detection of the clock generation circuit
- Register settings

The GTETRGn (n = A to D) pins can be used as GPT external trigger input pins.

Table 20.1 lists the POEG specifications, Figure 20.1 shows a block diagram, and Table 20.2 lists the input pins.

Table 20.1 POEG specifications

Parameter	Specifications
Output-disable control through input level detection	The GPT output pins can be disabled when a GTETRGn rising edge or high level is sampled after polarity and filter selection.
Output-disable request from the GPT	When the GTIOCxA pin and the GTIOCxB pin are driven to an active level simultaneously, the GPT generates an output-disable request to the POEG. Through reception of these requests, the POEG can control whether the GTIOCxA and GTIOCxB pins are output-disabled.
Output-disable control through the comparator (ACMPHS) interrupt request detection	The GPT output pins can be disabled when an interrupt request is generated by a change in the output results of any of the comparators.
Output-disable control through oscillation stop detection	The GPT output pins can be disabled when oscillation of the clock generation circuit stops.
Output-disable control by software (registers)	The GPT output pins can be disabled by modifying the register settings.
Interrupt	Interrupts can be generated by detecting the input level of external trigger input pins (GTETRGn pins). Interrupts can be generated when all GPT or ACMPHS output pins are driven to an active level simultaneously.
External trigger output to the GPT	The GTETRGn signals can be output to the GPT after polarity and filter selection. (count start, count stop, count clear, up-count, down-count, or input capture function)
Noise filtering	For input from the GTETRGn pins, PCLKB/1, PCLKB/8, PCLKB/32, or PCLKB/128 can be selected as the noise filtering clock. (Filtering is performed by sampling the input signals three times using the selected clock.) Positive or negative polarity can be selected for any of the GTETRGn input pins. Signal state after polarity and filter selection can be monitored.
Module-stop function	Module-stop state can be set for each groups to reduce power consumption.
TrustZone Filter	Security and Privilege attribution can be set for each groups.

Note: n = A to D, x = 0 to 13

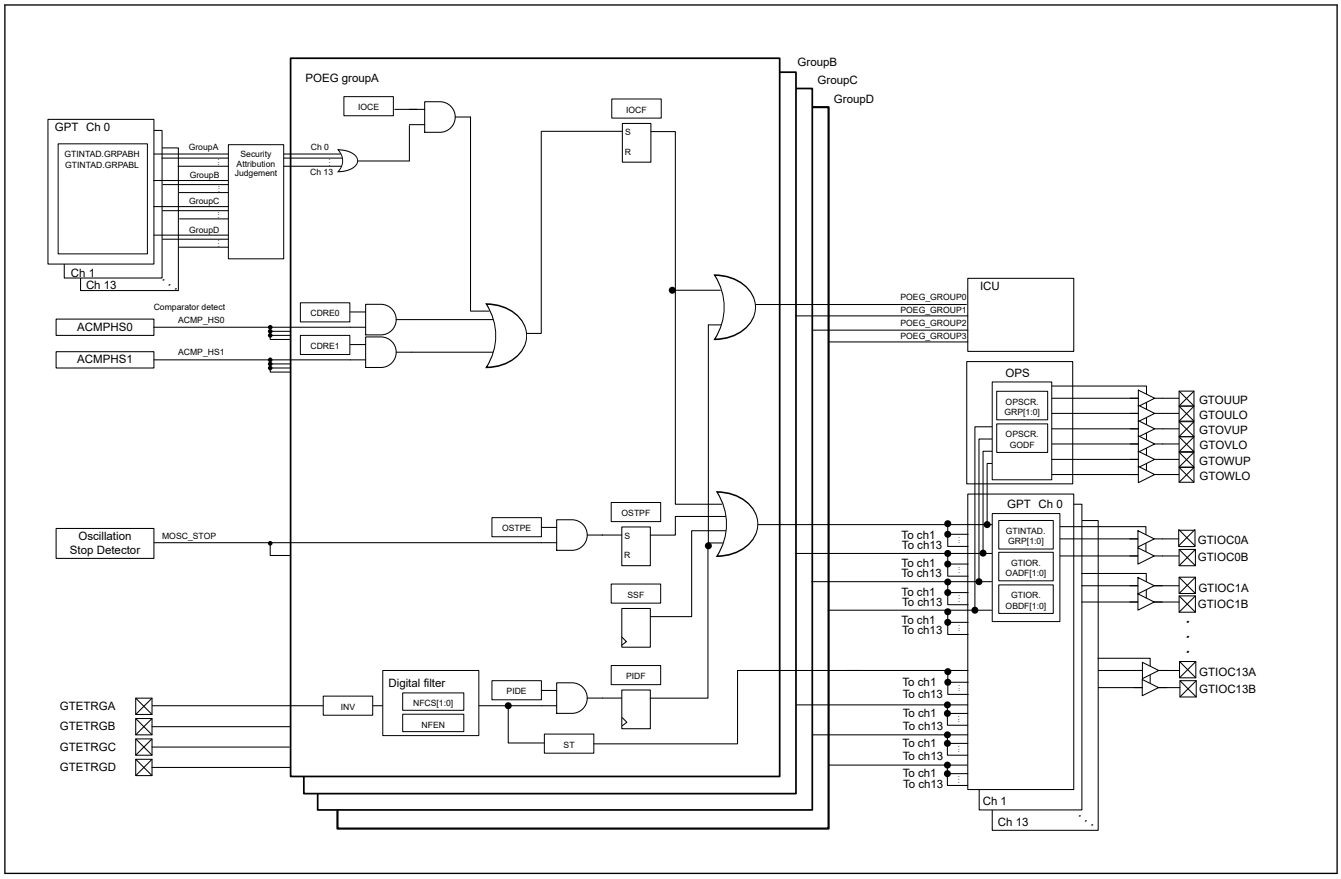


Figure 20.1 POEG block diagram

Table 20.2 POEG input pins

Pin name	I/O	Description
GTETRGA	Input	GPT output pin output-disable request signal or GPT external trigger input pin A
GTETRGB	Input	GPT output pin output-disable request signal or GPT external trigger input pin B
GTETRGC	Input	GPT output pin output-disable request signal or GPT external trigger input pin C
GTETRGD	Input	GPT output pin output-disable request signal or GPT external trigger input pin D

20.2 Register Descriptions

20.2.1 POEGGn : POEG Group n Setting Register (n = A to D)

Base address: POEG = 0x4021_2000
POEG_NS = 0x5021_2000

Offset address: 0x000 (POEGGA)
0x100 (POEGGB)
0x200 (POEGGC)
0x300 (POEGGD)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFCS[1:0]		NFEN	INV	—	—	—	—	—	—	—	—	—	—	—	ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CDRE1	CDRE0	—	OSTPE	IOCE	PIDE	SSF	OSTPF	IOCF	PIDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIDF	Port Input Detection Flag 0: No output-disable request from the GTETRGn pin occurred 1: Output-disable request from the GTETRGn pin occurred.	R/W*1
1	IOCF	Detection Flag for GPT or ACMPHS Output-Disable Request 0: No output-disable request from GPT or comparator interrupt occurred. 1: Output-disable request from GPT or comparator interrupt occurred.	R/W*1
2	OSTPF	Oscillation Stop Detection Flag 0: No output-disable request from oscillation stop detection occurred 1: Output-disable request from oscillation stop detection occurred	R/W*1
3	SSF	Software Stop Flag 0: No output-disable request from software occurred 1: Output-disable request from software occurred	R/W
4	PIDE	Port Input Detection Enable 0: Disable output-disable requests from the GTETRGn pins 1: Enable output-disable requests from the GTETRGn pins	R/W*2
5	IOCE	Enable for GPT Output-Disable Request 0: Disable output-disable requests from GPT 1: Enable output-disable requests from GPT	R/W*2
6	OSTPE	Oscillation Stop Detection Enable 0: Disable output-disable requests from oscillation stop detection 1: Enable output-disable requests from oscillation stop detection	R/W*2
7	—	This bit is read as 0. The write value should be 0.	R/W
8	CDRE0	ACMP_HS0 Enable 0: Disable output-disable request from ACMPHS0 1: Enable output-disable request from ACMPHS0	R/W*2
9	CDRE1	ACMP_HS1 Enable 0: Disable output-disable request from ACMPHS1 1: Enable output-disable request from ACMPHS1	R/W*2
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	ST	GTETRGn Input Status Flag 0: GTETRGn input after filtering was 0 1: GTETRGn input after filtering was 1	R
27:17	—	These bits are read as 0. The write value should be 0.	R/W
28	INV	GTETRGn Input Reverse 0: Input GTETRGn as-is 1: Input GTETRGn in reverse	R/W
29	NFEN	Noise Filter Enable 0: Disable noise filtering 1: Enable noise filtering	R/W
31:30	NFCS[1:0]	Noise Filter Clock Select 0 0: Sample GTETRGn pin input level three times every PCLKB 0 1: Sample GTETRGn pin input level three times every PCLKB/8 1 0: Sample GTETRGn pin input level three times every PCLKB/32 1 1: Sample GTETRGn pin input level three times every PCLKB/128	R/W

Note: S-TYPE3, P-TYPE3
Note 1. Only 0 can be written to clear the flag.
Note 2. Can be modified only once after a reset.

The POEGGn (n = A to D) registers control the output-disable state of the GPT pins, interrupts, and the external trigger input to the GPT.

In the descriptions, POEGGn represents the POEGGn (n = A to D) registers.

20.3 Output-Disable Control Operation

If any of the following conditions is satisfied, the GTIOCxA, GTIOCxB, and the 3-phase PWM output for BLDC motor control pins can be set to output-disable:

- Input level or edge detection of the GTETRGn pins
  When POEGGn.PIDE is 1, the POEGGn.PIDF flag is set to 1.
- Output-disable request from the GPT
  When POEGGn.IOCE is 1, the POEGGn.IOCF flag is set to 1 if the disable request is enabled by GTINTAD.
  The GTINTAD.GRPABH and GTINTAD.GRPABL settings apply to the group selected by the GPT register GTINTAD.GRP[1:0] or OPSCR.GRP[1:0].

  Note: The disable request is valid only when the security attributions of GPT and POEG are same. The disable request from GPT with a security attribution different from the security attribution of POEG is invalidated by the security attribution judgment.
- Comparator (ACMPHS) interrupt request detection
  Comparator interrupt detection is activated when any of the POEGGn.CDREi (i = 0, 1) is 1. When the associated comparator interrupt is generated, the GPT output pins are disabled. POEGGn.IOCF indicates the detection status.
- Oscillation stop detection for the clock generation circuit
  While POEGGn.OSTPE is 1, the halt status of the main clock oscillator is detected and the POEGGn. OSTPF flag is set to 1.
- SSF bit setting
  When POEGGn.SSF is set to 1, the GPT and PWM output are disabled.

The output-disable state is controlled in the GPT module. The output-disable of the GTIOCxA and GTIOCxB pins is set in the GTINTAD.GRP[1:0], GTIOR.OADF[1:0], and GTIOR.OBDF[1:0] bits in GPTx. The output-disable of the 3-phase PWM output for BLDC motor control pins is set in the OPSCR.GRP[1:0] bits and OPSCR.GODF bit in GPT_OPS.

20.3.1 Pin Input Level Detection Operation

If the input conditions set in POEGGn.PIDE, POEGGn.NFCS[1:0], POEGGn.NFEN, and POEGGn.INV occur on the GTETRGn pins, the GPT output pins are output-disabled.

20.3.1.1 Digital Filter

Figure 20.2 shows high-level detection by the digital filter. When a high level associated with the POEGGn.INV polarity setting is detected three times consecutively with the sampling clock selected in POEGGn.NFCS[1:0] and POEGGn.NFEN, the detected level is recognized as high, and the GPT output pins are output-disabled. If even one low level is detected during this interval, the detected level is not recognized as high. In addition, in an interval where the sampling clock is not output, changes of the levels on the GTETRGn pins are ignored.

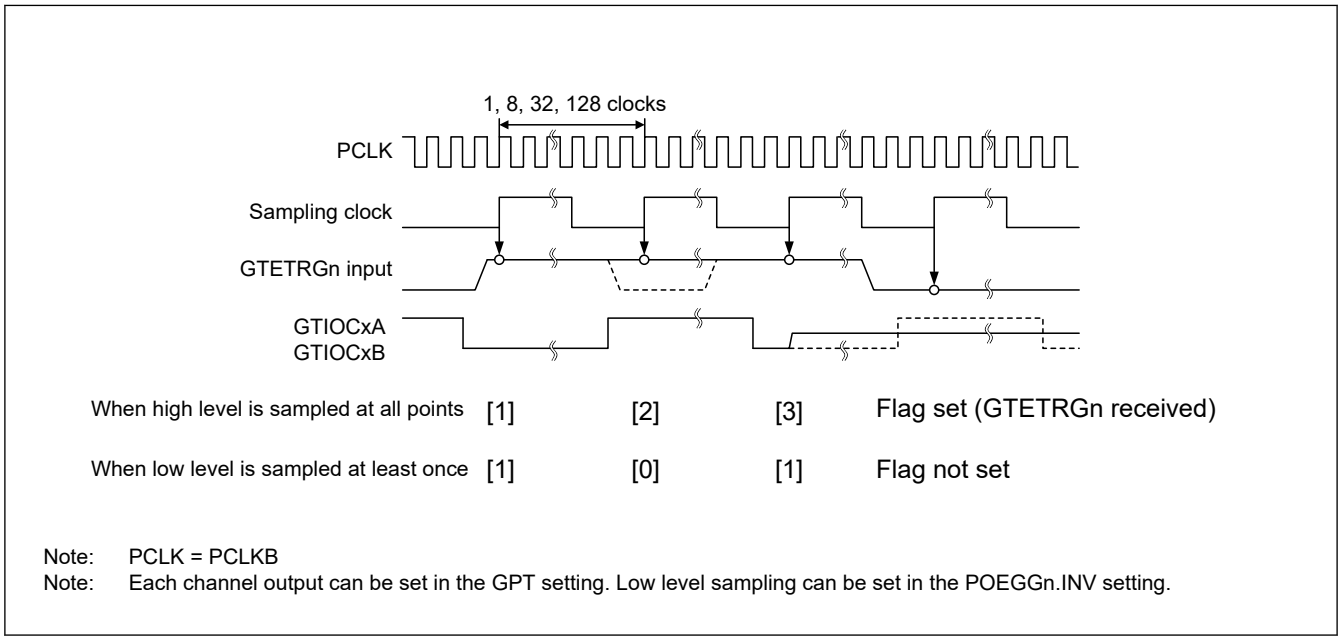


Figure 20.2 Example of digital filter operation

20.3.2 Output-Disable Requests from the GPT

For details on the operation, see the description for GTIOC Pin Output Negate Control in section 21, General PWM Timer (GPT).

20.3.3 Comparator Interrupt Detection

If POEGGn.CDREi (i = 0, 1) is 1 when an associated comparator interrupt request is generated, the GPT output pins are output-disabled for each group. The status flag is POEGGn.IOCF which is shared with GPT output-disable detection.

20.3.4 Output-Disable Control Using Detection of Stopped Oscillation

When the oscillation stop detection function in the clock generation circuit detects stopped oscillation while POEGGn.OSTPE is 1, the GPT output pins are output-disabled for each group.

20.3.5 Output-Disable Control Using Registers

The GPT output pins can be directly controlled by writing 1 to the Software Stop flag, POEGGn.SSF.

20.3.6 Release from Output-Disable

To release the GPT output pins placed in the output-disable state, either return them to their initial state with a reset or clear all of the following flags:

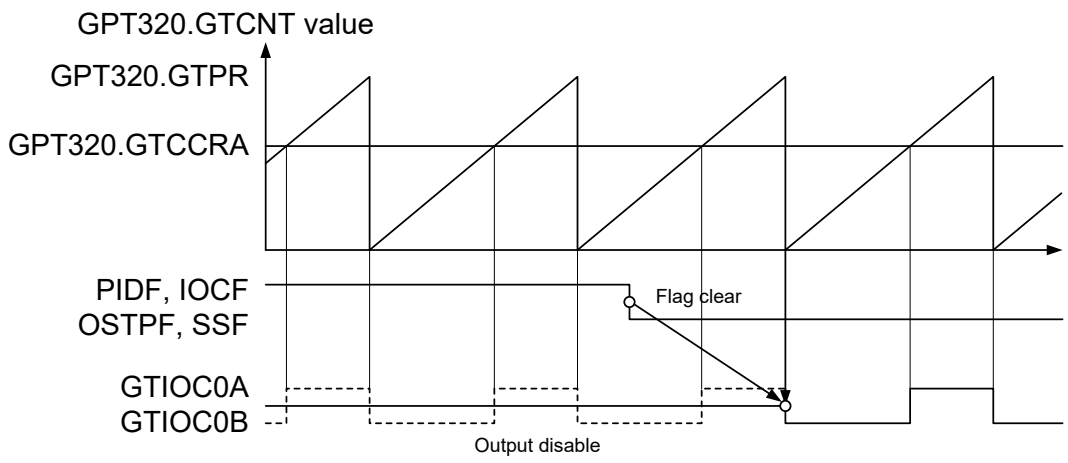
- POEGGn.PIDF
- POEGGn.IOCF
- POEGGn.OSTPF
- POEGGn.SSF

Writing 0 to the POEGGn.PIDF flag is ignored (the flag is not cleared) if the external input pins, GTETRGn are not disabled and the POEGGn.ST bit is not set to 0.

Writing 0 to the POEGGn.IOCF flag is valid (the flag is cleared) only if all of the GTST.OABHF and GTST.OABLF flags in the GPT are set to 0.

Writing 0 to the POEGGn.OSTPF flag is ignored (the flag is not cleared) if the OSTDSR.OSTDF flag in the clock generation circuit is not set to 0. In addition, when the flag set and release occur at the same time, the flag set takes precedence.

Figure 20.3 shows the release timing for output-disable. The output-disable is released at the beginning of the next count cycle of the GPT after the flag is cleared.



Note: The base address of the registers are the same.

Figure 20.3 Output-disable release timing for GPT pin outputs

20.4 Interrupt Sources

The POEG generates an interrupt request for the following factors:

- Output-disable control by the input level detection
- Output-disable request from the GPT or the comparator (ACMPHS) interrupt request detection

Table 20.3 lists the conditions for interrupt requests.

Table 20.3 Interrupt sources and conditions

Interrupt source	Symbol	Associated flag	Trigger conditions
POEG group A interrupt	POEG_GROUPA	POEGGA.IOCF	An output-disable request from a GPT disable request occurred
			An output-disable request from ACMPHS interrupt occurred
		POEGGA.PIDF	An output-disable request from the GTETRGA pin occurred
POEG group B interrupt	POEG_GROUPB	POEGGB.IOCF	An output-disable request from a GPT disable request occurred
			An output-disable request from ACMPHS interrupt occurred
		POEGGB.PIDF	An output-disable request from the GTETRGB pin occurred
POEG group C interrupt	POEG_GROUPC	POEGGC.IOCF	An output-disable request from a GPT disable request occurred
			An output-disable request from ACMPHS interrupt occurred
		POEGGC.PIDF	An output-disable request from the GTETRGC pin occurred
POEG group D interrupt	POEG_GROUPD	POEGGD.IOCF	An output-disable request from a GPT disable request occurred
			An output-disable request from ACMPHS interrupt occurred
		POEGGD.PIDF	An output-disable request from the GTETRGD pin occurred

20.5 External Trigger Output to the GPT

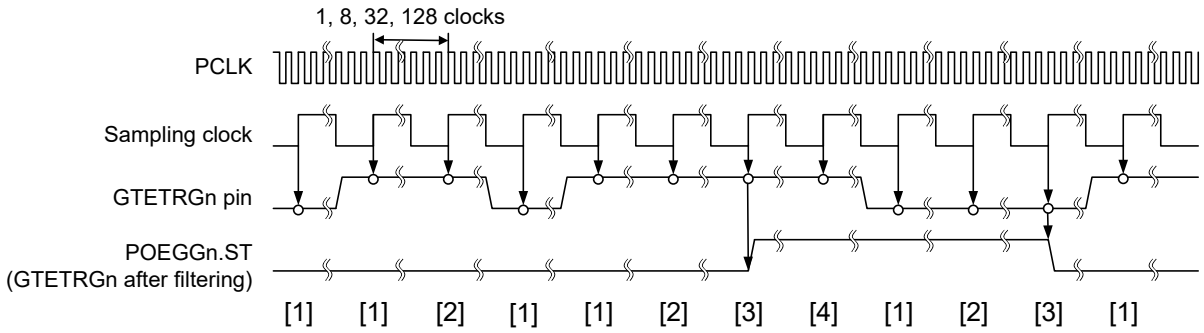
The POEG outputs signals generated by filtering and level detection of GTETRGn pins input signals as the GPT operation trigger signal for the following:

- Count start
- Count stop

- Count clear
- Up-count
- Down-count
- Input capture

For the POEGGn.INV polarity setting signal, when the same level is input three times continuously with the sampling clock selected in POEGGn.NFCS[1:0] and POEGGn.NFEN, that value is output. Set the control registers the same as for the input level detection operation described in section 20.3.1. Pin Input Level Detection Operation. The state after filtering can be monitored in POEGGn.ST.

Figure 20.4 shows the output timing of an external trigger to the GPT.



Note: PCLK = PCLKB
Note: Each channel output can be set in the GPT settings. Polarity can be reversed in POEGGn.INV.

Figure 20.4 Output timing of external trigger to the GPT

20.6 Usage Notes

20.6.1 Transition to Software Standby Mode

When using the POEG, do not invoke Software Standby mode. In this mode, the POEG stops and therefore output disable of the pins cannot be controlled.

20.6.2 Specifying Pins Associated with the GPT

The POEG controls output-disable only when a pin is associated with the GPT in the PmnPFS.PMR and PmnPFS.PSEL settings. When the pin is specified as a general I/O pin, the POEG does not perform output-disable control.

21. General PWM Timer (GPT)

21.1 Overview

The General PWM Timer (GPT) is a 32-bit timer with GPT32 × 8 channels and a 16-bit timer with GPT16 × 6 channels. PWM waveforms can be generated by controlling the up-counter, down-counter, or the up- and down-counter. This GPT provides the A/D conversion start request function. The GPT can also be used as a general-purpose timer.

Table 21.1 lists the GPT specifications, Table 21.2 shows the GPT functions, and Figure 21.1 shows a block diagram.

Table 21.1 GPT specifications

Parameter	Description
Functions	• 32 bits × 8 channels (GPT32n (n = 0 to 7)) • 16 bits × 6 channels (GPT16m (m = 8 to 13)) • Up-counting or down-counting (saw waves) or up/down-counting (triangle waves) for each counter • Clock sources independently selectable for each channel • Two input/output pins per channel • Two output compare/input capture registers per channel • For the two output compare/input capture registers of each channel, four registers are provided as buffer registers and are capable of operating as comparison registers when buffering is not in use • In output compare operation, buffer switching can be at crests or troughs, enabling the generation of laterally asymmetric PWM waveforms • Registers for setting up frame cycles in each channel with capability for generating interrupts at overflow or underflow • Generation of dead times in PWM operation • Synchronous starting, stopping and clearing counters for arbitrary channels • Count start, count stop, count clear, up-count, down-count, or input capture operation in response to a maximum of 8 ELC events • Count start, count stop, count clear, up-count, down-count, or input capture operation in response to the status of two input pins • Count start, count stop, count clear, up-count, down-count, or input capture operation in response to a maximum of 4 external triggers • Control output disable request from the POEG • A/D conversion start request generation function • PWM waveform for controlling brushless DC motors can be generated • Compare match A to F event and overflow/underflow event can be output to the ELC • Enables the noise filter for input capture • Period count function • Logical operation between the channel output • Bus clock: PCLKA, Core clock: PCLKD • Frequency ratio: PCLKA:PCLKD = 1:N (N = 1/2/4/8/16/32/64)

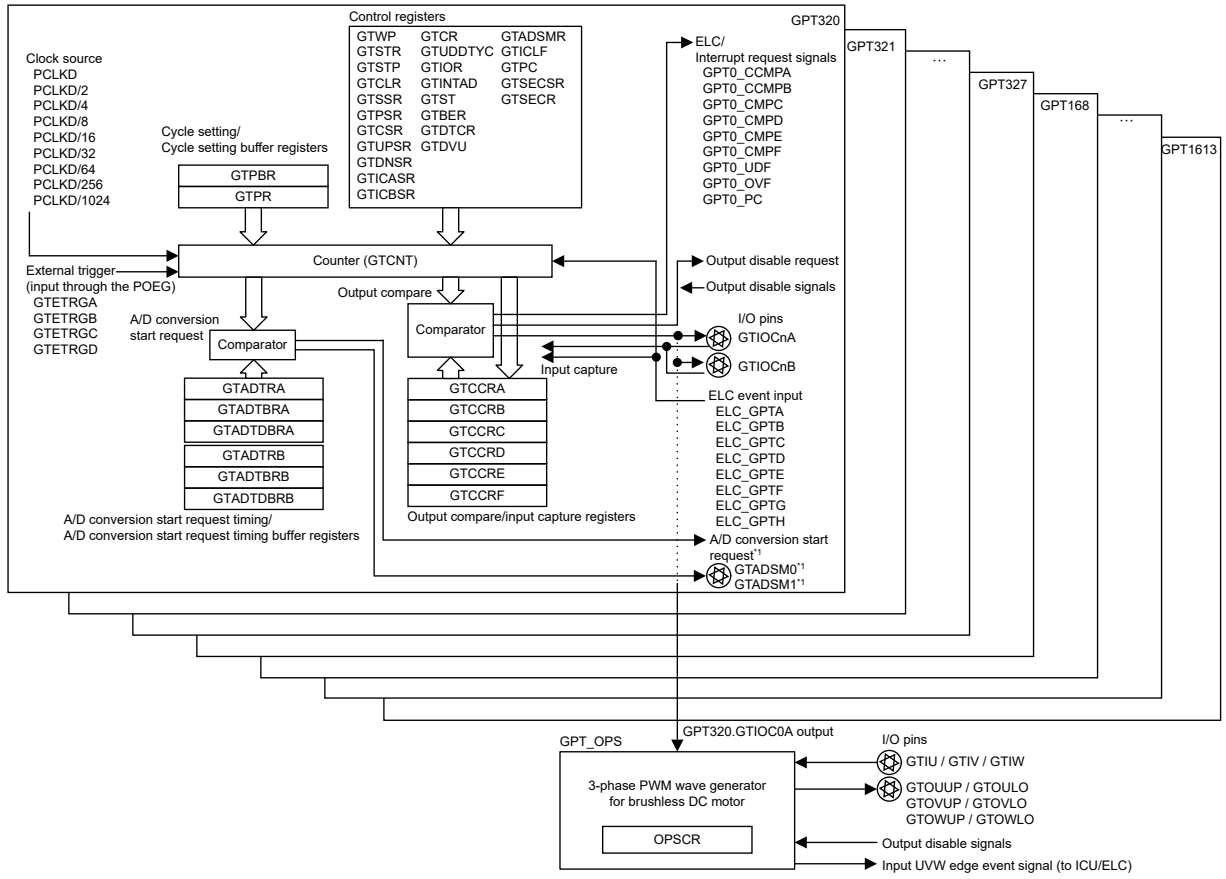
Table 21.2 GPT functions (1 of 2)

Parameter	Description
Count clock	PCLKD PCLKD/2 PCLKD/4 PCLKD/8 PCLKD/16 PCLKD/32 PCLKD/64 PCLKD/256 PCLKD/1024 GTETRGA, GTETRGB, GTETRGC, GTETRGD
Output compare/input capture registers (GTCCR)	GTCCRA GTCCRB
Compare/buffer registers	GTCCRC GTCCRD GTCCRE GTCCRF
Cycle setting buffer register	GTPBR
I/O pins	GTIOCnA GTIOCnB (n = 0 to 13)

Table 21.2 GPT functions (2 of 2)

Parameter		Description
External trigger input pin[*1]		GTETRGA GTETRGB GTETRGC GTETRGD
Counter clear sources		GTPR register compare match Input capture Input pin status ELC event input GTETRGn (n = A to D) pin input
Period count function		Available (GPT32n (n = 0 to 3), GPT16m (m = 8 to 10))
Compare match output	Low output	Available
	High output	Available
	Toggle output	Available
Input capture function		Available
Automatic addition of dead time		Available (no dead time buffer)
PWM mode		Available
Phase count function		Available
Buffer operation		Simultaneous operation disable control for multiple channels
One-shot operation		Available
DMAC/DTC activation		All the interrupt sources
A/D conversion start request		Compare match of GTADTRA or GTADTRB register
Interrupt sources		9 sources ● GTCCRA compare match/input capture(GPTn_CCMPA) ● GTCCRB compare match/input capture(GPTn_CCMPB) ● GTCCRC compare match(GPTn_CMPC) ● GTCCRD compare match(GPTn_CMPD) ● GTCCRE compare match(GPTn_CMPE) ● GTCCRF compare match(GPTn_CMPF) ● GTCNT overflow (GTPR compare match) (GPTn_OVF) ● GTCNT underflow (GPTn_UDF) ● GTPC count stop(GPTx_PC) (x = 0 to 3, 8 to 10)
Event linking (ELC) function		Available[*2]
Noise filtering function		Available
Logical operation between the channel output		Available
Module-stop function		Module-stop state can be set for each channels to reduce power consumption
TrustZone Filter		Security and Privilege attribution can be set for each channels

Note 1. GTETRGn connects to GPT through the POEG module. Therefore, to use the GPT function, supply the POEG clock by clearing the MSTPCRD.MSTPDn (n = 11 to 14) bit.
Note 2. See section 21.6. Operations Linked by ELC.



Note 1. This function is only available in GPT32n (n = 0 to 7).

Figure 21.1 GPT block diagram (Saw-wave PWM mode, Saw-wave one-shot pulse mode, Triangle-wave PWM mode 1,2,3)

Figure 21.2 shows an example using multiple GPTs.

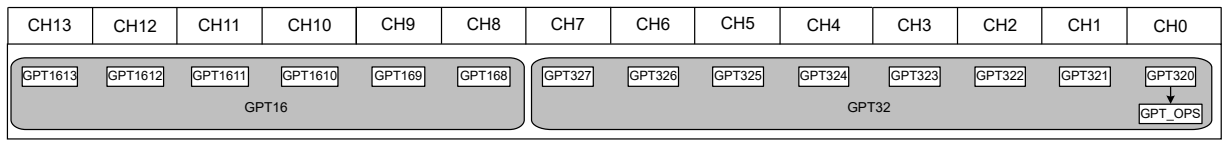


Figure 21.2 Association between GPT channels and module names

Table 21.3 lists the I/O pins.

Table 21.3 GPT I/O pins (1 of 2)

Channel	Pin name	I/O	Function
Common	GTETRGx	Input	External trigger input pin x (input through the POEG)
	GTADSM0	Output	A/D conversion start request monitor 0 output pin
	GTADSM1	Output	A/D conversion start request monitor 1 output pin
GPT32n	GTIOCnA	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOCnB	I/O	GTCCRB register input capture input/output compare output/PWM output pin

Table 21.3 GPT I/O pins (2 of 2)

Channel	Pin name	I/O	Function
GPT16m	GTIOCmA	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOCmB	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT_OPS	GTIU	Input	Hall sensor input pin U
	GTIV	Input	Hall sensor input pin V
	GTIW	Input	Hall sensor input pin W
	GTOUUP	Output	3-phase PWM output for BLDC motor control (positive U-phase)
	GTOULO	Output	3-phase PWM output for BLDC motor control (negative U-phase)
	GTOVUP	Output	3-phase PWM output for BLDC motor control (positive V-phase)
	GTOVLO	Output	3-phase PWM output for BLDC motor control (negative V-phase)
	GTOWUP	Output	3-phase PWM output for BLDC motor control (positive W-phase)
	GTOWLO	Output	3-phase PWM output for BLDC motor control (negative W-phase)

Note: x: A to D
n: 0 to 7
m: 8 to 13

21.2 Register Descriptions

21.2.1 GTWP : General PWM Timer Write-Protection Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PRKEY[7:0]								—	—	—	CMNWP	CLRWP	STPWP	STRWP	WP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WP	Register Write Disable 0: Write to the register enabled 1: Write to the register disabled	R/W
1	STRWP	GTSTR.CSTRT Bit Write Disable 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
2	STPWP	GTSTP.CSTOP Bit Write Disable 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
3	CLRWP	GTCLR.CCLR Bit Write Disable 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
4	CMNWP	Common Register Write Disabled 0: Write to the register is enabled 1: Write to the register is disabled	R/W

Bit	Symbol	Function	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	GTWP Key Code When 0xA5 is written to these bits, writing to the WP, STRWP, STPWP, CLRWP, and CMNWP bits are permitted. These bits are read as 0.	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

GTWP enables or disables writing to registers to prevent accidental modification. Protection by the GTWP register is only for the writes by the CPU.GTWP does not protect registers from updates that occur in association with CPU writes.

WP bit (Register Write Disable)

The following is a list of write enabled or disabled registers:

GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTADTRA, GTADTBRA, GTADTDBRA, GTADTRB, GTADTBRB, GTADTDBRB, GTDTCR, GTDVU, GTADSMR, GTICLF, GTPC.

STRWP bit (GTSTR.CSTRT Bit Write Disable)

The STRWP bit enables or disables starting the updating of counter values by writing to the CSTRTn bit (n = 0 to 13) corresponding to a channel number in the GTSTR register.

The bit position of each CSTRTn bit in the GTSTR register is allocated to the channel with the corresponding number, and writing to the GTSTR register for any channel results in writing to the registers of all channels. The STRWP bit for each channel does not control writing but only controls updating of the CSTRT bit for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the CSTRT bits of a channel for which the setting of the STRWP bit is 1(disabling writing), the CSTRT bit for the given channel is not updated, but the CSTRT bits corresponding to channel for which the setting of the STRWP bit is 0 (enabling writing) are updated. For example, when the setting of the GPT320.GTWP.STRWP bit is 0 (enabling writing), writing 1 to the GPT321.GTSTR.CSTRT0 bit when its current setting is 0 causes the value to be updated, and the GPT320.GTCNT counter starts to run. When the setting of the GPT320.GTWP.STRWP bit is 1 (disabling writing), writing 1 to the GPT321.GTSTR.CSTRT0 bit when its current setting is 0 leaves the bit with the value 0, and the GPT320.GTCNT counter does not run.

If you want to protect all bits in the GTSTR register from being updated, set the STRWP bits of all channels to 1.

STPWP bit (GTSTP.CSTOP Bit Write Disable)

The STPWP bit enables or disables starting the updating of counter values by writing to the CSTOPn bit (n = 0 to 13) corresponding to a channel number in the GTSTP register.

The bit position of each CSTOPn bit in the GTSTP registers is allocated to the channel with the corresponding number, and the writing to the GTSTP register for any channel results in writing to the registers of all channels. The STPWP bit for each channel does not control writing but only controls updating of the CSTOP bit for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the CSTOP bits of a channel for which the setting of the STPWP bit is 1(disabling writing), the CSTOP bit for the given channel is not updated, but the CSTOP bits corresponding to channel for which the setting of the STPWP bit is 0 (enabling writing) are updated. For example, when the setting of the GPT320.GTWP.STPWP bit is 0 (enabling writing), writing 1 to the GPT321.GTSTP.CSTOP0 bit when its current setting is 0 causes the value to be updated, and the GPT320.GTCNT counter is stopped. When the setting of the GPT320.GTWP.STPWP bit is 1 (disabling writing), writing 1 to the GPT321.GTSTP.CSTOP0 bit when its current setting is 0 leaves the bit with the value 0, and the GPT320.GTCNT counter is not stopped.

If you want to protect all bits in the GTSTP register from being updated, set the STPWP bits of all channels to 1.

CLRWP bit (GTCLR.CCLR Bit Write Disable)

CLRWP bit enables or disables starting the updating of counter values by writing to the CCLRn bit (n = 0 to 13) corresponding to a channel number in the GTCLR register.

The bit position of each CCLRn bit in the GTCLR registers is allocated to the channel with the corresponding number, and the writing to the GTCLR register for any channel results in writing to the registers of all channels. The CLRWP bit for each channel does not control writing but only controls updating of the CCLR bit for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the CCLR bits of a channel for which the setting of the CLRWP bit is 1(disabling writing), the CCLR bit for the given channel is not updated, but the CCLR bits corresponding to channel for which the setting of the CLRWP bit is 0 (enabling writing) are updated. For example, when the setting of the GPT320.GTWP.CLRWP bit is 0 (enabling writing), writing 1 to the GPT321.GTCLR.CCLR0 bit when its current setting is 0 causes the value to be updated, and the GPT320.GTCNT counter is cleared. When the setting of the GPT320.GTWP.CLRWP bit is 1 (disabling writing), writing 1 to the GPT321.GTCLR.CCLR0 bit when its current setting is 0 leaves the bit with the value 0, and the GPT320.GTCNT counter is not cleared.

If you want to protect all bits in the GTCLR register from being updated, set the CLRWP bits of all channels to 1.

CMNWP bit (Common Register Write Disabled)

CMNWP bit enables or disables starting the updating of counter values by writing to the SECSELn bit (n = 0 to 13) corresponding to a channel number in the GTSECSR register or to the GTSECR register.

The bit position of each SECSEL bit in the GTSECSR registers is allocated to the channel with the corresponding number, and the writing to the GTSECSR register for any channel results in writing to the registers of all channels. Writing to the GTSECR register of any channel leads to writing to the registers of all channels. The CMNWP bit for each channel does not control writing but only controls updating of the SECSEL bit and the GTSECR register value for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the SECSEL bit and the GTSECR register value of a channel for which the setting of the CMNWP bit is 1(disabling writing), the SECSEL bit and the GTSECR register value for the given channel is not updated, but the SECSEL bit and the GTSECR register value corresponding to channel for which the setting of the CMNWP bit is 0 (enabling writing) are updated.

For example, when the setting of the GPT320.GTWP.CMNWP bit is 0 (enabling writing), writing to the GPT321.GTSECSR.SECSEL0 bit causes the value of the GPT320.GTSECSR.SECSEL0 bit to be updated. In the same way, writing to the GPT321.GTSECR register updates the value of the GPT320.GTSECR register. When the setting of the GPT320.GTWP.CMNWP bit is 1 (disabling writing), writing to the GPT321.GTSECSR.SECSEL0 bit does not cause the value of the GPT320.GTSECSR.SECSEL0 bit to be updated. In the same way, writing to the GPT321.GTSECR register does not update the value of the GPT320.GTSECR register.

If you want to protect all bits in the GTSECSR and GTSECR registers from being updated, set the CMNWP bits of all channels to 1.

PRKEY[7:0] bit (GTWP Key Code)

This bit controls whether the WP, STRWP, STPWP, CLRWP, and CMNWP bits can be overwritten.

21.2.2 GTSTR : General PWM Timer Software Start Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CSTRT13	CSTRT12	CSTRT11	CSTRT10	CSTRT9	CSTRT8	CSTRT7	CSTRT6	CSTRT5	CSTRT4	CSTRT3	CSTRT2	CSTRT1	CSTRT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	CSTRT0 to CSTRT13	Channel n GTCNT Count Start (n : the same as bit position value) 0: GTCNT counter not start 1: GTCNT counter start	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The GTSTR starts the GTCNT counter operation for each channel n, where n = 0 to 13.

The GTSTR bit number represents the channel number. The GTSTR register of each channel is shared by all of the channels. The GTCNT counter starts for the channel associated with the GTSTR bit number where 1 is written. Writing 0 has no effect on the status of GTCNT counter and the value of GTSTR register.

For the association between module names and channel numbers, see Figure 21.2.

Contrary to a security attribute for write or read access, if the security and privilege attributes set on each channel has the security violation or the privilege violation, the bit of the channel number that violates security cannot be written or read, and the read value of the bit is "0".

CSTRTn bits (Channel n GTCNT Count Start (n = 0 to 13))

The CSTRTn bits start channel n of the GTCNT counter operation. Writing to the GTSTR.CSTRTn bit (n = 0 to 13) has no effect unless the GTSSR.CSTRT bit is set to 1.

The read data shows the counter status of each channel (GTCR.CST bit). A value of 0 means the counter is stopped and 1 means the counter is running.

21.2.3 GTSTP : General PWM Timer Software Stop Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CSTOP13	CSTOP12	CSTOP11	CSTOP10	CSTOP9	CSTOP8	CSTOP7	CSTOP6	CSTOP5	CSTOP4	CSTOP3	CSTOP2	CSTOP1	CSTOP0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
13:0	CSTOP0 to CSTOP13	Channel n GTCNT Count Stop (n : the same as bit position value) 0: GTCNT counter not stop 1: GTCNT counter stop	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The GTSTP stops the GTCNT counter operation for each channel n, where n = 0 to 13.

The GTSTP bit number represents the channel number. The GTSTP register of each channel is shared by all the channels. The GTCNT counter stops for the channel associated with the GTSTP bit number where 1 is written. Writing 0 has no effect on the status of the GTCNT counter and the value of GTSTP register.

For the association between module names and channel numbers, see Figure 21.2.

Contrary to a security attribute for write or read access, if the security and privilege attributes set on each channel has the security violation or the privilege violation, the bit of the channel number that violates security cannot be written or read, and the read value of the bit is "0".

CSTOPn bits (Channel n GTCNT Count Stop (n = 0 to 13))

The CSTOPn bits stop channel n of the GTCNT counter operation. Writing to the GTSTP.CSTOPn bit (n = 0 to 13) has no effect unless the GTPSR.CSTOP bit is set to 1. The read data shows the counter status of each channel (invert of GTCR.CST bit). A value of 0 means the counter is running and 1 means the counter is stopped.

21.2.4 GTCLR : General PWM Timer Software Clear Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CCLR13	CCLR12	CCLR11	CCLR10	CCLR9	CCLR8	CCLR7	CCLR6	CCLR5	CCLR4	CCLR3	CCLR2	CCLR1	CCLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	CCLR0 to CCLR13	Channel n GTCNT Count Clear (n : the same as bit position value) 0: GTCNT counter is not cleared 1: GTCNT counter is cleared	W
31:14	—	These bits are read as 0. The write value should be 0.	W

The GTCLR is a write-only register that clears the GTCNT counter operation for each channel n, where n = 0 to 13.

The GTCLR bit number represents the channel number. The GTCLR register of each channel is shared by all the channels. The GTCNT counter is cleared for the channel associated with the GTCLR bit number where 1 is written. Writing 0 has no effect on the status of GTCNT counter.

For the association between module names and channel numbers, see Figure 21.2.

Contrary to a security attribute for write or read access, if the security and privilege attributes set on each channel has the security violation or the privilege violation, the bit of the channel number that violates security cannot be written or read, and the read value of the bit is "0".

CCLRn bits (Channel n GTCNT Count Clear (n = 0 to 13))

When the counting direction flag is set for decrement (GTST.TUCF flag = 0) with saw-wave mode selected in the GTCR.MD[2:0] bits, the value of the GTCNT counter becomes that of the corresponding GTPR register in response to writing 1 to the CCLRn bit. The value of the counter becomes 0x0000 0000 with other settings. These bits are read as 0.

21.2.5 GTSSR : General PWM Timer Start Source Select Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CSTRT	—	—	—	—	—	—	—	SSELCH	SSELCG	SSELCF	SSELCE	SSELCD	SSELCC	SSELCB	SSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SSCBFAH	SSCBFAL	SSCBRAH	SSCBRAL	SSCAFBH	SSCAFBL	SSCARBH	SSCARBL	SSGTRGDF	SSGTRGDR	SSGTRGCF	SSGTRGCR	SSGTRGBF	SSGTRGBR	SSGTRGAF	SSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SSGTRGAR	GTETRGA Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGA input 1: Counter start enabled on the rising edge of GTETRGA input	R/W
1	SSGTRGAF	GTETRGA Pin Falling Input Source Counter Start Enable 0: Counter start disabled on the falling edge of GTETRGA input 1: Counter start enabled on the falling edge of GTETRGA input	R/W
2	SSGTRGBR	GTETRGB Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGB input 1: Counter start enabled on the rising edge of GTETRGB input	R/W
3	SSGTRGBF	GTETRGB Pin Falling Input Source Counter Start Enable 0: Counter start disabled on the falling edge of GTETRGB input 1: Counter start enabled on the falling edge of GTETRGB input	R/W
4	SSGTRGCR	GTETRGC Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGC input 1: Counter start enabled on the rising edge of GTETRGC input	R/W
5	SSGTRGCF	GTETRGC Pin Falling Input Source Counter Start Enable 0: Counter start disabled on the falling edge of GTETRGC input 1: Counter start enabled on the falling edge of GTETRGC input	R/W
6	SSGTRGDR	GTETRGD Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGD input 1: Counter start enabled on the rising edge of GTETRGD input	R/W
7	SSGTRGDF	GTETRGD Pin Falling Input Source Counter Start Enable 0: Counter start disabled on the falling edge of GTETRGD input 1: Counter start enabled on the falling edge of GTETRGD input	R/W
8	SSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter start enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	SSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter start enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	SSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter start enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W

Bit	Symbol	Function	R/W
11	SSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter start enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	SSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter start enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	SSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter start enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	SSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter start enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	SSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter start enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	SSELCA	ELC_GPTA Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTA input 1: Counter start enabled at the ELC_GPTA input	R/W
17	SSELCB	ELC_GPTB Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTB input 1: Counter start enabled at the ELC_GPTB input	R/W
18	SSELCC	ELC_GPTC Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTC input 1: Counter start enabled at the ELC_GPTC input	R/W
19	SSELCD	ELC_GPTD Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTD input 1: Counter start enabled at the ELC_GPTD input	R/W
20	SSELCE	ELC_GPTE Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTE input 1: Counter start enabled at the ELC_GPTE input	R/W
21	SSELCF	ELC_GPTF Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTF input 1: Counter start enabled at the ELC_GPTF input	R/W
22	SSELCG	ELC_GPTG Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTG input 1: Counter start enabled at the ELC_GPTG input	R/W
23	SSELCH	ELC_GPTH Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTH input 1: Counter start enabled at the ELC_GPTH input	R/W
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	CSTRT	Software Source Counter Start Enable 0: Counter start disabled by the GTSTR register 1: Counter start enabled by the GTSTR register	R/W

Note: n = 0 to 13

The GTSSR sets the source to start the GTCNT counter.

Input from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

SSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Start Enable)

The SSGTRGAR bit enables or disables the GTCNT counter start on the rising edge of the GTETRGA pin input.

SSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Start Enable)

The SSGTRGAF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGA pin input.

SSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Start Enable)

The SSGTRGBR bit enables or disables the GTCNT counter start on the rising edge of the GTETRGB pin input.

SSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Start Enable)

The SSGTRGBF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGB pin input.

SSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Start Enable)

The SSGTRGCR bit enables or disables the GTCNT counter start on the rising edge of the GTETRGC pin input.

SSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Start Enable)

The SSGTRGCF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGC pin input.

SSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Start Enable)

The SSGTRGDR bit enables or disables the GTCNT counter start on the rising edge of the GTETRGD pin input.

SSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Start Enable)

The SSGTRGDF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGD pin input.

SSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Start Enable)

The SSCARBL bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

SSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Start Enable)

The SSCARBH bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

SSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Start Enable)

The SSCAFBL bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

SSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Start Enable)

The SSCAFBH bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

SSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Start Enable)

The SSCBRAL bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

SSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Start Enable)

The SSCBRAH bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

SSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Start Enable)

The SSCBFAL bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

SSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Start Enable)

The SSCBFAH bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

SSELCm bit (ELC_GPTm Event Source Counter Start Enable) (m = A to H)

The SSELCm bit enables or disables the GTCNT counter start at the ELC_GPTm event input.

CSTRT bit (Software Source Counter Start Enable)

The CSTRT bit enables or disables the GTCNT counter start by GTSTR register.

21.2.6 GTPSR : General PWM Timer Stop Source Select Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CSTOP	—	—	—	—	—	—	—	PSELCH	PSELCG	PSELCF	PSELCE	PSELCD	PSELCC	PSELCB	PSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PSCBFAH	PSCBFAL	PSCBRAH	PSCBRAL	PSCAFBH	PSCAFBL	PSCARBH	PSCARBL	PSGTRGDF	PSGTRGDR	PSGTRGCF	PSGTRGCR	PSGTRGBF	PSGTRGBR	PSGTRGAF	PSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PSGTRGAR	GTETRGA Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGA input 1: Counter stop enabled on the rising edge of GTETRGA input	R/W
1	PSGTRGAF	GTETRGA Pin Falling Input Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTETRGA input 1: Counter stop enabled on the falling edge of GTETRGA input	R/W
2	PSGTRGBR	GTETRGB Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGB input 1: Counter stop enabled on the rising edge of GTETRGB input	R/W
3	PSGTRGBF	GTETRGB Pin Falling Input Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTETRGB input 1: Counter stop enabled on the falling edge of GTETRGB input	R/W
4	PSGTRGCR	GTETRGC Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGC input 1: Counter stop enabled on the rising edge of GTETRGC input	R/W
5	PSGTRGCF	GTETRGC Pin Falling Input Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTETRGC input 1: Counter stop enabled on the falling edge of GTETRGC input	R/W
6	PSGTRGDR	GTETRGD Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGD input 1: Counter stop enabled on the rising edge of GTETRGD input	R/W
7	PSGTRGDF	GTETRGD Pin Falling Input Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTETRGD input 1: Counter stop enabled on the falling edge of GTETRGD input	R/W
8	PSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter stop enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W

Bit	Symbol	Function	R/W
9	PSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter stop enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	PSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter stop enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	PSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter stop enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	PSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter stop enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	PSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter stop enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	PSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter stop enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	PSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter stop enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	PSELCA	ELC_GPTA Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTA input 1: Counter stop enabled at the ELC_GPTA input	R/W
17	PSELCB	ELC_GPTB Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTB input 1: Counter stop enabled at the ELC_GPTB input	R/W
18	PSELCC	ELC_GPTC Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTC input 1: Counter stop enabled at the ELC_GPTC input	R/W
19	PSELCD	ELC_GPTD Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTD input 1: Counter stop enabled at the ELC_GPTD input	R/W
20	PSELCE	ELC_GPTE Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTE input 1: Counter stop enabled at the ELC_GPTE input	R/W
21	PSELCF	ELC_GPTF Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTF input 1: Counter stop enabled at the ELC_GPTF input	R/W
22	PSELCG	ELC_GPTG Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTG input 1: Counter stop enabled at the ELC_GPTG input	R/W
23	PSELCH	ELC_GPTH Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTH input 1: Counter stop enabled at the ELC_GPTH input	R/W

Bit	Symbol	Function	R/W
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	CSTOP	Software Source Counter Stop Enable 0: Counter stop disabled by the GTSTP register 1: Counter stop enabled by the GTSTP register	R/W

Note: n = 0 to 13

The GTPSR sets the source to stop the GTCNT counter.

Inputs from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

PSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Stop Enable)

The PSGTRGAR bit enables or disables the GTCNT counter stop on the rising edge of the GTETRGA pin input.

PSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Stop Enable)

The PSGTRGAF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGA pin input.

PSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Stop Enable)

PSGTRGBR bit enables or disables the GTCNT counter stop on the rising edge of the GTETRGB pin input.

PSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Stop Enable)

The PSGTRGBF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGB pin input.

PSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Stop Enable)

PSGTRGCR bit enables or disables the GTCNT counter stop on the rising edge of the GTETRGC pin input.

PSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Stop Enable)

The PSGTRGCF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGC pin input.

PSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Stop Enable)

PSGTRGDR bit enables or disables the GTCNT counter stop on the rising edge of the GTETRGD pin input.

PSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Stop Enable)

The PSGTRGDF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGD pin input.

PSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Stop Enable)

The PSCARBL bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

PSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Stop Enable)

The PSCARBH bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

PSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Stop Enable)

The PSCAFBL bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

PSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Stop Enable)

The PSCAFBH bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

PSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Stop Enable)

The PSCBRAL bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

PSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Stop Enable)

The PSCBRAH bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

PSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Stop Enable)

The PSCBFAL bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

PSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Stop Enable)

The PSCBFAH bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

PSELCm bit (ELCm Event Source Counter Stop Enable) (m = A to H)

The PSELCm bit enables or disables the GTCNT counter stop at the ELC_GPTm event input.

CSTOP bit (Software Source Counter Stop Enable)

The CSTOP bit enables or disables the GTCNT counter stop by the GTSTP register.

21.2.7 GTCSR : General PWM Timer Clear Source Select Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CCLR	—	—	—	—	—	—	—	CSELCH	CSELCG	CSELCF	CSELCE	CSELCD	CSELCC	CSELCB	CSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CSCBFAH	CSCBFAL	CSCBRAH	CSCBRAL	CSCAFBH	CSCAFBL	CSCARBH	CSCARBL	CSGTRGDF	CSGTRGDR	CSGTRGCF	CSGTRGCR	CSGTRGBF	CSGTRGBR	CSGTRGAF	CSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CSGTRGAR	GTETRGA Pin Rising Input Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTETRGA input 1: Counter clear enabled on the rising edge of GTETRGA input	R/W
1	CSGTRGAF	GTETRGA Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGA input 1: Counter clear enabled on the falling edge of GTETRGA input	R/W
2	CSGTRGBR	GTETRGB Pin Rising Input Source Counter Clear Enable 0: Disable counter clear on the rising edge of GTETRGB input 1: Enable counter clear on the rising edge of GTETRGB input	R/W
3	CSGTRGBF	GTETRGB Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGB input 1: Counter clear enabled on the falling edge of GTETRGB input	R/W
4	CSGTRGCR	GTETRGC Pin Rising Input Source Counter Clear Enable 0: Disable counter clear on the rising edge of GTETRGC input 1: Enable counter clear on the rising edge of GTETRGC input	R/W
5	CSGTRGCF	GTETRGC Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGC input 1: Counter clear enabled on the falling edge of GTETRGC input	R/W

Bit	Symbol	Function	R/W
6	CSGTRGDR	GTETRGD Pin Rising Input Source Counter Clear Enable 0: Disable counter clear on the rising edge of GTETRGD input 1: Enable counter clear on the rising edge of GTETRGD input	R/W
7	CSGTRGDF	GTETRGD Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGD input 1: Counter clear enabled on the falling edge of GTETRGD input	R/W
8	CSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter clear enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	CSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter clear enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	CSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter clear enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	CSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter clear enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	CSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter clear enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	CSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter clear enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	CSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter clear enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	CSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter clear enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	CSELCA	ELC_GPTA Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTA input 1: Counter clear enabled at the ELC_GPTA input	R/W
17	CSELCB	ELC_GPTB Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTB input 1: Counter clear enabled at the ELC_GPTB input	R/W
18	CSELCC	ELC_GPTC Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTC input 1: Counter clear enabled at the ELC_GPTC input	R/W
19	CSELCD	ELC_GPTD Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTD input 1: Counter clear enabled at the ELC_GPTD input	R/W

Bit	Symbol	Function	R/W
20	CSELCE	ELC_GPTE Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTE input 1: Counter clear enabled at the ELC_GPTE input	R/W
21	CSELCF	ELC_GPTF Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTF input 1: Counter clear enabled at the ELC_GPTF input	R/W
22	CSELCG	ELC_GPTG Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTG input 1: Counter clear enabled at the ELC_GPTG input	R/W
23	CSELCH	ELC_GPTH Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTH input 1: Counter clear enabled at the ELC_GPTH input	R/W
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	CCLR	Software Source Counter Clear Enable 0: Counter clear disabled by the GTCLR register 1: Counter clear enabled by the GTCLR register	R/W

Note: n = 0 to 13

The GTCSR sets the source to clear the GTCNT counter.

Counter clearing can be executed whether the counter is running (GTCR.CST = 1) or stopped (GTCR.CST = 0).

Inputs from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

CSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Clear Enable)

The CSGTRGAR bit enables or disables the GTCNT counter clear on the rising edge of the GTETRGA pin input.

CSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Clear Enable)

The CSGTRGAF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGA pin input.

CSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Clear Enable)

The CSGTRGBR bit enables or disables the GTCNT counter clear on the rising edge of the GTETRGB pin input.

CSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Clear Enable)

The CSGTRGBF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGB pin input.

CSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Clear Enable)

The CSGTRGCR bit enables or disables the GTCNT counter clear on the rising edge of the GTETRGC pin input.

CSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Clear Enable)

The CSGTRGCF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGC pin input.

CSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Clear Enable)

The CSGTRGDR bit enables or disables the GTCNT counter clear on the rising edge of the GTETRGD pin input.

CSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Clear Enable)

The CSGTRGDF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGD pin input.

CSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Clear Enable)

The CSCARBL bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

CSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Clear Enable)

The CSCARBH bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

CSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Clear Enable)

The CSCAFBL bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

CSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Clear Enable)

The CSCAFBH bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

CSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Clear Enable)

The CSCBRAL bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

CSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Clear Enable)

The CSCBRAH bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

CSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Clear Enable)

The CSCBFAL bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

CSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Clear Enable)

The CSCBFAH bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

CSELCm bit (ELCm Event Source Counter Clear Enable) (m = A to H)

The CSELCm bit enables or disables the GTCNT counter clear at the ELC_GPTm event input.

CCLR bit (Software Source Counter Clear Enable)

The CCLR bit enables or disables the GTCNT counter clear by the GTCLR register.

21.2.8 GTUPSR : General PWM Timer Up Count Source Select Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x1C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	USELCH	USELCG	USELCF	USELCE	USELCD	USELCC	USELCB	USELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	USCBFAH	USCBFAL	USCBRAH	USCBRAL	USCAFBH	USCAFBL	USCARBH	USCARBL	USGTRGDF	USGTRGDR	USGTRGCF	USGTRGCR	USGTRGBF	USGTRGBR	USGTRGAF	USGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	USGTRGAR	GTETRGA Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGA input 1: Counter count up enabled on the rising edge of GTETRGA input	R/W
1	USGTRGAF	GTETRGA Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGA input 1: Counter count up enabled on the falling edge of GTETRGA input	R/W

Bit	Symbol	Function	R/W
2	USGTRGBR	GTETRGB Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGB input 1: Counter count up enabled on the rising edge of GTETRGB input	R/W
3	USGTRGBF	GTETRGB Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGB input 1: Counter count up enabled on the falling edge of GTETRGB input	R/W
4	USGTRGCR	GTETRGC Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGC input 1: Counter count up enabled on the rising edge of GTETRGC input	R/W
5	USGTRGCF	GTETRGC Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGC input 1: Counter count up enabled on the falling edge of GTETRGC input	R/W
6	USGTRGDR	GTETRGD Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGD input 1: Counter count up enabled on the rising edge of GTETRGD input	R/W
7	USGTRGDF	GTETRGD Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGD input 1: Counter count up enabled on the falling edge of GTETRGD input	R/W
8	USCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count up enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	USCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count up enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	USCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count up enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	USCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count up enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	USCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count up enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	USCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count up enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	USCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count up enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	USCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count up enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W

Bit	Symbol	Function	R/W
16	USELCA	ELC_GPTA Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTA input 1: Counter count up enabled at the ELC_GPTA input	R/W
17	USELCB	ELC_GPTB Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTB input 1: Counter count up enabled at the ELC_GPTB input	R/W
18	USELCC	ELC_GPTC Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTC input 1: Counter count up enabled at the ELC_GPTC input	R/W
19	USELCD	ELC_GPTD Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTD input 1: Counter count up enabled at the ELC_GPTD input	R/W
20	USELCE	ELC_GPTE Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTE input 1: Counter count up enabled at the ELC_GPTE input	R/W
21	USELCF	ELC_GPTF Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTF input 1: Counter count up enabled at the ELC_GPTF input	R/W
22	USELCG	ELC_GPTG Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTG input 1: Counter count up enabled at the ELC_GPTG input	R/W
23	USELCH	ELC_GPTH Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTH input 1: Counter count up enabled at the ELC_GPTH input	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: n = 0 to 13

The GTUPSR sets the source to count up the GTCNT counter.

When at least one bit in the GTUPSR register is set to 1, the GTCNT counter is counted up by the source that is set to 1 in this register. In this case, GTCR.TPCS has no effect.

Number of increment in counting is one even when multiple sources are generated simultaneously.

Inputs from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

USGTRGAR bit (GTETRGA Pin Rising Input Source Counter Count Up Enable)

The USGTRGAR bit enables or disables the GTCNT counter count up on the rising edge of the GTETRGA pin input.

USGTRGAF bit (GTETRGA Pin Falling Input Source Counter Count Up Enable)

The USGTRGAF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGA pin input.

USGTRGBR bit (GTETRGB Pin Rising Input Source Counter Count Up Enable)

The USGTRGBR bit enables or disables the GTCNT counter count up on the rising edge of the GTETRGB pin input.

USGTRGBF bit (GTETRGB Pin Falling Input Source Counter Count Up Enable)

The USGTRGBF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGB pin input.

USGTRGCR bit (GTETRGC Pin Rising Input Source Counter Count Up Enable)

The USGTRGCR bit enables or disables the GTCNT counter count up on the rising edge of the GTETRGC pin input.

USGTRGCF bit (GTETRGC Pin Falling Input Source Counter Count Up Enable)

The USGTRGCF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGC pin input.

USGTRGDR bit (GTETRGD Pin Rising Input Source Counter Count Up Enable)

The USGTRGDR bit enables or disables the GTCNT counter count up on the rising edge of the GTETRGD pin input.

USGTRGDF bit (GTETRGD Pin Falling Input Source Counter Count Up Enable)

The USGTRGDF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGD pin input.

USCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Up Enable)

The USCARBL bit enables or disables GTCNT counter count up on the rising edge of GTIOCnA pin input, when GTIOCnB input is 0.

USCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Up Enable)

The USCARBH bit enables or disables the GTCNT counter count up on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

USCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Up Enable)

The USCAFBL bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

USCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Up Enable)

The USCAFBH bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

USCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Up Enable)

The USCBRAL bit enables or disables the GTCNT counter count up on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

USCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Up Enable)

The USCBRAH bit enables or disables the GTCNT counter count up on the rising edge of the GTIOCnB pin input, when the GTIOCnA input is 1.

USCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Up Enable)

The USCBFAL bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnB pin input, when the GTIOCnA input is 0.

USCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Up Enable)

The USCBFAH bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnB pin input, when the GTIOCnA input is 1.

USELCm bit (ELC_GPTm Event Source Counter Count Up Enable) (m = A to H)

The USELCm bit enables or disables the GTCNT counter count up at the ELC_GPTm event input.

21.2.9 GTDNSR : General PWM Timer Down Count Source Select Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DSELCH	DSELCG	DSELCF	DSELCE	DSELCD	DSELCC	DSELCB	DSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DSCBFAH	DSCBFAL	DSCBRAH	DSCBRAL	DSCAFBH	DSCAFBL	DSCARBH	DSCARBL	DSGTRGDF	DSGTRGDR	DSGTRGCF	DSGTRGCR	DSGTRGBF	DSGTRGBR	DSGTRGAF	DSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DSGTRGAR	GTETRGA Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGA input 1: Counter count down enabled on the rising edge of GTETRGA input	R/W
1	DSGTRGAF	GTETRGA Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGA input 1: Counter count down enabled on the falling edge of GTETRGA input	R/W
2	DSGTRGBR	GTETRGB Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGB input 1: Counter count down enabled on the rising edge of GTETRGB input	R/W
3	DSGTRGBF	GTETRGB Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGB input 1: Counter count down enabled on the falling edge of GTETRGB input	R/W
4	DSGTRGCR	GTETRGC Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGC input 1: Counter count down enabled on the rising edge of GTETRGC input	R/W
5	DSGTRGCF	GTETRGC Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGC input 1: Counter count down enabled on the falling edge of GTETRGC input	R/W
6	DSGTRGDR	GTETRGD Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGD input 1: Counter count down enabled on the rising edge of GTETRGD input	R/W
7	DSGTRGDF	GTETRGD Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGD input 1: Counter count down enabled on the falling edge of GTETRGD input	R/W
8	DSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count down enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	DSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count down enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	DSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count down enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	DSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count down enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	DSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count down enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W

Bit	Symbol	Function	R/W
13	DSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count down enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	DSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count down enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	DSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count down enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	DSELCA	ELC_GPTA Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTA input 1: Counter count down enabled at the ELC_GPTA input	R/W
17	DSELCB	ELC_GPTB Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTB input 1: Counter count down enabled at the ELC_GPTB input	R/W
18	DSELCC	ELC_GPTC Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTC input 1: Counter count down enabled at the ELC_GPTC input	R/W
19	DSELCD	ELC_GPTD Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTD input 1: Counter count down enabled at the ELC_GPTD input	R/W
20	DSELCE	ELC_GPTE Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTE input 1: Counter count down enabled at the ELC_GPTE input	R/W
21	DSELCF	ELC_GPTF Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTF input 1: Counter count down enabled at the ELC_GPTF input	R/W
22	DSELCG	ELC_GPTG Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTG input 1: Counter count down enabled at the ELC_GPTG input	R/W
23	DSELCH	ELC_GPTF Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTF input 1: Counter count down enabled at the ELC_GPTF input	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: n = 0 to 13

The GTDNSR sets the source to count down the GTCNT counter.

When at least one bit in the GTDNSR register is set to 1, the GTCNT counter is counted down by the source that is set to 1 in this register. In this case, GTCR.TPCS has no effect.

Number of decrement in counting is one even when multiple sources are generated simultaneously.

Inputs from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

DSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Count Down Enable)

The DSGTRGAR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGA pin input.

DSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Count Down Enable)

The DSGTRGAF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGA pin input.

DSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Count Down Enable)

The DSGTRGBR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGB pin input.

DSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Count Down Enable)

The DSGTRGBF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGB pin input.

DSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Count Down Enable)

The DSGTRGCR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGC pin input.

DSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Count Down Enable)

The DSGTRGCF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGC pin input.

DSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Count Down Enable)

The DSGTRGDR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGD pin input.

DSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Count Down Enable)

The DSGTRGDF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGD pin input.

DSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Down Enable)

The DSCARBL bit enables or disables the GTCNT counter count down on the rising edge of the GTIOCnA pin input, when the GTIOCnB input is 0.

DSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Down Enable)

The DSCARBH bit enables or disables the GTCNT counter count down on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

DSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Down Enable)

The DSCAFBL bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

DSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Down Enable)

The DSCAFBH bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

DSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Down Enable)

The DSCBRAL bit enables or disables the GTCNT counter count down on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

DSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Down Enable)

The DSCBRAH bit enables or disables the GTCNT counter count down on the rising edge of GTIOCnB pin input, when GTIOCnA input is 1.

DSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Down Enable)

The DSCBFAL bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

DSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Down Enable)

The DSCBFAH bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

DSELCm bit (ELC_GPTm Event Source Counter Count Down Enable) (m = A to H)

The DSELCm bit enables or disables the GTCNT counter count down at the ELC_GPTm event input.

21.2.10 GTICASR : General PWM Timer Input Capture Source Select Register A

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	ASELCH	ASELCG	ASELCF	ASELCE	ASELCD	ASELCC	ASELCB	ASELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ASCBFAH	ASCBFAL	ASCBRAH	ASCBRAL	ASCAFBH	ASCAFBL	ASCARBH	ASCARBL	ASGTRGDF	ASGTRGDR	ASGTRGCF	ASGTRGCR	ASGTRGBF	ASGTRGBR	ASGTRGAF	ASGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ASGTRGAR	GTETRGA Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGA input 1: GTCCRA input capture enabled on the rising edge of GTETRGA input	R/W
1	ASGTRGAF	GTETRGA Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGA input 1: GTCCRA input capture enabled on the falling edge of GTETRGA input	R/W
2	ASGTRGBR	GTETRGB Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGB input 1: GTCCRA input capture enabled on the rising edge of GTETRGB input	R/W
3	ASGTRGBF	GTETRGB Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGB input 1: GTCCRA input capture enabled on the falling edge of GTETRGB input	R/W
4	ASGTRGCR	GTETRGC Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGC input 1: GTCCRA input capture enabled on the rising edge of GTETRGC input	R/W
5	ASGTRGCF	GTETRGC Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGC input 1: GTCCRA input capture enabled on the falling edge of GTETRGC input	R/W
6	ASGTRGDR	GTETRGD Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGD input 1: GTCCRA input capture enabled on the rising edge of GTETRGD input	R/W
7	ASGTRGDF	GTETRGD Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGD input 1: GTCCRA input capture enabled on the falling edge of GTETRGD input	R/W
8	ASCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRA input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W

Bit	Symbol	Function	R/W
9	ASCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRA input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	ASCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRA input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	ASCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRA input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	ASCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRA input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	ASCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRA input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	ASCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRA input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	ASCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRA input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	ASELCA	ELC_GPTA Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTA input 1: GTCCRA input capture enabled at the ELC_GPTA input	R/W
17	ASELCB	ELC_GPTB Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTB input 1: GTCCRA input capture enabled at the ELC_GPTB input	R/W
18	ASELCC	ELC_GPTC Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTC input 1: GTCCRA input capture enabled at the ELC_GPTC input	R/W
19	ASELCD	ELC_GPTD Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTD input 1: GTCCRA input capture enabled at the ELC_GPTD input	R/W
20	ASELCE	ELC_GPTE Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTE input 1: GTCCRA input capture enabled at the ELC_GPTE input	R/W
21	ASELCF	ELC_GPTF Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTF input 1: GTCCRA input capture enabled at the ELC_GPTF input	R/W

Bit	Symbol	Function	R/W
22	ASELCG	ELC_GPTG Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTG input 1: GTCCRA input capture enabled at the ELC_GPTG input	R/W
23	ASELCH	ELC_GPTH Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTH input 1: GTCCRA input capture enabled at the ELC_GPTH input	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: n = 0 to 13

The GTICASR sets the source of input capture for GTCCRA.

When at least one bit among bits in the GTICASR register is set to 1, input capture operation making the GTCCRA register as an input capture register is performed.

Inputs from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

ASGTRGAR bit (GTETRGA Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGAR bit enables or disables the input capture for GTCCRA on the rising edge of the GTETRGA pin input.

ASGTRGAF bit (GTETRGA Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGAF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGA pin input.

ASGTRGBR bit (GTETRGB Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGBR bit enables or disables the input capture for GTCCRA on the rising edge of the GTETRGB pin input.

ASGTRGBF bit (GTETRGB Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGBF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGB pin input.

ASGTRGCR bit (GTETRGC Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGCR bit enables or disables the input capture for GTCCRA on the rising edge of the GTETRGC pin input.

ASGTRGCF bit (GTETRGC Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGCF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGC pin input.

ASGTRGDR bit (GTETRGD Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGDR bit enables or disables the input capture for GTCCRA on the rising edge of the GTETRGD pin input.

ASGTRGDF bit (GTETRGD Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGDF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGD pin input.

ASCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable)

The ASCARBL bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

ASCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRA Input Capture Enable)

The ASCARBH bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

ASCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable)

The ASCAFBL bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

ASCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRA Input Capture Enable)

The ASCAFBH bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnA pin input, when the GTIOCnB input is 1.

ASCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable)

The ASCBRAL bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnB pin input, when the GTIOCnA input is 0.

ASCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRA Input Capture Enable)

The ASCBRAH bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

ASCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable)

The ASCBFAL bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

ASCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRA Input Capture Enable)

The ASCBFAH bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

ASELCm bit (ELC_GPTm Event Source Counter GTCCRA Input Capture Enable) (m = A to H)

The ASELCm bit enables or disables the input capture for GTCCRA at the ELC_GPTm event input.

21.2.11 GTICBSR : General PWM Timer Input Capture Source Select Register B

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	BSELCH	BSELCG	BSELCF	BSELCE	BSELCD	BSELCC	BSELCB	BSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSCBFAH	BSCBFAL	BSCBRAH	BSCBRAL	BSCAFBH	BSCAFBL	BSCARBH	BSCARBL	BSGTRGDF	BSGTRGDR	BSGTRGCF	BSGTRGCR	BSGTRGBF	BSGTRGBR	BSGTRGAF	BSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BSGTRGAR	GTETRGA Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGA input 1: GTCCRB input capture enabled on the rising edge of GTETRGA input	R/W
1	BSGTRGAF	GTETRGA Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGA input 1: GTCCRB input capture enabled on the falling edge of GTETRGA input	R/W
2	BSGTRGBR	GTETRGB Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGB input 1: GTCCRB input capture enabled on the rising edge of GTETRGB input	R/W

Bit	Symbol	Function	R/W
3	BSGTRGBF	GTETRGB Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGB input 1: GTCCRB input capture enabled on the falling edge of GTETRGB input	R/W
4	BSGTRGCR	GTETRGC Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGC input 1: GTCCRB input capture enabled on the rising edge of GTETRGC input	R/W
5	BSGTRGCF	GTETRGC Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGC input 1: GTCCRB input capture enabled on the falling edge of GTETRGC input	R/W
6	BSGTRGDR	GTETRGD Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGD input 1: GTCCRB input capture enabled on the rising edge of GTETRGD input	R/W
7	BSGTRGDF	GTETRGD Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGD input 1: GTCCRB input capture enabled on the falling edge of GTETRGD input	R/W
8	BSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRB input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	BSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRB input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	BSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRB input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	BSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRB input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	BSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRB input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	BSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRB input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	BSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRB input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W

Bit	Symbol	Function	R/W
15	BSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRB input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	BSELCA	ELC_GPTA Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTA input 1: GTCCRB input capture enabled at the ELC_GPTA input	R/W
17	BSELCB	ELC_GPTB Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTB input 1: GTCCRB input capture enabled at the ELC_GPTB input	R/W
18	BSELCC	ELC_GPTC Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTC input 1: GTCCRB input capture enabled at the ELC_GPTC input	R/W
19	BSELCD	ELC_GPTD Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTD input 1: GTCCRB input capture enabled at the ELC_GPTD input	R/W
20	BSELCE	ELC_GPTE Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTE input 1: GTCCRB input capture enabled at the ELC_GPTE input	R/W
21	BSELCF	ELC_GPTF Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTF input 1: GTCCRB input capture enabled at the ELC_GPTF input	R/W
22	BSELCG	ELC_GPTG Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTG input 1: GTCCRB input capture enabled at the ELC_GPTG input	R/W
23	BSELCH	ELC_GPTH Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTH input 1: GTCCRB input capture enabled at the ELC_GPTH input	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: n = 0 to 13

The GTICBSR sets the source of input capture for GTCCRB.

When at least one bit among bits in the GTICBSR register is set to 1, input capture operation making the GTCCRB register as an input capture register is performed.

Inputs from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

BSGTRGAR bit (GTETRGA Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGAR bit enables or disables the input capture for GTCCRB on the rising edge of the GTETRGA pin input.

BSGTRGAF bit (GTETRGA Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGAF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGA pin input.

BSGTRGBR bit (GTETRGB Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGBR bit enables or disables the input capture for GTCCRB on the rising edge of GTETRGB pin input.

BSGTRGBF bit (GTETRGB Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGBF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGB pin input.

BSGTRGCR bit (GTETRGC Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGCR bit enables or disables the input capture for GTCCRB on the rising edge of GTETRGC pin input.

BSGTRGCF bit (GTETRGC Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGCF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGC pin input.

BSGTRGDR bit (GTETRGD Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGDR bit enables or disables the input capture for GTCCRB on the rising edge of GTETRGD pin input.

BSGTRGDF bit (GTETRGD Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGDF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGD pin input.

BSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable)

The BSCARBL bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnA pin input, when the GTIOCnB input is 0.

BSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRB Input Capture Enable)

The BSCARBH bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

BSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable)

The BSCAFBL bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

BSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRB Input Capture Enable)

The BSCAFBH bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

BSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable)

The BSCBRAL bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

BSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRB Input Capture Enable)

The BSCBRAH bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

BSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable)

The BSCBFAL bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

BSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRB Input Capture Enable)

The BSCBFAH bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

BSELCm bit (ELC_GPTm Event Source Counter GTCCRB Input Capture Enable) (m = A to H)

The BSELCm bit enables or disables the input capture for GTCCRB at the ELC_GPTm event input.

21.2.12 GTCR : General PWM Timer Control Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	TPCS[3:0]				—	—	—	—	MD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CST	Count Start 0: Count operation is stopped 1: Count operation is performed	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
18:16	MD[2:0]	Mode Select 0 0 0: Saw-wave PWM mode (single buffer or double buffer possible) 0 0 1: Saw-wave one-shot pulse mode (fixed buffer operation) 0 1 0: Setting prohibited 0 1 1: Setting prohibited 1 0 0: Triangle-wave PWM mode 1 (32-bit transfer at trough) (single buffer or double buffer is possible) 1 0 1: Triangle-wave PWM mode 2 (32-bit transfer at crest and trough) (single buffer or double buffer is possible) 1 1 0: Triangle-wave PWM mode 3 (64-bit transfer at trough) (fixed buffer operation) 1 1 1: Setting prohibited	R/W
22:19	—	These bits are read as 0. The write value should be 0.	R/W
26:23	TPCS[3:0]	Timer Prescaler Select 0 0 0 0: PCLKD/1 0 0 0 1: PCLKD/2 0 0 1 0: PCLKD/4 0 0 1 1: PCLKD/8 0 1 0 0: PCLKD/16 0 1 0 1: PCLKD/32 0 1 1 0: PCLKD/64 0 1 1 1: Setting prohibited 1 0 0 0: PCLKD/256 1 0 0 1: Setting prohibited 1 0 1 0: PCLKD/1024 1 0 1 1: Setting prohibited 1 1 0 0: GTETRGA (Via the POEG) 1 1 0 1: GTETRGB (Via the POEG) 1 1 1 0: GTETRGC (Via the POEG) 1 1 1 1: GTETRGD (Via the POEG)	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

The GTCR controls GTCNT.

CST bit (Count Start)

The CST bit controls the GTCNT counter start and stop.

[Setting conditions]

- The GTSTR value where the channel number associated with the bit number is set to 1 with the GTSSR.CSTRT bit at 1
- The ELC event input, the external trigger, or the GTIOCnA/GTIOCnB input that are enabled by GTSSR for the starting counter source, occurs (n = 0 to 13)
- 1 is written by software directly.

[Clearing conditions]

- The GTSTP value where the channel number associated with the bit number is set to 1 with the GTPSR.CSTOP bit at 1

- The ELC event input, the external trigger, or the GTIOCnA/GTIOCnB input enabled by GTPSR as the counter stop source, occurs (n = 0 to 13)
- 0 is written by software directly.
- When the period count function is finished while the GTPC.ASTP bit is 1.

MD[2:0] bits (Mode Select)

The MD[2:0] bits select the GPT operating mode.

Only the MD[2] bit is valid at input capture. Counting in saw-wave mode is performed with 0 for the MD[2] bit, and counting in triangle-wave mode is performed with 1 for the MD[2] bit. The MD[2:0] bits must be set while the GTCNT operation is stopped. During the event count operation (when at least one bit among the bits for the GTUPSR and GTDNSR registers is set to 1), set the MD[2:0] bits to the initial value (000b).

TPCS[3:0] bits (Timer Prescaler Select)

The TPCS[3:0] bits select the clock for GTCNT. A clock prescaler can be selected independently for each channel. The TPCS[3:0] bits must be set while the GTCNT operation is stopped. When the GTETRGA, GTETRGB, GTETRGC, or GTETRGD is selected, output for the POEG at rising becomes a clock source. Set the polarity of these signals with the POEG.

21.2.13 GTUDDTYC : General PWM Timer Count Direction and Duty Setting Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	OBDTYR	OBDTYF	OBDTY[1:0]		—	—	—	—	OADTYR	OADTYF	OADTY[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UDF	UD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	UD	Count Direction Setting 0: GTCNT counts down 1: GTCNT counts up	R/W
1	UDF	Forcible Count Direction Setting 0: Not forcibly set 1: Forcibly set	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
17:16	OADTY[1:0]	GTIOCnA Output Duty Setting 0 0: GTIOCnA pin duty depends on the compare match 0 1: GTIOCnA pin duty depends on the compare match 1 0: GTIOCnA pin duty 0% 1 1: GTIOCnA pin duty 100%	R/W
18	OADTYF	Forcible GTIOCnA Output Duty Setting 0: Not forcibly set 1: Forcibly set	R/W
19	OADTYR	GTIOCnA Output Value Selecting after Releasing 0%/100% Duty Setting 0: The function selected by the GTIOA[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting. 1: The function selected by the GTIOA[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting.	R/W

Bit	Symbol	Function	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
25:24	OBDTY[1:0]	GTIOCnB Output Duty Setting 0 0: GTIOCnB pin duty depends on the compare match 0 1: GTIOCnB pin duty depends on the compare match 1 0: GTIOCnB pin duty 0% 1 1: GTIOCnB pin duty 100%	R/W
26	OBDTYF	Forcible GTIOCnB Output Duty Setting 0: Not forcibly set 1: Forcibly set	R/W
27	OBDTYR	GTIOCnB Output Value Selecting after Releasing 0%/100% Duty Setting 0: The function selected by the GTIOB[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting. 1: The function selected by the GTIOB[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: n = 0 to 13

The GTUDDTYC sets the direction in which the GTCNT counts (up-counting or down-counting), and sets the duty of the GTIOCnA/GTIOCnB pin output.

The setting is invalid during the event count operation.

Count Direction:

- In saw-wave mode.
  When the UD value is set to 0 during up-counting, the count direction changes at an overflow (the timing synchronous with count clock after the GTCNT value becomes the GTPR value). When the UD value is set to 1 during down-counting, the count direction changes at an underflow (the timing synchronous with count clock after the GTCNT value becomes 0).
  When the UD value changes from 1 to 0 with the UDF bit being 0 and while counting stops, the counter starts up-counting and the count direction changes at an overflow (the timing synchronous with count clock after the GTCNT value becomes the GTPR value). When the UD value changes from 0 to 1 with the UDF bit being 0 and while counting stops, the counter starts down-counting and the count direction changes at an underflow (the timing synchronous with count clock after the GTCNT value becomes 0).
  When the UDF bit is set to 1 while counting stops, the UD bit value is reflected in the count direction when counting starts.
- In triangle-wave mode.
  When the UD value changes during counting, the count direction does not change. When the UD value changes while the UDF bit is 0 and counting stops, the change is not reflected in the count direction when counting starts.
  When the UDF bit is set to 1 while counting is stopped, the UD value is reflected in the count direction when counting starts.

UD bit (Count Direction Setting)

The UD bit sets the count direction (up-counting or down-counting) for GTCNT.

UDF bit (Forcible Count Direction Setting)

The UDF bit forcibly sets the count direction when GTCNT starts operation as the UD value. Only 0 should be written to this bit during counter operation. When 1 is written to this bit while counting stops, return this bit to 0 before counting starts.

Output duty

- In saw-wave mode.
  When the OADTY/OBDTY value changes during up-counting, the duty is reflected at an overflow (GTCNT = GTPR).
  When the OADTY/OBDTY value is changed during down-counting, the duty is reflected at an underflow (GTCNT = 0).
  When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 0 and while counting stops the output duty is not reflected at the starting counter operation. When the count direction is up, the output duty is reflected

at an overflow (GTCNT = GTPR). When the count direction is down, the output duty is reflected at an underflow (GTCNT = 0).
When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 1 and while counting stops, the output duty is reflected at starting counter operation.

- In triangle-wave mode.
  When the OADTY/OBDTY value changes during counting, the duty is reflected at an underflow.
  When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 0 and while counting stops, the output duty is not reflected at the starting counter operation. The output duty is reflected at an underflow.
  When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 1 and while counting stops, the output duty is reflected at starting counter operation.

In both saw-wave mode and triangle-wave mode, when the OADTYF/OBDTYF bit is set back to 0 and the OADTY[1:0]/OBDTY[1:0] bits are set after setting the OADTYF/OBDTYF bit to 1 and setting the OADTY[1:0]/OBDTY[1:0] bits for the duty of first cycle while count operation is stopped, these duty-cycle set during stopping count operation are reflected in the first cycle and the second cycle after starting count operation.

OmDTY[1:0] bits (GTIOCnm Output Duty Setting) (m = A, B)

The OmDTY[1:0] bits set the output duty (0%, 100% or compare match control) of the GTIOCnm pin.

OmDTYF bit (Forcible GTIOCnm Output Duty Setting) (m = A, B)

The OmDTYF bit forcibly sets the output duty cycle to the OmDTY setting. Set this bit to 0 during counter operation.

OmDTYR bit (GTIOCnm Output Value Selecting after Releasing 0%/100% Duty Setting) (m = A, B)

The OmDTYR bit selects the value that is the object of output retained or toggled at cycle end, when the control changes from 0% or 100% duty setting to compare match for the GTIOCnm pin and GTIOR.GTIOm[3:2] bits are set to 00b (output retained at cycle end) or the GTIOR.GTIOm[3:2] bits are set to 11b (output toggled at cycle end).

The GPT internally continues to perform compare match operation during duty-cycle 0% or 100% operation. When the OmDTYR bit is 1, the value after the period has elapsed due this compare match operation is target for the GTIOm[3:2] bits.

21.2.14 GTIOR : General PWM Timer I/O Control Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x34

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFCSB[1:0]		NFBEN	—	—	OBDF[1:0]		OBE	OBHLD	OBDFLT	—	GTIOB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NFCSA[1:0]		NFAEN	—	—	OADF[1:0]		OAE	OAHLD	OADFLT	—	GTIOA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	GTIOA[4:0]	GTIOCnA Pin Function Select See Table 21.4.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	OADFLT	GTIOCnA Pin Output Value Setting at the Count Stop 0: The GTIOCnA pin outputs low when counting stops 1: The GTIOCnA pin outputs high when counting stops	R/W

Bit	Symbol	Function	R/W
7	OAHLD	GTIOCnA Pin Output Setting at the Start/Stop Count 0: The GTIOCnA pin output level at the start or stop of counting depends on the register setting 1: The GTIOCnA pin output level is retained at the start or stop of counting	R/W
8	OAE	GTIOCnA Pin Output Enable 0: Output is disabled 1: Output is enabled	R/W
10:9	OADF[1:0]	GTIOCnA Pin Disable Value Setting 0 0: None of the below options are specified 0 1: GTIOCnA pin is set to Hi-Z in response to controlling the output negation 1 0: GTIOCnA pin is set to 0 in response to controlling the output negation 1 1: GTIOCnA pin is set to 1 in response to controlling the output negation	R/W
12:11	—	These bits are read as 0. The write value should be 0.	R/W
13	NFAEN	Noise Filter A Enable 0: The noise filter for the GTIOCnA pin is disabled 1: The noise filter for the GTIOCnA pin is enabled	R/W
15:14	NFCSA[1:0]	Noise Filter A Sampling Clock Select 0 0: PCLKD/1 0 1: PCLKD/4 1 0: PCLKD/16 1 1: PCLKD/64	R/W
20:16	GTIOB[4:0]	GTIOCnB Pin Function Select See Table 21.4.	R/W
21	—	This bit is read as 0. The write value should be 0.	R/W
22	OBDFLT	GTIOCnB Pin Output Value Setting at the Count Stop 0: The GTIOCnB pin outputs low when counting stops 1: The GTIOCnB pin outputs high when counting stops	R/W
23	OBHLD	GTIOCnB Pin Output Setting at the Start/Stop Count 0: The GTIOCnB pin output level at the start/stop of counting depends on the register setting 1: The GTIOCnB pin output level is retained at the start/stop of counting	R/W
24	OBE	GTIOCnB Pin Output Enable 0: Output is disabled 1: Output is enabled	R/W
26:25	OBDF[1:0]	GTIOCnB Pin Disable Value Setting 0 0: None of the below options are specified 0 1: GTIOCnB pin is set to Hi-Z in response to controlling the output negation 1 0: GTIOCnB pin is set to 0 in response to controlling the output negation 1 1: GTIOCnB pin is set to 1 in response to controlling the output negation	R/W
28:27	—	These bits are read as 0. The write value should be 0.	R/W
29	NFBEN	Noise Filter B Enable 0: The noise filter for the GTIOCnB pin is disabled 1: The noise filter for the GTIOCnB pin is enabled	R/W
31:30	NFCSB[1:0]	Noise Filter B Sampling Clock Select 0 0: PCLKD/1 0 1: PCLKD/4 1 0: PCLKD/16 1 1: PCLKD/64	R/W

Note: n = 0 to 13

The GTIOR sets the functions of the GTIOCnA and GTIOCnB pins. (n = 0 to 13)

GTIOA[4:0] bits (GTIOCnA Pin Function Select)

The GTIOA[4:0] bits select the GTIOCnA pin function. For details, see Table 21.4.

OADFLT bit (GTIOCnA Pin Output Value Setting at the Count Stop)

The OADFLT bit sets whether the GTIOCnA pin outputs high or low when counting stops.

OAHLD bit (GTIOCnA Pin Output Setting at the Start/Stop Count)

The OAHLD bit specifies whether the GTIOCnA pin output level is retained or the level at the start or stop of counting depends on the register setting.

When the OAHLD bit is set to 0:

- The value specified in bit [4] of the GTIOA[4:0] bits is output when counting starts
- The value specified in the OADFLT bit is output when counting stops
- If the OADFLT bit is modified while counting stops, the new value is immediately reflected in the output.

When the OAHLD bit is set to 1:

- The output is retained when counting starts or stops.

OAE bit (GTIOCnA Pin Output Enable)

The OAE bit disables or enables the GTIOCnA pin output.

When GTCCRA register is used as the input capture register (at least one bit in the GTICASR register is set to 1), the GTIOCnA pin does not output regardless of the OAE bit value.

OADF[1:0] bits (GTIOCnA Pin Disable Value Setting)

The OADF[1:0] bits select the output value of the GTIOCnA pin in response to a request to disable output from the POEG.

NFAEN bit (Noise Filter A Enable)

The NFAEN bit disables or enables the noise filter for input from the GTIOCnA pin. Because changing the value of the bit might lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the GTIOR register before doing so.

NFCSA[1:0] bits (Noise Filter A Sampling Clock Select)

The NFCSA[1:0] bits set the sampling interval for the noise filter of the GTIOCnA pin. When setting these bits, wait for 2 cycles of the selected sampling interval before setting the input capture function.

GTIOB[4:0] bits (GTIOCnB Pin Function Select)

The GTIOB[4:0] bits select the GTIOCnB pin function. For details, see Table 21.4.

OBDFLT bit (GTIOCnB Pin Output Value Setting at the Count Stop)

The OBDFLT bit sets whether the GTIOCnB pin outputs high or low when counting stops.

OBHLD bit (GTIOCnB Pin Output Setting at the Start/Stop Count)

The OBHLD bit specifies whether theGTIOCnB pin output level is retained or the level at the start or stop of counting depends on the register setting.

When the OBHLD bit is set to 0:

- The value specified in bit [4] of the GTIOB[4:0] bits is output when counting starts
- The value specified in the OBDFLT bit is output when counting stops
- If the OBDFLT bit is modified while counting stops, the new value is immediately reflected in the output.

When the OBHLD bit is set to 1:

- The output is retained when counting starts or stops.

OBE bit (GTIOCnB Pin Output Enable)

The OBE bit disables or enables the GTIOCnB pin output.

When GTCCRB register is used as the input capture register (at least one bit in the GTICBSR register is set to 1), the GTIOCnB pin does not output regardless of the OBE bit value.

OBDF[1:0] bits (GTIOCnB Pin Disable Value Setting)

The OBDF[1:0] bits select the output value of the GTIOCnB pin in response to a request to disable output from the POEG.

NFBEN bit (Noise Filter B Enable)

The NFBEN bit disables or enables the noise filter for input from the GTIOCnB pin. Because changing the value of the bit might lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the GTIOR register before doing so.

NFCSB[1:0] bits (Noise Filter B Sampling Clock Select)

The NFCSB[1:0] bits set the sampling interval for the noise filter of the GTIOCnB pin. When setting these bits, wait for 2 cycles of the selected sampling interval before setting the input capture function.

Table 21.4 Settings of GTIOA[4:0] and GTIOB[4:0] bits

GTIOA/GTIOB[4:0] bits					Function		
b4	b3	b2	b1	b0	b4	b3, b2*1 *2 *3	b1, b0*2
0	0	0	0	0	Initial output is low	Output retained at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	0	1	0			High output at GTCCRA/GTCCRB compare match
0	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	0	1	0	0		Low output at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	1	1	0			High output at GTCCRA/GTCCRB compare match
0	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	0	0	0		High output at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	0	1	0			High output at GTCCRA/GTCCRB compare match
0	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	1	0	0		Output toggled at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	1	1	0			High output at GTCCRA/GTCCRB compare match
0	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	0	0	0	0	Initial output is high	Output retained at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	0	1	0			High output at GTCCRA/GTCCRB compare match
1	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	0	1	0	0		Low output at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	1	1	0			High output at GTCCRA/GTCCRB compare match
1	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	0	0	0		High output at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	0	1	0			High output at GTCCRA/GTCCRB compare match
1	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	1	0	0		Output toggled at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	1	1	0			High output at GTCCRA/GTCCRB compare match
1	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match

Note 1. The cycle end means an overflow (GTCNT changes from GTPR to 0 in up-counting), an underflow (GTCNT changes from 0 to GTPR in down-counting), or counter clearing for saw-wave mode, and means a trough (GTCNT changes from 0 to 1) for triangle-wave mode.

Note 2. When the timing of a cycle end and the timing of a GTCCRA/GTCCRB compare match are the same in a compare-match operation, the b3 and b2 settings are given priority in saw-wave PWM mode, and the b1 and b0 settings are given priority in any other mode.
Note 3. In event count operation where at least one bit in GTUPSR or GTDNSR is set to 1, the setting of b3 and b2 is ignored.

21.2.15 GTINTAD : General PWM Timer Interrupt Output Setting Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	GRPABL	GRPABH	—	—	—	GRP[1:0]		—	—	—	—	ADTRBDEN	ADTRBUEN	ADTRADEN	ADTRAUEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	ADTRAUEN	GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
17	ADTRADEN	GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
18	ADTRBUEN	GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
19	ADTRBDEN	GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
25:24	GRP[1:0]	Output Disable Source Select 0 0: Group A output disable source is selected 0 1: Group B output disable source is selected 1 0: Group C output disable source is selected 1 1: Group D output disable source is selected	R/W
28:26	—	These bits are read as 0. The write value should be 0.	R/W
29	GRPABH	Same Time Output Level High Disable Request Enable 0: Same time output level high disable request disabled 1: Same time output level high disable request enabled	R/W
30	GRPABL	Same Time Output Level Low Disable Request Enable 0: Same time output level low disable request disabled 1: Same time output level low disable request enabled	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The GTINTAD enables or disables interrupt requests, A/D conversion start request, and output disable requests.

ADTRAUEN bit (GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by GTADTRA register compare matches during GTCNT counter up-counting.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

ADTRADEN bit (GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by GTADTRA register compare matches during GTCNT counter down-counting.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

ADTRBUEN bit (GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by GTADTRB register compare matches during GTCNT counter up-counting.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

ADTRBDEN bit (GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by GTADTRB register compare matches during GTCNT counter down-counting.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

GRP[1:0] bits (Output Disable Source Select)

These bits select the group of output disable request from GPT to POEG and the group of output disable for GTIOCnA pin and GTIOCnB pin from POEG to GPT.

The output disable request to POEG is output to the group selected in the GRP[1:0] bit, with dead-time errors, simultaneous high output, and simultaneous low output factors following their respective disable request enable bits.

GTST.ODF shows the request of the output disable source group that is selected with the GRP[1:0] bits. Set the GRP[1:0] bits when both GTIOR.OAE and GTIOR.OBE bits are 0.

If the POEG other than the group connected to GPT is selected in the GRP[1:0] bits, ODF flag is always 0, and the status never change to the output disable state.

GRPABH bit (Same Time Output Level High Disable Request Enable)

The GRPABH bit enables or disables the output disable request when the GTIOCnA pin and GTIOCnB pin output 1 at the same time.

GRPABL bit (Same Time Output Level Low Disable Request Enable)

The GRPABL bit enables or disables the output disable request when the GTIOCnA pin and GTIOCnB pin output 0 at the same time.

21.2.16 GTST : General PWM Timer Status Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x3C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PCF	OABLF	OABHF	—	—	—	—	ODF	—	—	—	—	ADTRBDF	ADTRBUF	ADTRADF	ADTRAUF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TUCF	—	—	—	—	—	—	—	TCFPU	TCFPO	TCFF	TCFE	TCFD	TCFC	TCFB	TCFA
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCFA	Input Capture/Compare Match Flag A 0: No input capture/compare match of GTCCRA is generated 1: An input capture/compare match of GTCCRA is generated	R/W*1
1	TCFB	Input Capture/Compare Match Flag B 0: No input capture/compare match of GTCCRB is generated 1: An input capture/compare match of GTCCRB is generated	R/W*1
2	TCFC	Input Compare Match Flag C 0: No compare match of GTCCRC is generated 1: A compare match of GTCCRC is generated	R/W*1
3	TCFD	Input Compare Match Flag D 0: No compare match of GTCCRD is generated 1: A compare match of GTCCRD is generated	R/W*1
4	TCFE	Input Compare Match Flag E 0: No compare match of GTCCRE is generated 1: A compare match of GTCCRE is generated	R/W*1
5	TCFF	Input Compare Match Flag F 0: No compare match of GTCCRF is generated 1: A compare match of GTCCRF is generated	R/W*1
6	TCFPO	Overflow Flag 0: No overflow (crest) occurred 1: An overflow (crest) occurred	R/W*1
7	TCFPU	Underflow Flag 0: No underflow (trough) occurred 1: An underflow (trough) occurred	R/W*1
14:8	—	These bits are read as 0. The write value should be 0.	R/W
15	TUCF	Count Direction Flag 0: GTCNT counter counts downward 1: GTCNT counter counts upward	R
16	ADTRAUF	GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Flag 0: No GTADTRA register compare match has occurred in up-counting. 1: A GTADTRA register compare match has occurred in up-counting.	R/W*1
17	ADTRADF	GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Flag 0: No GTADTRA register compare match has occurred in down-counting. 1: A GTADTRA register compare match has occurred in down-counting.	R/W*1
18	ADTRBUF	GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Flag 0: No GTADTRB register compare match has occurred in up-counting. 1: A GTADTRB register compare match has occurred in up-counting.	R/W*1

Bit	Symbol	Function	R/W
19	ADTRBDF	GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Flag 0: No GTADTRB register compare match has occurred in down-counting. 1: A GTADTRB register compare match has occurred in down-counting.	R/W*1
23:20	—	These bits are read as 0. The write value should be 0.	R/W
24	ODF	Output Disable Flag 0: No output disable request is generated 1: An output disable request is generated	R
28:25	—	These bits are read as 0. The write value should be 0.	R/W
29	OABHF	Same Time Output Level High Flag 0: No simultaneous generation of 1 both for the GTIOCA and GTIOCB pins has occurred. 1: A simultaneous generation of 1 both for the GTIOCA and GTIOCB pins has occurred.	R
30	OABLF	Same Time Output Level Low Flag 0: No simultaneous generation of 0 both for the GTIOCA and GTIOCB pins has occurred. 1: A simultaneous generation of 0 both for the GTIOCA and GTIOCB pins has occurred.	R
31	PCF*2	Period Count Function Finish Flag 0: No period count function finish has occurred 1: A period count function finish has occurred	R/W*1

Note 1. Only 0 can be written to this bit. Do not write 1.
When clearing the ADTRAUF, ADTRADF, ADTRBUF, or ADTRBDF flag, be sure to write 0 only to the target flag or flags for clearing and to write 1 to the other flags not for clearing.
Note 2. This bit is only available in GPT320 to GPT323, GPT168 to GPT1610. In GPT324 to GPT327, GPT1611 to GPT1613, this bit is read as 0. The write value should be 0.

The GTST indicates the status of the GPT.

TCFA flag (Input Capture/Compare Match Flag A)

The TCFA flag indicates the status for the input capture or compare match of GTCCRA.

[Setting conditions]

- GTCNT = GTCCRA, when the GTCCRA register functions as a compare match register
- GTCNT counter value is transferred to GTCCRA by the input capture signal when the GTCCRA register functions as an input capture register.

[Clearing condition]

- 0 is written to this flag.

TCFB flag (Input Capture/Compare Match Flag B)

The TCFB flag indicates the status for the input capture or compare match of GTCCRB.

[Setting conditions]

- GTCNT = GTCCRB, when the GTCCRB register functions as a compare match register
- GTCNT counter value is transferred to GTCCRB by the input capture signal when the GTCCRB register functions as an input capture register.

[Clearing condition]

- 0 is written to this flag.

TCFC flag (Input Compare Match Flag C)

The TCFC flag indicates the status for the compare match of GTCCRC.

When GTCCRC performs buffer operation, GTCCRC doesn't perform compare match.

[Setting condition]

- GTCNT = GTCCRC.

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 01b, 10b, 11b (GTCCRC performs buffer operation).

TCFD flag (Input Compare Match Flag D)

The TCFD flag indicates the status for the compare match of GTCCRD.

When GTCCRD performs buffer operation, GTCCRD doesn't perform compare match.

[Setting condition]

- GTCNT = GTCCRD.

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (Triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 10b, 11b (GTCCRD performs buffer operation).

TCFE flag (Input Compare Match Flag E)

The TCFE flag indicates the status for the compare match of GTCCRE.

When GTCCRE performs buffer operation, GTCCRE doesn't perform compare match.

[Setting condition]

- GTCNT = GTCCRE.

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (Triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 01b, 10b, 11b (GTCCRE performs buffer operation).

TCFF flag (Input Compare Match Flag F)

The TCFF flag indicates the status for the compare match of GTCCRF.

When GTCCRF performs buffer operation, GTCCRF doesn't perform compare match.

[Setting condition]

- GTCNT = GTCCRF.

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)

- GTCR.MD[2:0] = 110b (Triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 10b, 11b (GTCCRF performs buffer operation).

TCFPO flag (Overflow Flag)

The TCFPO flag indicates when an overflow or crest has occurred.

[Setting conditions]

- In saw-wave mode, an overflow (GTCNT changes from GTPR to 0 in up-counting) has occurred
- In triangle-wave mode, a crest (GTCNT changes from GTPR to GTPR - 1) has occurred
- In counting by hardware sources, an overflow (GTCNT changes from GTPR to 0 in up-counting) has occurred.

[Clearing condition]

- 0 is written to this flag.

TCFPU flag (Underflow Flag)

The TCFPU flag indicates when an underflow or trough has occurred.

[Setting conditions]

- In saw-wave mode, an underflow (GTCNT changes from 0 to GTPR in down-counting) has occurred
- In triangle-wave mode, a trough (GTCNT changes from 0 to 1) has occurred
- In counting by hardware sources, an underflow (GTCNT changes from 0 to GTPR in down-counting) has occurred.

[Clearing condition]

- 0 is written to this bit.

TUCF flag (Count Direction Flag)

The TUCF flag indicates the count direction of GTCNT. In event count operation, this flag is set to 1 in up-counting and to 0 in down-counting.

ADTRAUF flag (GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRA register compare match in up-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRA register in up-counting.

[Clearing condition]

- 0 is written to the ADTRAUF flag.

ADTRADF flag (GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRA register compare match in down-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRA register in down-counting.

[Clearing condition]

- 0 is written to the ADTRADF flag.

ADTRBUF flag (GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRB register compare match in up-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRB register in up-counting.

[Clearing condition]

- 0 is written to the ADTRBUF flag.

ADTRBDF flag (GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRB register compare match in down-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRB register in down-counting.

[Clearing condition]

- 0 is written to the ADTRBDF flag.

ODF flag (Output Disable Flag)

The ODF flag shows the request of the output disable source group that is selected in the GRP[1:0] bits.

When output is disabled, an output disable control is not released within the same cycle in which an output disable request is negated. It is released in the next cycle.

OABHF flag (Same Time Output Level High Flag)

The OABHF flag indicates that the GTIOCnA pin and GTIOCnB pin output 1 at the same time.

When the GTIOCnA or GTIOCnB pin outputs 0, this flag returns to 0. This flag is read only. Writing 0 to clear the flag is prohibited.

When the output disable request by the OABHF flag is enabled (GTINTAD.GRPABH = 1), the OABHF flag is output to POEG as an output disable request. The GPT does not have an interrupt to indicate that outputs have been simultaneous driven to the high level. Use the interrupt function in the POEG if this is necessary.

[Setting condition]

- The GTIOCnA and GTIOCnB pins output 1 at the same time when both OAE and OBE bits are set to 1.

[Clearing conditions]

- The GTIOCnA pin output value is different from the GTIOCnB pin output value when both OAE and OBE bits are set to 1
- The GTIOCnA and GTIOCnB pins output 0 at the same time when both OAE and OBE bits are set to 1
- Either the OAE bit or OBE bit is set to 0.

OABLF flag (Same Time Output Level Low Flag)

The OABLF flag indicates that the GTIOCnA and GTIOCnB pins output 0 at the same time.

When the GTIOCnA pin or GTIOCnB pin outputs 1, this flag returns to 0. This flag is read only. Writing 0 to clear the flag is prohibited.

When the output disable request by the OABLF flag is enabled (GTINTAD.GRPABL = 1), the OABLF flag is output to POEG as an output disable request. The GPT does not have an interrupt to indicate that outputs have been simultaneous driven to the low level. Use the interrupt function in the POEG if this is necessary.

[Setting condition]

- The GTIOCnA and GTIOCnB pins output 0 at the same time when both OAE and OBE bits are set to 1.

[Clearing conditions]

- The GTIOCnA pin output value is different from the GTIOCnB pin output value when both OAE and OBE bits are set to 1
- The GTIOCnA and GTIOCnB pins output 1 at the same time when both OAE and OBE bits are set to 1
- Either the OAE bit or the OBE bit is set to 0.

The compare-target signals to generate the OABHF/OABLF flag are the compare match outputs (PWM outputs) signals before they are masked by the output disable function. Even during the output disable condition, compare match operation continues internally, where the OABHF or OABLF flag is updated based on the operation results.

PCF flag (Period Count Function Finish Flag)

This bit is status flag of period count function finish.

[Setting condition]

- The GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 1 at the end of cycle.
- The GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 0 at the count clock.

[Clearing condition]

- 0 is written to this bit.

21.2.17 GTBER : General PWM Timer Buffer Enable Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	ADTDB	ADTTB[1:0]		—	ADTDA	ADTTA[1:0]		—	CCRSWT	PR[1:0]		CCRB[1:0]		CCRA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	BD2	BD1	BD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BD0	GTCCR Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
1	BD1	GTPR Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
2	BD2	GTADTRA/GTADTRB Registers Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
17:16	CCRA[1:0]	GTCCRA Buffer Operation 0 0: No buffer operation 0 1: Single buffer operation (GTCCRA ↔GTCCRC) Others: Double buffer operation (GTCCRA ↔ GTCCRC ↔ GTCCRD)	R/W
19:18	CCRB[1:0]	GTCCRB Buffer Operation 0 0: No buffer operation 0 1: Single buffer operation (GTCCRB ↔ GTCCRE) Others: Double buffer operation (GTCCRB ↔ GTCCRE ↔ GTCCRF)	R/W
21:20	PR[1:0]	GTPR Buffer Operation 0 0: No buffer operation 0 1: Single buffer operation (GTPBR → GTPR) Others: Setting prohibited	R/W
22	CCRSWT	GTCCRA and GTCCRB Forcible Buffer Operation Writing 1 to this bit forces a buffer transfer of GTCCRA and GTCCRB. This bit automatically returns to 0 after 1 is written. This bit is read as 0.	W
23	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
25:24	ADTTA[1:0]	GTADTRA Register Buffer Transfer Timing Select 0 0: In triangle wave mode, no transfer. In saw-wave mode, no transfer. 0 1: In triangle wave mode, transfer at crest. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing. 1 0: In triangle wave mode, transfer at trough. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing. 1 1: In triangle wave, transfer at both crest and trough. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing.	R/W
26	ADTDA	GTADTRA Register Double Buffer Operation 0: Single buffer operation (GTADTBRA → GTADTRA) 1: Double buffer operation (GTADTDBRA → GTADTBRA → GTADTRA)	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
29:28	ADTTB[1:0]	GTADTRB Register Buffer Transfer Timing Select 0 0: In triangle wave mode, no transfer. In saw-wave mode, no transfer. 0 1: In triangle wave mode, transfer at crest. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing. 1 0: In triangle wave mode, transfer at trough. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing. 1 1: In triangle wave mode, transfer at both crest and trough. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing.	R/W
30	ADTDB	GTADTRB Register Double Buffer Operation 0: Single buffer operation (GTADTBRB → GTADTRB) 1: Double buffer operation (GTADTDBRB → GTADTBRB → GTADTRB)	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The GTBER register provides settings for the buffer operation. Set the GTBER register while the GTCNT counter is stopped.

BD0 bit (GTCCR Buffer Operation Disable)

The BD0 bit disables the buffer operation using GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, and GTCCRF combined.

When GTDTCR.TDE is 1 and when BD0 is set to 0, GTCCRB does not perform buffer operation. The GTCCRB register is automatically set to a compare match value for negative-phase waveform with dead time.

A value for the BD0 bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the GTSECR.SBDCE or GTSECR.SBDCD.

BD1 bit (GTPR Buffer Operation Disable)

The BD1 bit disables the buffer operation using GTPR and GTPBR combined.

A value for the BD1 bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the GTSECR.SBDPE or GTSECR.SBDPD.

BD2 bit (GTADTRA/GTADTRB Registers Buffer Operation Disable)

This bit disables buffer operation using the GTADTRA, GTADTBRA, and GTADTDBRA registers together and buffer operation using the GTADTRB, GTADTBRB, and GTADTDBRB registers together.

The setting is invalid during the event count operation, and the buffer operation using the GTADTRA and GTADTRB registers is not performed.

A value for the BD2 bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the SBDAE or SBDAD bit in the GTSECR register.

CCRA[1:0] bits (GTCCRA Buffer Operation)

The CCRA[1:0] bits set the buffer operation with GTCCRA, GTCCRC, and GTCCRD combined. When the buffer operation is restricted by the operating mode set in GTCR, the GTCR setting is given priority.

The buffer operation mode is fixed in saw-wave one-shot pulse mode or triangle-wave PWM mode 3 (64-bit transfer at trough).

CCRB[1:0] bits (GTCCRB Buffer Operation)

The CCRB[1:0] bits set the buffer operation using GTCCRB, GTCCRE, and GTCCRF combined. When the buffer operation is restricted by the operating mode set in GTCR, the GTCR setting is given priority.

The buffer operation mode is fixed in saw-wave one-shot pulse mode or triangle-wave PWM mode 3 (64-bit transfer at trough).

PR[1:0] bits (GTPR Buffer Operation)

The PR[1:0] bits set the buffer operation with GTPR and GTPBR combined.

CCRSWT bit (GTCCRA and GTCCRB Forcible Buffer Operation)

Writing 1 to the CCRSWT bit forces a buffer transfer of GTCCRA and GTCCRB. This bit automatically returns to 0 after the 1 is written. This bit is read as 0, and is valid only when counting is stopped with a compare match operation specified.

ADTTA[1:0] bits (GTADTRA Register Buffer Transfer Timing Select)

These bits set the transfer timing for buffer operation of the GTADTRA, GTADTBRA, and GTADTDBRA registers.

The setting is invalid during the event count operation.

ADTDA bit (GTADTRA Register Double Buffer Operation)

These bits set buffer operation with the GTADTRA, GTADTBRA, and GTADTDBRA registers combined.

The setting is invalid during the event count operation.

ADTTB[1:0] bits (GTADTRB Register Buffer Transfer Timing Select)

These bits set the transfer timing for buffer operation of the GTADTRB, GTADTBRB, and GTADTDBRB registers.

The setting is invalid during the event count operation.

ADTDB bit (GTADTRB Register Double Buffer Operation)

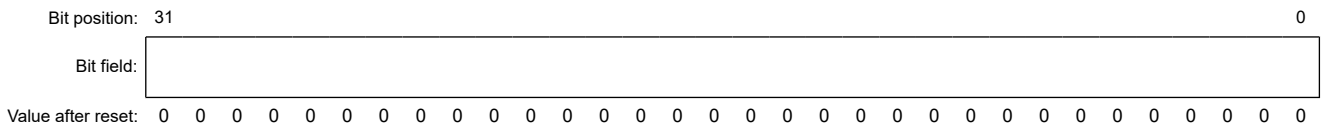
These bits set buffer operation with the GTADTRB, GTADTBRB, and GTADTDBRB registers combined.

The setting is invalid during the event count operation.

21.2.18 GTCNT : General PWM Timer Counter

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x48

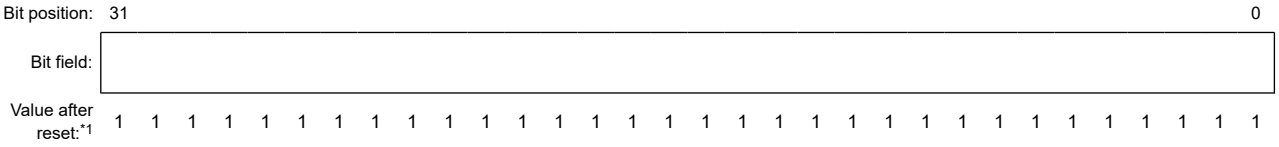


Bit	Symbol	Function	R/W
31:0	n/a	GTCNT is a 32-bit read/write counter for GPT32n (n = 0 to 7). For GPT16m (m = 8 to 13), GTCNT is a 16-bit register. GTCNT can only be written to after counting stops. For GPT16m (m = 8 to 13), the upper 16 bits for access in a 32-bit unit are always read as 0x0000, and writing to these bits is ignored. GTCNT must be set within the range of 0 ≤ GTCNT ≤ GTPR.	R/W

21.2.19 GTCCRk : General PWM Timer Compare Capture Register k (k = A to F)

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x4C (GTCCRA)
0x50 (GTCCRB)
0x54 (GTCCRC)
0x58 (GTCCRE)
0x5C (GTCCRD)
0x60 (GTCCRF)



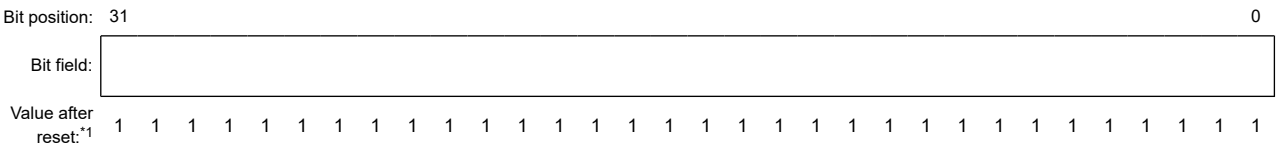
Bit	Symbol	Function	R/W
31:0	n/a	GTCCRk registers are read/write registers. The effective size of GTCCRk is the same as GTCNT (16- or 32-bit). If the effective size of GTCCRk is 16 bits, the upper 16 bits for access in a 32-bit unit are always read as 0x0000, and writing to these bits is ignored. GTCCRA and GTCCRB are registers used for both output compare and input capture. GTCCRC and GTCCRE are compare match registers, and can also function as buffer registers for GTCCRA and GTCCRB. GTCCRD and GTCCRF are compare match registers, and can also function as buffer registers for GTCCRC and GTCCRE (double-buffer registers for GTCCRA and GTCCRB).	R/W

Note 1. For GPT16m (m = 8 to 13), the value of the upper 16 bits after reset is 0x0000.

21.2.20 GTPR : General PWM Timer Cycle Setting Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x64



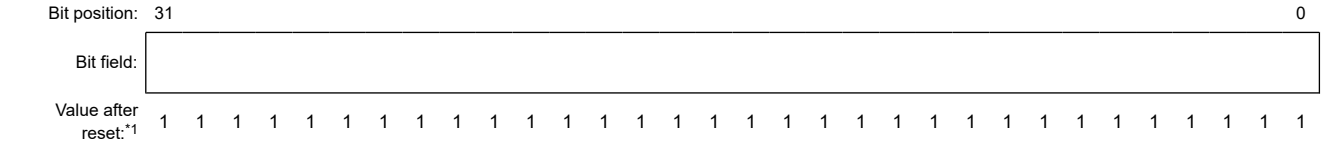
Bit	Symbol	Function	R/W
31:0	n/a	GTPR is a read/write register that sets the maximum count value of GTCNT. The effective size of GTPR is the same as GTCNT (16- or 32-bit). If the effective size of GTPR is 16 bits, the upper 16 bits for access in a 32-bit unit are always read as 0x0000, and writing to these bits is ignored. For saw waves, the value of (GTPR + 1) is the cycle. For triangle waves, the value of (GTPR value × 2) is the cycle.	R/W

Note 1. For GPT16m (m = 8 to 13), the value of the upper 16 bits after reset is 0x0000.

21.2.21 GTPBR : General PWM Timer Cycle Setting Buffer Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x68



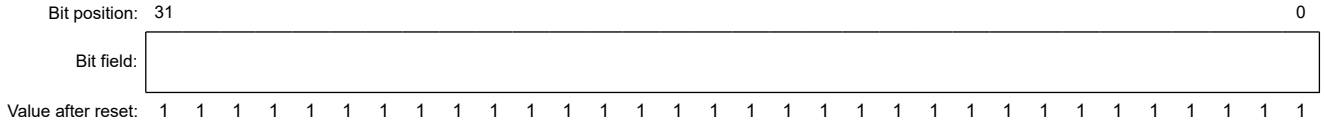
Bit	Symbol	Function	R/W
31:0	n/a	GTPBR is a read/write register that functions as a buffer register for GTPR. The effective size of GTPBR is the same as GTCNT (16- or 32-bit). If the effective size of GTPBR is 16 bits, the upper 16 bits for access in a 32-bit unit are always read as 0x0000, and writing to these bits is ignored.	R/W

Note 1. For GPT16m (m = 8 to 13), the value of the upper 16 bits after reset is 0x0000.

21.2.22 GTADTRk : A/D Conversion Start Request Timing Register k (k = A, B)

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)

Offset address: 0x70 (GTADTRA)
0x7C (GTADTRB)

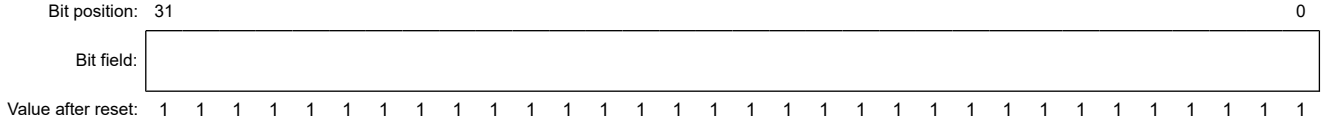


Bit	Symbol	Function	R/W
31:0	n/a	Set the timing of A/D conversion start request generation Access in 8-bit or 16-bit units to the GTADTRk register is prohibited, and it should be accessed in 32-bit units. When the GTADTRk register value matches the GTCNT counter value, an A/D conversion start request is generated.	R/W

21.2.23 GTADTBRk : A/D Conversion Start Request Timing Buffer Register k (k = A, B)

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)

Offset address: 0x74 (GTADTBRA)
0x80 (GTADTBRB)



Bit	Symbol	Function	R/W
31:0	n/a	The buffer registers for the GTADTRk register Access in 8-bit or 16-bit units to the GTADTBRk register is prohibited, and it should be accessed in 32-bit units.	R/W

21.2.24 GTADTDBRk : A/D Conversion Start Request Timing Double-Buffer Register k (k = A, B)

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)

Offset address: 0x78 (GTADTDBRA)
0x84 (GTADTDBRB)

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The buffer registers for the GTADTBRk register (double buffer registers for the GTADTRk register) Access in 8-bit or 16-bit units to the GTADTDBRk register is prohibited, and it should be accessed in 32-bit units.	R/W

21.2.25 GTDTCR : General PWM Timer Dead Time Control Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x88

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDE	Negative-Phase Waveform Setting 0: GTCCRB is set without using GTDVU 1: GTDVU is used to set the compare match value for negative-phase waveform with dead time automatically in GTCCRB	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

GTDTCR enables automatic setting of a compare match value for negative-phase waveform with dead time.

The setting is invalid during the event count operation.

TDE bit (Negative-Phase Waveform Setting)

The TDE bit specifies whether to use GTDVU. When GTDVU is used, the compare match value for a negative-phase waveform with dead time obtained by the compare match value of a positive-phase waveform (GTCCRA) and the dead time value (GTDVU) is automatically set in GTCCRB.

The TDE bit setting is ignored in saw-wave PWM mode, and the GTCCRB is not automatic setting.

The GTCCRB value is automatically set and has the following upper and lower limit values. If the obtained GTCCRB value is not within the upper or lower limit, the following limit value is set in GTCCRB.

- Triangle waves:
  Upper limit value: GTPR − 1
  Lower limit value: 1 in up-counting, 0 in down-counting

- Saw-wave one-shot pulse mode:
  Upper limit value: GTPR
  Lower limit value: 0.

21.2.26 GTDVU : General PWM Timer Dead Time Value Register U

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0x8C

Bit position:	31																													0
Bit field:																														
Value after reset:*1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	GTDVU is a read/write register that sets the dead time for generating PWM waveforms with dead time. The effective size of GTDVU is the same as GTCNT (16 or 32 bits). If the effective size of GTDVU is 16 bits, the upper 16 bits for access in a 32-bit unit are always read as 0x0000, and writing to these bits is ignored. Setting a GTDVU value greater than or equal to GTPR is prohibited. When using the automatic dead time setting function, do not set a value that makes a change point of the waveform exceeding the count period. The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. When GTDVU is used, writing to GTCCRB is prohibited. When this register is set to 0, waveforms without dead time are output.	R/W

Note 1. For GPT16m (m = 8 to 13), the value of the upper 16 bits after reset is 0x0000.

21.2.27 GTADSMR : General PWM Timer A/D Conversion Start Request Signal Monitoring Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0xA4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	ADSMEN1	—	—	—	—	—	—	ADSMS1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADSMEN0	—	—	—	—	—	—	ADSMS0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ADSMS0[1:0]	A/D Conversion Start Request Signal Monitor 0 Selection 0 0: A/D conversion start request signal generated by the GTADTRA register during up-counting. 0 1: A/D conversion start request signal generated by the GTADTRA register during down-counting. 1 0: A/D conversion start request signal generated by the GTADTRB register during up-counting. 1 1: A/D conversion start request signal generated by the GTADTRB register during down-counting.	R/W

Bit	Symbol	Function	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	ADSMEN0	A/D Conversion Start Request Signal Monitor 0 Output Enabling 0: Output of A/D conversion start request signal monitor 0 is disabled. 1: Output of A/D conversion start request signal monitor 0 is enabled.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
17:16	ADSMS1[1:0]	A/D Conversion Start Request Signal Monitor 1 Selection 0 0: A/D conversion start request signal generated by the GTADTRA register during up-counting. 0 1: A/D conversion start request signal generated by the GTADTRA register during down-counting. 1 0: A/D conversion start request signal generated by the GTADTRB register during up-counting. 1 1: A/D conversion start request signal generated by the GTADTRB register during down-counting.	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
24	ADSMEN1	A/D Conversion Start Request Signal Monitor 1 Output Enabling 0: Output of A/D conversion start request signal monitor 1 is disabled. 1: Output of A/D conversion start request signal monitor 1 is enabled.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The GTADSMR register is used to control monitors for the A/D conversion start request signal that is synchronized with a frame period.

ADSMSk[1:0] bits (A/D Conversion Start Request Signal Monitor k Selection) (k = 0, 1)

These bits are used to select A/D conversion start request signal synchronized with a frame period which is monitored by the GTASMk pin.

In triangle-wave PWM mode, the following settings are prohibited:

- Set ADSMSk[1:0] bit to 00b (A/D conversion start request during up-counting) when GTADTRA = 0
- Set ADSMSk[1:0] bit to 10b (A/D conversion start request during up-counting) when GTADTRB = 0
- Set ADSMSk[1:0] bit to 01b (A/D conversion start request during down-counting) when GTADTRA = GTPR
- Set ADSMSk[1:0] bit to 11b (A/D conversion start request during down-counting) when GTADTRB = GTPR

ADSMENk bit (A/D Conversion Start Request Signal Monitor k Output Enabling) (k = 0, 1)

This bit enables or disables the monitor output to the GTADSMk pin.

When the output is disabled, the GTADSMk pin goes to the low level.

When the bit is 1, the signal on the GTADSMk pin goes to the high level on assertion of the signal to request to the start of A/D conversion selected by the ADSMSk[1:0] bits and returns to the low level at the end of the current cycle of the timer for the channel that generated the given signal to request the start of A/D conversion. When the counter stops, the value when the counter stopped is retained for output. Set the ADSMENk bit to 0 to output the low level.

When a signal to request the start of A/D conversion is generated at the end of a timer period, the generation of this signal has priority in terms of monitoring output and the output remains at the high level till the end of the next period.

When the output of the same A/D conversion start request signal monitoring output is enabled for multiple channels, ORed signals will be output from the GPT.

21.2.28 GTICLF : General PWM Timer Inter Channel Logical Operation Function Setting Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0xB8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	ICLFSELD[5:0]						—	ICLFB[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	ICLFSELC[5:0]						—	ICLFA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	ICLFA[2:0]	GTIOCnA Output Logical Operation Function Select 0 0 0: A (no delay) 0 0 1: NOT A (no delay) 0 1 0: C (1 PCLKD delay) 0 1 1: NOT C (1 PCLKD delay) 1 0 0: A AND C (1 PCLKD delay)*2 1 0 1: A OR C (1 PCLKD delay)*2 1 1 0: A EXOR C (1 PCLKD delay)*2 1 1 1: A NOR C (1 PCLKD delay)*2	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
9:4	ICLFSELC[5:0]	Inter Channel Signal C Select*1*2 0x00: GTIOC0A 0x01: GTIOC0B 0x02: GTIOC1A 0x03: GTIOC1B ⋮ ⋮ 0x1A: GTIOC13A 0x1B: GTIOC13B Others: Setting prohibited	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
18:16	ICLFB[2:0]	GTIOCnB Output Logical Operation Function Select 0 0 0: B (no delay) 0 0 1: NOT B (no delay) 0 1 0: D (1 PCLKD delay) 0 1 1: NOT D (1 PCLKD delay) 1 0 0: B AND D (1 PCLKD delay)*3 1 0 1: B OR D (1 PCLKD delay)*3 1 1 0: B EXOR D (1 PCLKD delay)*3 1 1 1: B NOR D (1 PCLKD delay)*3	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
25:20	ICLFSELD[5:0]	Inter Channel Signal D Select*1*3 0x00: GTIOC0A 0x01: GTIOC0B 0x02: GTIOC1A 0x03: GTIOC1B ⋮ ⋮ 0x1A: GTIOC13A 0x1B: GTIOC13B Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note: n = 0 to 13
Note 1. The signal before performing output disable control is selected.
Note 2. When channel's own GTIOCnA is selected, C is treated as "1".
Note 3. When channel's own GTIOCnB is selected, D is treated as "1".

The GTICLF register sets the logical operation function between compare match outputs. The logical operation is performed with the signals that the duty 0%/100% control is performed after compare match control. (The output disable control is performed with the signal after logical operation.)

Access in 8-bit units to GTICLF is prohibited.

ICLFm[2:0] bit (GTIOCnm Output Logical Operation Function Select) (m = A, B)

These bits select the logical operation function between signals before performing output disable control for GTIOCnm. To prevent hazard to the GPT output, the signal after logical operation is latched with PCLKD. After latching, the output disable control is performed. When the logical operation function which causes the delay of 1 PCLKD is selected, the output enable signal is also delayed with 1 PCLKD and input to the output disable control.

When the same signal to operate logical function AND, OR, EXOR and NOR is selected, one signal is treated as "1".

ICLFSELk[5:0] bit (Inter Channel Signal k Select) (k = C, D)

These bits select the signal k that the logical operation is performed with the signal before performing output disable control for GTIOCnm.

21.2.29 GTPC : General PWM Timer Period Count Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 3)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 3)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 10)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 10)

Offset address: 0xBC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	PCNT[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ASTP	—	—	—	—	—	—	—	PCEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PCEN	Period Count Function Enable 0: Period count function is disabled 1: Period count function is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	ASTP	Automatic Stop Function Enable 0: Automatic stop function is disabled 1: Automatic stop function is enabled	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
27:16	PCNT[11:0]	Period Counter Counter for the number of period	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The GTPC register counts the number of period.

PCEN bit (Period Count Function Enable)

This bit enables or disables period count function.

Writing is available when counting is both in progress and stopped.

When 1 is written to either the GTSECR.SPCE bit or the GTSECR.SPCD bit, the value is simultaneously set to the PCEN bit in the channels set to 1 by the GTSECSR register.

ASTP bit (Automatic Stop Function Enable)

This bit enables or disables the GTCNT counter automatic stopping after finishing counting the number of period.

When the PCEN bis is 0, writing is available.

When the PCEN bit is 1, writing is disabled.

When the PCEN bit is 1, the ASTP bit is 1, and the PCNT counter is stopped at PCNT = 0, the GTCNT counter is also stopped. When the ASTP bit is 0, the GTCNT counter continues to count.

PCNT[11:0] bits (Period Counter)

This counter counts the number of period.

When the PCEN bis is 0, writing the number of period is available.

When the PCEN bit is 1, writing is disabled, and down-counting is performed at the end of period. In saw-wave mode, the end of period refers to overflow, underflow, or counter clearing. In triangle-wave mode, it refers to trough.

When the PCNT counter is 1 at the end of period, it becomes 0 and counting is stopped.

When the GTCNT counter is stopped while period count function is enabled, the PCNT counter keeps its value. When the GTCNT counter restarts counting and the PCEN bit is 1, the PCNT counter restarts down-counting from the hold value.

When the PCEN bit is changed from 0 to 1 while the PCNT counter is 0 and the ASTP bit is 1, the GTCNT counter is stopped at the count clock immediately after that.

21.2.30 GTSECSR : General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0xD0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	SECSEL13	SECSEL12	SECSEL11	SECSEL10	SECSEL9	SECSEL8	SECSEL7	SECSEL6	SECSEL5	SECSEL4	SECSEL3	SECSEL2	SECSEL1	SECSEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SECSEL0	Channel 0 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
1	SECSEL1	Channel 1 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
2	SECSEL2	Channel 2 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W

Bit	Symbol	Function	R/W
3	SECSEL3	Channel 3 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
4	SECSEL4	Channel 4 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
5	SECSEL5	Channel 5 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
6	SECSEL6	Channel 6 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
7	SECSEL7	Channel 7 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
8	SECSEL8	Channel 8 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
9	SECSEL9	Channel 9 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
10	SECSEL10	Channel 10 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
11	SECSEL11	Channel 11 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
12	SECSEL12	Channel 12 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
13	SECSEL13	Channel 13 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The GTSECSR register selects an intended channel n (n = 0 to 13) for updating an operation enable bit by the GTSECR register. A bit position for the GTSECSR register indicates a channel number. The GTSECSR register of each channel is a common register, and writing 1 to a bit in the GTSECSR register in any channel and updating it changes a channel, related to the position of the bit written with 1 by the GTSECSR register, to be simultaneously controlled of the operation enable bit by the GTSECR register.

Contrary to a security attribute for write or read access, if the security and privilege attributes set on each channel has the security violation or the privilege violation, the bit of the channel number that violates security cannot be written or read, and the read value of the bit is "0".

For the association between the GTSECSR bit number and a channel number, see Figure 21.2.

Access in 8-bit or 16-bit units to GTSECSR is prohibited, and it should be accessed in 32-bit units.

SECSELn bit (Operation Enable Bit Simultaneous Control Channel Select) (n = 0 to 13)

This bit enables or disables the simultaneous control of operation enable in channel n.

When the bit is set to 1, the simultaneous control is enabled, and disabled when the bit is 0.

21.2.31 GTSECR : General PWM Timer Operation Enable Bit Simultaneous Control Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 7)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 7)
GPT16m = 0x4032_2000 + 0x0100 × m (m = 8 to 13)
GPT16m_NS = 0x5032_2000 + 0x0100 × m (m = 8 to 13)

Offset address: 0xD4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	SPCD	—	—	—	—	—	—	—	SPCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SBDAD	SBDPD	SBDCD	—	—	—	—	—	SBDAE	SBDPE	SBDCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SBDCE	GTCCR Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTCCR buffer operations 1: Enable GTCCR register buffer operations simultaneously	R/W
1	SBDPE	GTPR Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTPR buffer operations 1: Enable GTPR register buffer operations simultaneously	R/W
2	SBDAE	GTADTR Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTADTR buffer operations 1: Enable GTADTR register buffer operations simultaneously	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	SBDCD	GTCCR Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTCCR buffer operations 1: Disable GTCCR register buffer operations simultaneously	R/W
9	SBDPD	GTPR Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTPR buffer operations 1: Disable GTPR register buffer operations simultaneously	R/W
10	SBDAD	GTADTR Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTADTR buffer operations 1: Disable GTADTR register buffer operations simultaneously	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
16	SPCE	Period Count Function Simultaneous Enable*1 0: Disable simultaneous enabling period count function 1: Enable period count function simultaneously	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
24	SPCD	Period Count Function Simultaneous Disable*1 0: Disable simultaneous disabling period count function 1: Disable period count function simultaneously	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit is only available in GPT320 to GPT323, GPT168 to GPT1610.
In GPT324 to GPT327, GPT1611 to GPT1613, this bit is read as 0. The write value should be 0.

The GTSECR register simultaneously updates the value for operation enable bits of a channel set by the GTSECSR register.

Writing 1 to a bit in the GTSECR register in any channel and updating it updates an operation enable bit for all channels, related to the position of the bit written with 1 by the all GTSECSR registers.

Setting enable and disable bits for the same operation enable bit to 1 in the GTSECR is prohibited.

Contrary to a security attribute for write or read access, if the security and privilege attributes set on each channel has the security violation or the privilege violation, the bit of the channel number that violates security cannot be written or read, and the read value of the bit is "0".

A bit written to 1 is automatically cleared. When the GTSECR is read, 0 is read.

Access in 8-bit or 16-bit units to the GTSECR register is prohibited, and it should be accessed in 32-bit units.

SBDCE bit (GTCCR Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[0] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTCCRA, GTCCRC, and GTCCRD registers and using the GTCCRB, GTCCRE, and GTCCRF registers are enabled.

Simultaneous setting of SBDCE and SBDCD bits to 1 is prohibited.

SBDPE bit (GTPR Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[1] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTPR and GTPBR registers are enabled.

Simultaneous setting of SBDPE and SBDPD bits to 1 is prohibited.

SBDAE bit (GTADTR Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[2] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTADTRA, GTADTBRA, and GTADTDBRA registers and using the GTADTRB, GTADTBRB, and GTADTDBRB registers are enabled.

Simultaneous setting of SBDAE and SBDAD bits to 1 is prohibited.

SBDCD bit (GTCCR Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[0] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTCCRA, GTCCRC, and GTCCRD registers and using the GTCCRB, GTCCRE, and GTCCRF registers are disabled.

Simultaneous setting of SBDCE and SBDCD bits to 1 is prohibited.

SBDPD bit (GTPR Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[1] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTPR and GTPBR registers are disabled.

Simultaneous setting of SBDPE and SBDPD bits to 1 is prohibited.

SBDAD bit (GTADTR Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[2] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTADTRA, GTADTBRA, and GTADTDBRA registers and using the GTADTRB, GTADTBRB, and GTADTDBRB registers are disabled.

Simultaneous setting of SBDAE and SBDAD bits to 1 is prohibited.

SPCE bit (Period Count Function Simultaneous Enable)

When 1 is written to this bit, 1 is simultaneously set to GTPC.PCEN bit in the channels set to 1 by the GTSECSR register, and period count function is enabled.

Simultaneous setting of SPCE and SPCD bits to 1 is prohibited.

SPCD bit (Period Count Function Simultaneous Disable)

When 1 is written to this bit, 0 is simultaneously set to GTPC.PCEN bit in the channels set to 1 by the GTSECSR register, and period count function is disabled.

Simultaneous setting of SPCE and SPCD bits to 1 is prohibited.

21.2.32 OPSCR : Output Phase Switching Control Register

Base address: GPT_OPS = 0x4032_3F00
GPT_OPS_NS = 0x5032_3F00

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFCS[1:0]		NFEN	—	—	GODF	GRP[1:0]		—	—	ALIGN	RV	INV	N	P	FB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	EN	—	W	V	U	—	WF	VF	UF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	UF	Input Phase Soft Setting These bits set the input phase from software settings. Setting these bits is valid when OPSCR.FB = 1.	R/W
1	VF		R/W
2	WF		R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	U	Input U-Phase Monitor This bit monitors the state of the input phase. OPSCR.FB = 0 : External input that are synchronized by PCLKD OPSCR.FB = 1 : Software settings (UF)	R
5	V	Input V-Phase Monitor This bit monitors the state of the input phase. OPSCR.FB = 0 : External input that are synchronized by PCLKD OPSCR.FB = 1 : Software settings (VF)	R
6	W	Input W-Phase Monitor This bit monitors the state of the input phase. OPSCR.FB = 0 : External input that are synchronized by PCLKD OPSCR.FB = 1 : Software settings (WF)	R
7	—	This bit is read as 0. The write value should be 0.	R/W
8	EN	Output Phase Enable 0: Do not output (Hi-Z external pin) 1: Output*1	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	FB	External Feedback Signal Enable This bit selects the input phase from software settings and external input. 0: Select the external input 1: Select the soft setting (OPSCR.UF, VF, WF)	R/W
17	P	Positive-Phase Output (P) Control 0: Level signal output 1: PWM signal output	R/W
18	N	Negative-Phase Output (N) Control 0: Level signal output 1: PWM signal output	R/W
19	INV	Output Phase Invert Control 0: Positive logic (active-high) output 1: Negative logic (active-low) output	R/W
20	RV	Output Phase Rotation Direction Reversal Control 0: Positive rotation 1: Reverse rotation	R/W
21	ALIGN	Input Phase Alignment 0: Input phase aligned to PCLKD 1: Input phase aligned to the falling edge of PWM	R/W

Bit	Symbol	Function	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
25:24	GRP[1:0]	Output Disabled Source Selection 0 0: Select group A output disable source 0 1: Select group B output disable source 1 0: Select group C output disable source 1 1: Select group D output disable source	R/W
26	GODF	Group Output Disable Function 0: This bit function is ignored 1: Group disable clears the OPSCR.EN bit*1	R/W
28:27	—	These bits are read as 0. The write value should be 0.	R/W
29	NFEN	External Input Noise Filter Enable 0: Do not use a noise filter on the external input 1: Use a noise filter on the external input	R/W
31:30	NFCS[1:0]	External Input Noise Filter Clock Selection Noise filter sampling clock setting of the external input. 0 0: PCLKD/1 0 1: PCLKD/4 1 0: PCLKD/16 1 1: PCLKD/64	R/W

Note 1. When OPSCR.GODF = 1 and the signal value selected by the OPSCR.GRP[1:0] bit is high, the OPSCR.EN bit is set to 0.

The OPSCR register sets the output of the signal waveform required for brushless DC motor control.

UF , VF , WF bits (Input Phase Soft Setting)

The UF , VF , WF bits set the input phase from the software settings. When OPSCR.FB bit is 1, these bits are valid. The set value of the UF /VF /WF takes the place of the U/V/W external input.

U, V, W bits (Input Phase Monitor)

When the OPSCR.FB bit is 0, external inputs that are synchronized by PCLKD are monitored by these bits. When the OPSCR.FB bit is 1, the OPSCR.U, OPSCR.V, and OPSCR.W bits can read the OPSCR.UF , OPSCR.VF , and OPSCR.WF bits.

EN bit (Output Phase Enable)

The EN bit controls the output enable signal of output phase (positive phase/negative phase).

When the OPSCR.EN bit is 1, the signal waveform is output.

When the OPSCR.EN bit is 0, first set OPSCR.FB, OPSCR.UF /VF /WF (software setting is selected), OPSCR.P/N, OPSCR.INV, OPSCR.RV, OPSCR.ALIGN, OPSCR.GRP[1:0], OPSCR.GODF, OPSCR.NFEN, OPSCR.NFCS. Then, set the EN bit to 1. The EN bit should be set when output disable request does not occur from POEG. Also when OPSCR.GODF is 1 and the signal value selected in the OPSCR.GRP[1:0] bit is high, the OPSCR.EN bit is set to 0. Even if 1 is written by software, the EN bit remains at 0.

For the return, after clearing the Output Disable Request by software, set the EN bit to 1.

Priority order of the EN bit is as follows (when the conflict occurs).

When writing 1 by software and clearing to 0 by the Output Disable Request conflict for the EN bit, clearing to 0 by the Output Disable Request is enabled.

FB bit (External Feedback Signal Enable)

The FB bit selects the input phase from the software settings (OPSCR.UF, VF, WF) and external input such as a Hall element.

P bit (Positive-Phase Output (P) Control)

The P bit selects one of the level signal output or PWM signal output for the positive-phase output (GTOUUP, GTOVUP and GTOWUP pins).

N bit (Negative-Phase Output (N) Control)

The N bit selects one of the level signal output or PWM signal output for the negative-phase output (GTOULO, GTOVLO and GTOWLO pins).

INV bit (Output Phase Invert Control)

The INV bit selects one of the positive logic (active-high) output or negative logic (active-low) output for the output phase.

RV bit (Output Phase Rotation Direction Reversal Control)

The RV bit reverses the direction of rotation of the motor by inverting the input phase.

ALIGN bit (Input Phase Alignment)

The ALIGN bit selects the PCLKD or PWM for the sampling of the input phase (input phase is specified in the OPSCR.FB bit).

When OPSCR.ALIGN bit is 0, input phase is aligned to PCLKD.

Note: When the chopping is performed, there are cases where the PWM width of output is shorter than the width of the PWM used to chop just before or after switching of output phase, depending on the phase difference between the phase output switch timing and the phase of PWM.

When OPSCR.ALIGN bit is 1, input phase is aligned with the falling edge of PWM.

GRP[1:0] bit (Output Disabled Source Selection)

The GRP[1:0] bit selects the output disable source.

The GRP bits should be set when GODF bit is 0. If GRP bits select a POEG except for the connected groups, the status of output pin never change to disable.

GODF bit (Group Output Disable Function)

When OPSCR.GODF is 1 and the signal value selected by the OPSCR.GRP[1:0] bit is high, the OPSCR.EN bit is set to 0.

When OPSCR.GODF bit is 0, this bit is ignored.

The GODF bit should be set when output disable request does not occur from POEG.

NFEN bit (External Input Noise Filter Enable)

The NFEN bit selects the noise filter for external input. When OPSCR.NFEN bit is 0, a noise filter for the external input is not used.

Note: Set this bit during the EN bit is 0 to avoid generation of unintentional internal edge caused by switching this bit.

NFCS[1:0] bits (External Input Noise Filter Clock Selection)

The NFCS[1:0] bits select the clock for the external input noise filter. When the OPSCR.NFEN bit is 1, noise filter sampling clock setting of the external input is enabled.

1. Set the NFCS[1:0].
2. Wait for 2 cycles.
3. Set the OPSCR.EN bit to 1.

21.3 Operation

21.3.1 Basic Operation

Each channel has a 32-bit and 16-bit timer that performs a periodic count operation using the count clock and hardware sources. The count function provides both up-counting and down-counting. The GTPR controls the count cycle.

When the GTCNT counter value matches the value in GTCCRA or GTCCRB, the output from the associated GTIOCnA or GTIOCnB can be changed (n = 0 to 13). GTCCRA or GTCCRB can be used as an input capture register with hardware resources.

GTCCRC and GTCCRD can function as buffer registers for GTCCRA. GTCCRE and GTCCRF can function as buffer registers for GTCCRB.

21.3.1.1 Counter Operation

(1) Counter start and stop

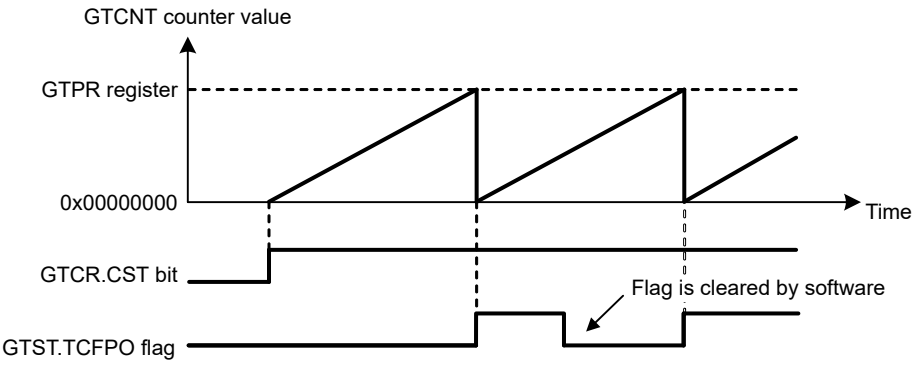
The counter of each channel starts the count operation when GTCR.CST is set to 1, and stops counting when the bit is set to 0. The GTCR.CST bit value is changed by the following sources:

- Writing to GTCR register
- Writing 1 to the bit in GTSTR associated with the GPT channel number when the GTSSR.CSTRT bit set to 1
- Writing 1 to the bit in GTSTP associated with the GPT channel number when the GTPSR.CSTOP bit set to 1
- The hardware source selected in the GTSSR register
- The hardware source selected in the GTPSR register
- Completion of the period count function while the GTPC.ASTP bit is 1

(2) Periodic count operation in up-counting by count clock

The GTCNT counter in each channel starts up-counting when the associated GTCR.CST bit is set to 1 with GTUPSR and GTDNSR registers set to 0x00000000. When the GTCNT value changes from the GTPR value to 0 (overflow), the GTST.TCFPO flag is set to 1, and the overflow interrupt (GPTn_OVF) is also generated. After GTCNT overflows, up-counting resumes from 0x00000000.

Figure 21.3 shows an example of a periodic count operation in up-counting by the count clock.



Note: The base address of registers are the same.

Figure 21.3 Example of periodic count operation in up-counting by the count clock

Table 21.5 shows an example for setting periodic count operation in up-counting by the count clock.

Table 21.5 Example for setting a periodic count operation in up-counting by the count clock

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.3, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.3, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 21.3, 0x00000000 is set.
6	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

(3) Periodic count operation in down-counting by count clock

The GTCNT counter in each channel can perform down-counting by setting GTUDDTYC.UD with GTUPSR and GTDNSR registers set to 0x00000000. When GTCNT changes from 0 to the GTPR value (underflow), GTST.TCFPU is set to 1, and the underflow interrupt(GPTn_UDF) is also generated. After the GTCNT counter underflows, down-counting resumes from the GTPR value.

Figure 21.4 shows an example of periodic count operation in down-counting by the count clock.

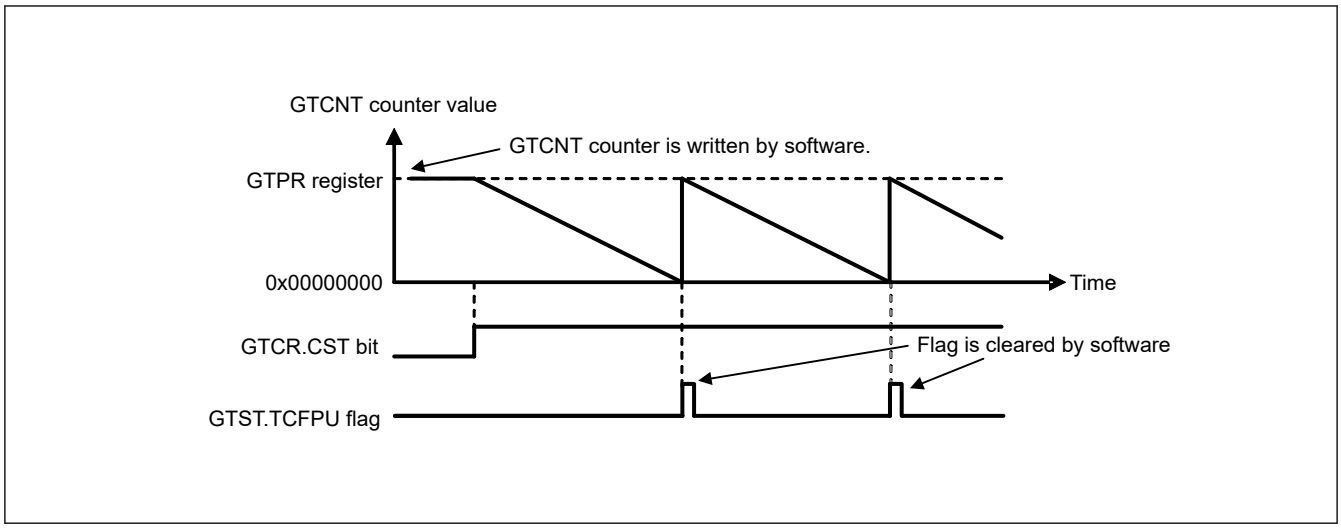


Figure 21.4 Example of periodic count operation in down-counting by the count clock

Table 21.6 shows an example for setting periodic count operation in down-counting by the count clock.

Table 21.6 Example for setting periodic count operation in down-counting by count clock

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.4, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction with the GTUDDTYC register. In Figure 21.4, after 10b is set in the GTUDDTYC[1:0] bits, 00b is set in the GTUDDTYC[1:0] bits (down-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 21.4, the GTPR register value is set.
6	Start count operation	Set the GTCR.CST bit to 1 to start count operation. In Figure 21.4, 1 is set in the CST bit.

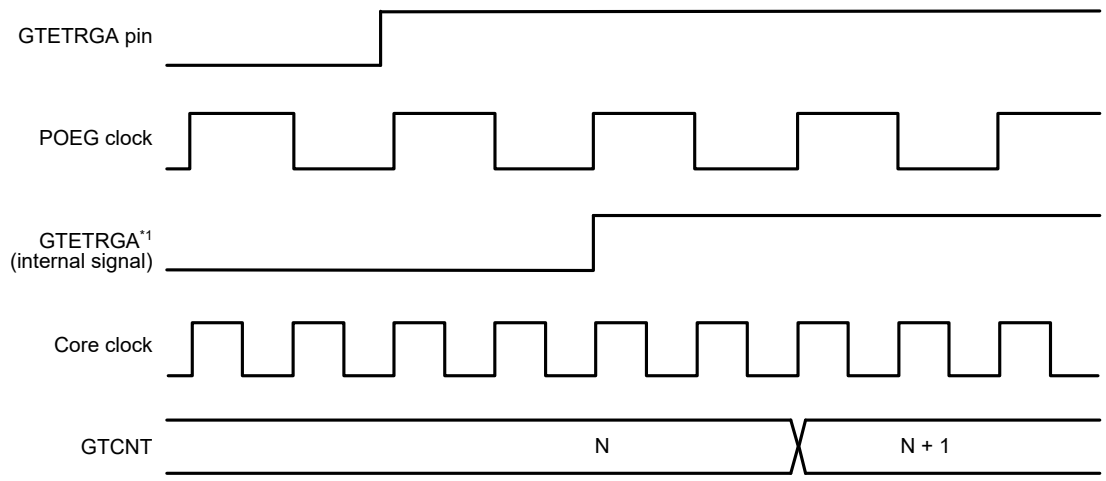
(4) Event count operation in up-counting using hardware sources

The GTCNT counter in each channel can perform up-counting using hardware sources as set in GTUPSR.

When GTUPSR is set to enable, the count clock selected in GTCR.TPCS[3:0] and the count direction selected in GTUDDTYC.UD are ignored. If up-counting and down-counting using hardware sources occur at the same time, the GTCNT counter value does not change. The overflow behavior when up-counting using hardware sources is the same as when up-counting by the count clock.

If you are using a hardware source to count up, set the GTCR.CST bit to 1 to enable the counting operation. After GTCR.CST is set to 1, the counter cannot count up for 1 clock cycle as specified in GTCR.TPCS[3:0] because the count operation is synchronized by the count clock selected in GTCR.TPCS[3:0]. Set GTCR.TPCS[3:0] to 000b to count up with a 1 PCLKD delay after GTCR.CST is set to 1.

Figure 21.5 shows an example of an event count operation in up-counting by a hardware resource (the rising edge of GTETRGA pin input).



Note: POEG clock: PCLKB, Core clock: PCLKD
Note 1. The GTETRGA internal signal reflects the signal being input to the GPT when the digital noise filter for the GTETRGA pin in the POEG is not in use.

Figure 21.5 Example of event count operation in up-counting using hardware sources

Table 21.7 shows an example for setting event count operation in up-counting by a hardware source.

Table 21.7 Example for setting an event count operation in up-counting using hardware sources

No.	Step Name	Description
1	Set count source	Select the counting-up hardware source with the GTUPSR register.
2	Set cycle	Set the cycle in the GTPR register.
3	Set initial value for counter	Set the initial value in the GTCNT counter.
4	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

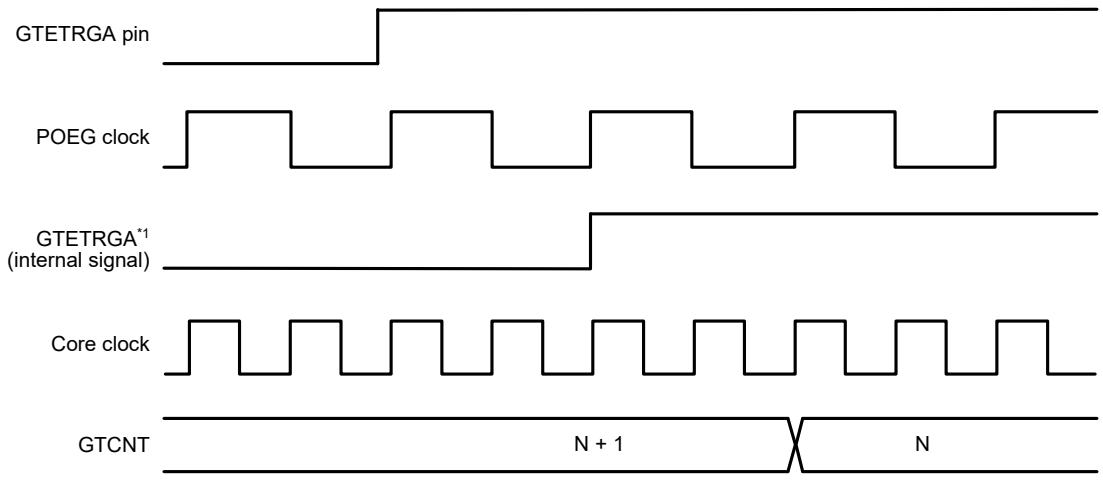
(5) Event count operation in down-counting using hardware sources

The GTCNT counter in each channel can perform down-counting using hardware sources set in the GTDNSR.

When GTDNSR is set to enable, the count clock selected in GTCR.TPCS[3:0] and the count direction selected in GTUDDTYC.UD are ignored. If up-counting and down-counting using hardware sources occur at the same time, the GTCNT counter value does not change. The underflow behavior when down-counting using hardware sources is the same as when down-counting by the count clock.

If you are using a hardware source to count down, set the GTCR.CST bit to 1 to enable the counting operation. After GTCR.CST is set to 1, the counter cannot count down for 1 clock cycle as specified in GTCR.TPCS[3:0] because the count operation is synchronized with the count clock selected in GTCR.TPCS[3:0]. Set GTCR.TPCS[3:0] to 000b to count down with a 1 PCLKD delay after GTCR.CST is set to 1.

Figure 21.6 shows an example of a event count operation in down-counting by a hardware resource (rising edge of GTETRGA pin).



Note: POEG clock: PCLKB, Core clock: PCLKD
Note 1. The GTETRGA internal signal reflects the signal being input to the GPT when the digital noise filter for the GTETRGA pin in the POEG is not in use.

Figure 21.6 Example of event count operation in down-counting using hardware sources

Table 21.8 shows an example for setting a periodic count operation in down-counting using a hardware resource.

Table 21.8 Example for setting an event count operation in down-counting using hardware sources

No.	Step Name	Description
1	Set count source	Select the counting-down hardware source with the GTDNSR register.
2	Set cycle	Set the cycle in the GTPR register.
3	Set initial value for counter	Set the initial value in the GTCNT counter.
4	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

(6) Counter clear operation

The counter of each channel is cleared by following sources:

- Writing 0 to GTCNT register
- Writing 1 to the bit in GTCLR associated with the GPT channel number when the GTCSR.CCLR bit set to 1
- The hardware source selected in GTCSR register.

Writing to the GTCNT register is prohibited during count operation. The GTCNT counter can be cleared both by writing 1 to the GTCLR and by the clear request of hardware sources, whether GTCNT is counting (GTCR.CST is 1) or not (GTCR.CST is 0).

When the count direction flag is set as decrement (GTST.TCUF flag = 0) in saw-wave mode selected with GTCR.MD[2:0] bits, the GTCNT register is set to the value of the GTPR register when writing 1 to the GTCLR register and when clearing by hardware sources are performed.

When not in saw-waves mode and down-counting, the GTCNT register is set to 0 when writing 1 to the GTCLR register and when clearing by hardware sources are performed.

In event count operation when at least 1 bit in the GTUPSR or GTDNSR is set to 1, after clear sources occur, both writing to GTCLR register and clearing by hardware sources are performed immediately to synchronize with PCLKD. If other settings are used, clear is synchronized with the counter clock selected in GTCR.TPCS[3:0].

21.3.1.2 Waveform output by compare match

Compare match means that the GTCNT counter value matches the value of GTCCRA or GTCCRB. When a compare match occurs, the compare match flag is generated synchronously with the count clock, including the event count. At the same time, the GPT can output low, high, or toggled output from the associated GTIOCnA or GTIOCnB output pin (n = 0 to 13). In addition, the GTIOCnA or GTIOCnB pin output can be low, high, or toggled at the cycle end which is determined by GTPR.

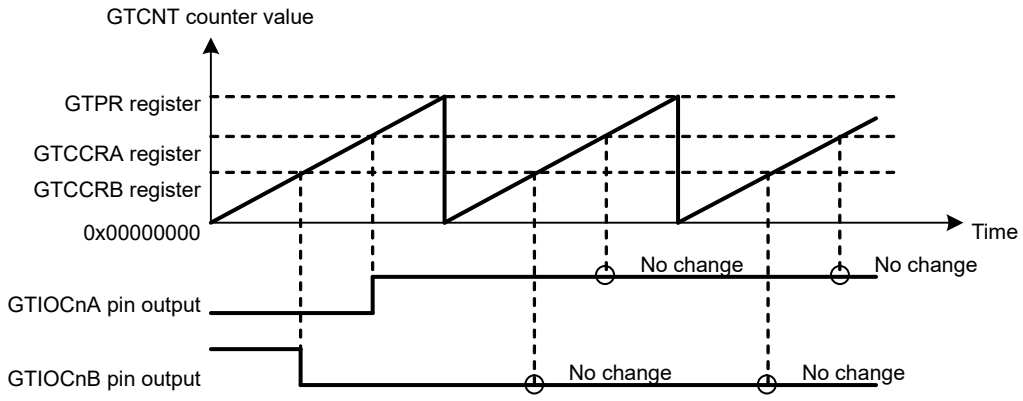
The cycle end is:

- For saw waves in up-counting – when GTCNT changes from the GTPR value to 0 (overflow)
- For saw waves in down-counting – when GTCNT changes from 0 to GTPR value (underflow)
- For saw waves – when the GTCNT counter is cleared
- For triangle waves – when the GTCNT changes from 0 to 1 (trough).

(1) Low output and high output

Figure 21.7 shows an example of low output and high output operation by a compare match of GTCCRA and GTCCRB.

In this example, the GTCNT counter performs up-counting, and settings are made so that high is output from the GTIOCnA pin by a GTCCRA compare match, and low is output from the GTIOCnB pin by a GTCCRB compare match. The pin level does not change when the specified level and pin level match.



[Setting examples]
GTIOR.GTIOA[4:0] bits: Initial output is low, high output at compare match, output retained at cycle end
GTIOR.GTIOB[4:0] bits: Initial output is high, low output at compare match, output retained at cycle end

Note: The base address of registers are the same.

Figure 21.7 Example of low output and high output operation

Table 21.9 shows an example for setting low output and high output operation.

Table 21.9 Example for setting low output and high output operation (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.7, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.7, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.

Table 21.9 Example for setting low output and high output operation (2 of 2)

No.	Step Name	Description
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.7, GTIOA[4:0] = 00010b, GTIOB[4:0] = 10001b.
7	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set compare match value	Set compare match values in the GTCCRA and GTCCRB registers.
9	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

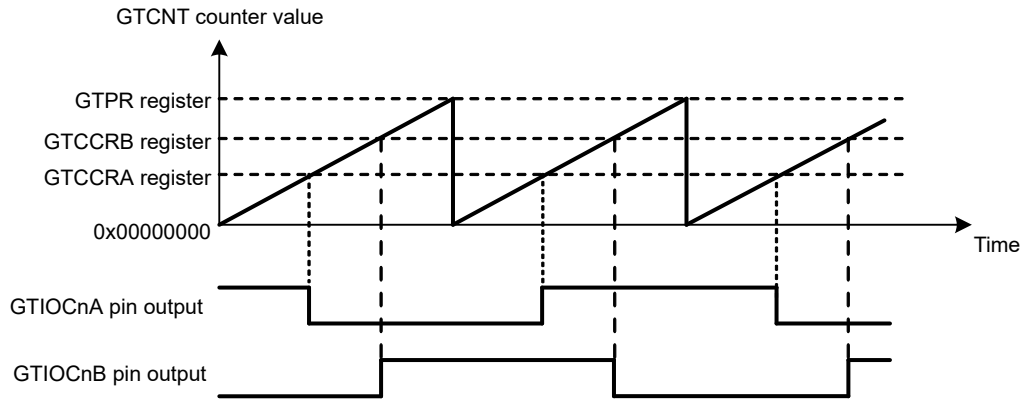
Note: n: 0 to 13
m: A, B

(2) Toggled output

Figure 21.8 and Figure 21.9 show examples of toggled output operation by compare matches of GTCCRA and GTCCRB.

In Figure 21.8, the GTCNT counter performs up-counting, and settings are made so that the GTIOCnA pin output by a GTCCRA compare match and GTIOCnB pin output by a GTCCRB compare match are toggled.

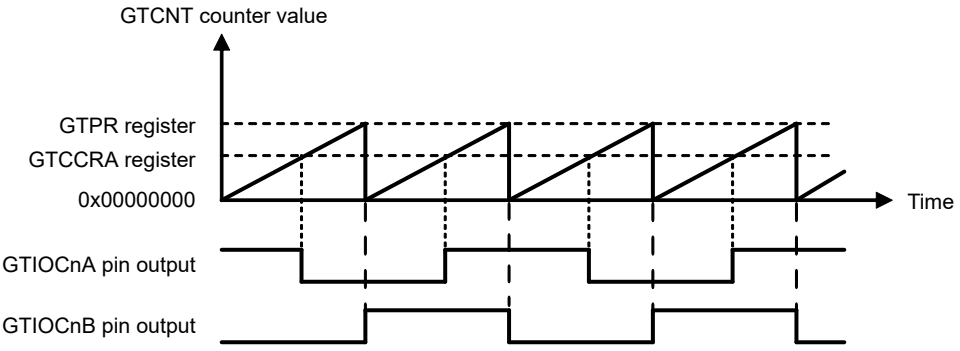
In Figure 21.9, the GTCNT counter performs up-counting, and settings are made so that a GTCCRA compare match toggles the GTIOCnA pin output level and a cycle end toggles the GTIOCnB pin output level.



[Setting examples]
GTIOR.GTIOA[4:0] bits: Initial output is high, output toggled at compare match, output retained at cycle end
GTIOR.GTIOB[4:0] bits: Initial output is low, output toggled at compare match, output retained at cycle end

Note: The base address of registers are the same.

Figure 21.8 Example of toggled output operation (1)



[Setting examples]
GTIOR.GTIOA[4:0] bits: Initial output is high, output toggled at compare match, output retained at cycle end
GTIOR.GTIOB[4:0] bits: Initial output is low, output retained at compare match, output toggled at cycle end

Note: The base address of registers are the same.

Figure 21.9 Example of toggled output operation (2)

Table 21.10 shows an example for setting toggled output operation.

Table 21.10 Example for setting toggled output operation

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.8 and Figure 21.9, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.8 and Figure 21.9, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.8, GTIOA[4:0] = 10011b, GTIOB[4:0] = 00011b, and in Figure 21.9, GTIOA[4:0] = 10011b, GTIOB[4:0] = 01100b.
7	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set compare match value	Set compare match values in the GTCCRA and GTCCRB registers.
9	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

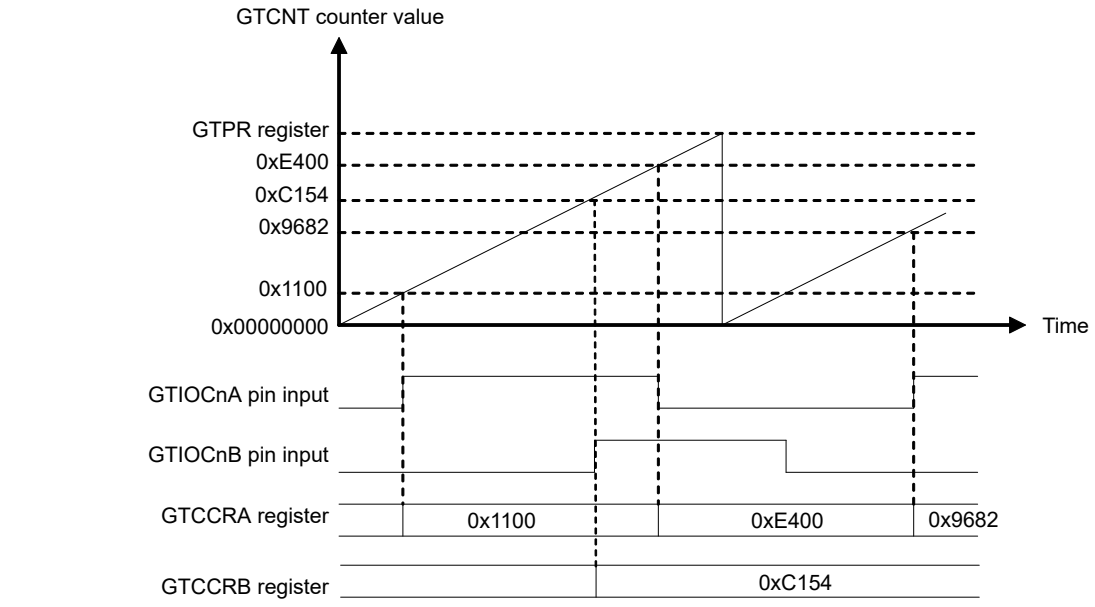
Note: n: 0 to 13
m: A, B

21.3.1.3 Input Capture Function

The GTCNT counter value can be transferred to either GTCCRA or GTCCRB on detection of the hardware source that is set in GTICASR and GTICBSR.

Figure 21.10 shows an example of the input capture function.

In this example, the GTCNT counter performs up-counting by the count clock, and settings are made so that an input capture is performed to GTCCRA at both edges of the GTIOCnA input pin and to GTCCRB on the rising edge of the GTIOCnB input pin.



Note: The base address of registers are the same.
Note: [Setting examples]
GTICASR setting: input capture at both edges
GTICBSR setting: input capture at the rising edge

Figure 21.10 Example of input capture operation

Table 21.11 and Table 21.14 show the example for setting an input capture operation with count operation by the count clock.

Table 21.11 Example for setting input capture operation

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.10, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.10, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Select input capture source	Select the input capture source in the GTICASR and GTICBSR registers. In Figure 21.10, GTICASR = 0x00000F00, GTICBSR = 0x00003000.
7	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

21.3.2 Buffer Operation

The following buffer operations can be set with GTBER:

- GTPR and GTPBR
- GTCCRA, GTCCRC, and GTCCRD
- GTCCRB, GTCCRE, and GTCCRF
- GTADTRA, GTADTBRA, and GTADTDBRA
- GTADTRB, GTADTBRB, and GTADTDBRB

21.3.2.1 GTPR Register Buffer Operation

GTPBR can function as a buffer register for GTPR.

The buffer transfer is performed at an overflow (during up-counting) or an underflow (during down-counting) in saw-wave mode or in event count, and at a trough in triangle-wave mode.

In saw-wave mode or in event count, the buffer transfer is performed when the following counter clear operations occur during counting:

- Clear by hardware sources (the clear source is selected in GTCSR register)
- Clear by software (when GTCSR.CCLR bit is 1 and GTCLR.CCLRn bit is set to 1, n = 0 to 13).

To set GTPR to function as a buffer, set the GTBER.PR bit to 1. To set GTPR not to function as a buffer, set the GTBER.PR bit to 0.

Figure 21.11 and Figure 21.12 show examples of GTPR buffer operation and Table 21.12 shows an example for setting GTPR buffer operation.

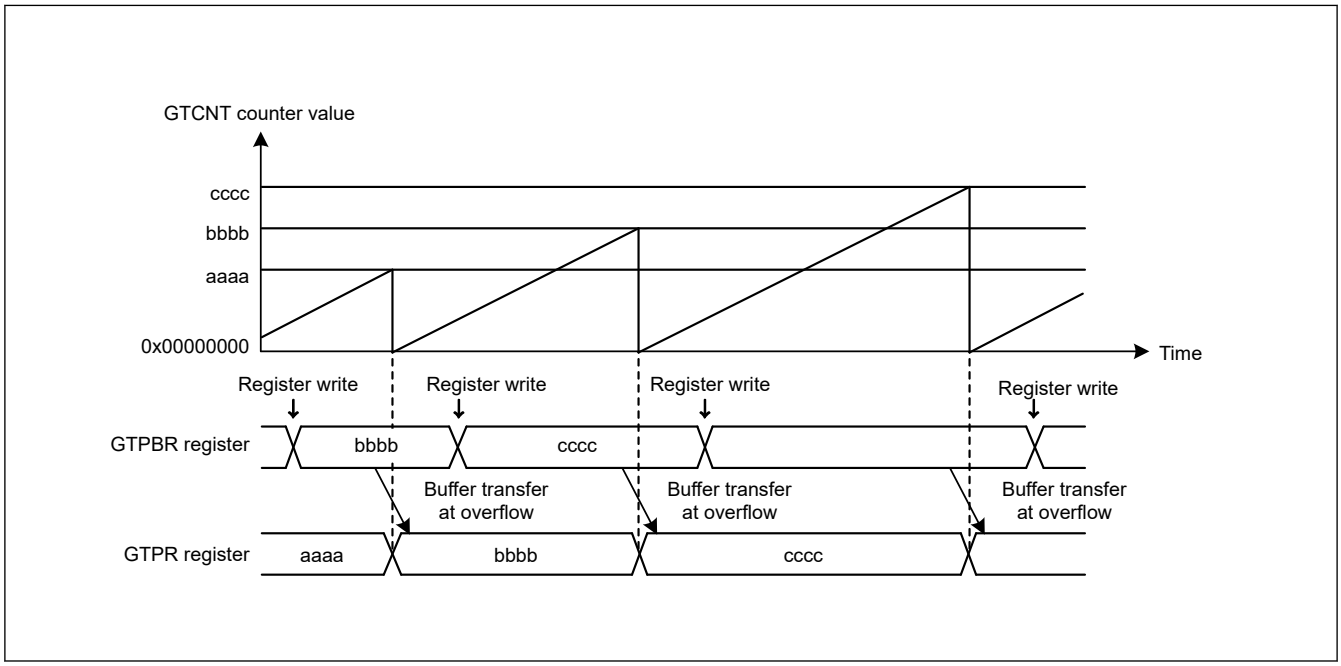


Figure 21.11 Example of GTPR buffer operation with saw waves in up-counting

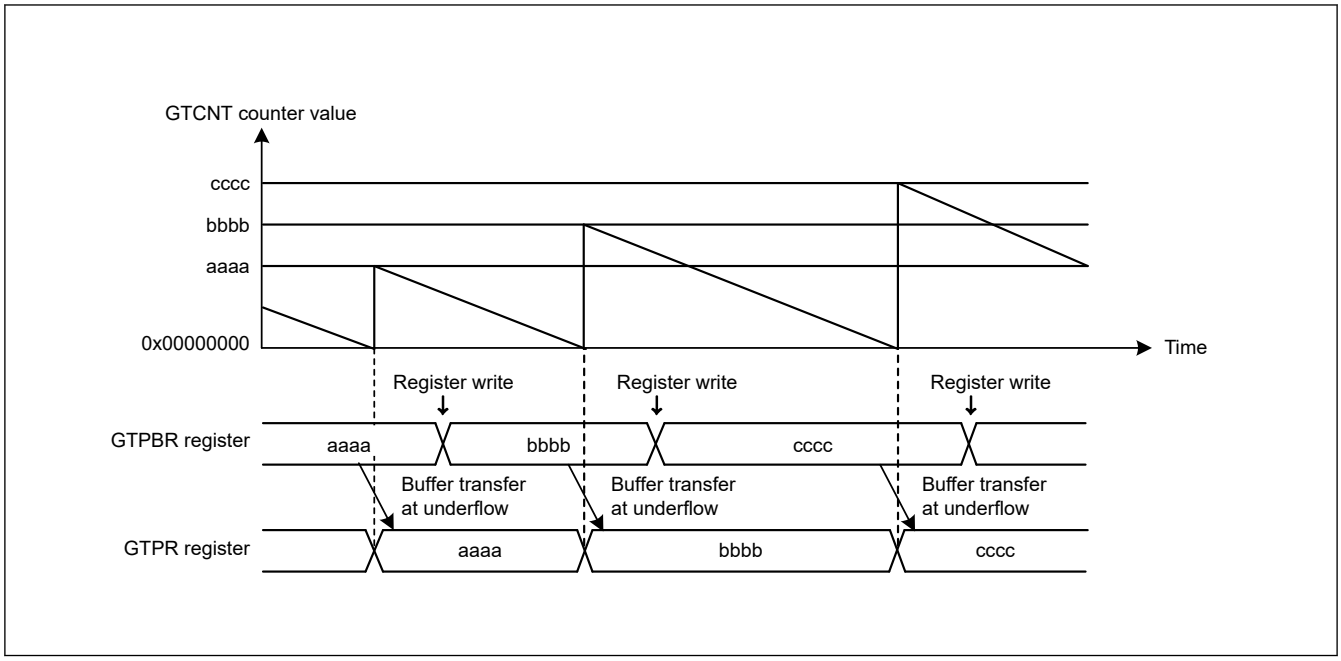


Figure 21.12 Example of GTPR buffer operation with saw waves in down-counting

Table 21.12 Example for setting GTPR register buffer operation

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.11 and Figure 21.12, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.11, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting). In Figure 21.12, after 10b is set in the GTUDDTYC[1:0] bits, 00b is set in the GTUDDTYC[1:0] bits (down-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set buffer operation	Set buffer operation with the GTBER.PR[1:0] bits. In Figure 21.11 and Figure 21.12, PR[1:0] = 01b.
7	Set buffer value	For buffer operation, set a value in one cycle after the current cycle in the GTPBR register.
8	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
9	Set buffer value for each cycle	For buffer operation, set a value in one cycle after the current cycle in the GTPBR register.

21.3.2.2 Buffer Operation for GTCCRA and GTCCRB Registers

GTCCRC can function as the GTCCRA buffer register and GTCCRD can function as the GTCCRC buffer register (double-buffer register for GTCCRA). Similarly, GTCCRE can function as the GTCCRB buffer register and GTCCRF can function as the GTCCRE buffer register (double-buffer register for GTCCRB).

To set GTCCRA or GTCCRB to function as a double buffer, set GTBER.CCRA[1:0] or GTBER.CCRB[1:0] to 10b or 11b. For single buffer operation, set 01b. To set GTCCRA or GTCCRB to not function as a buffer, set 00b.

In saw-wave one-shot pulse mode, triangle-wave PWM mode 3, the buffer operations that specific each PWM output operation mode are performed regardless of the setting of GTBER.CCRA [1:0] bits and GTBER.CCRB [1:0] bits.

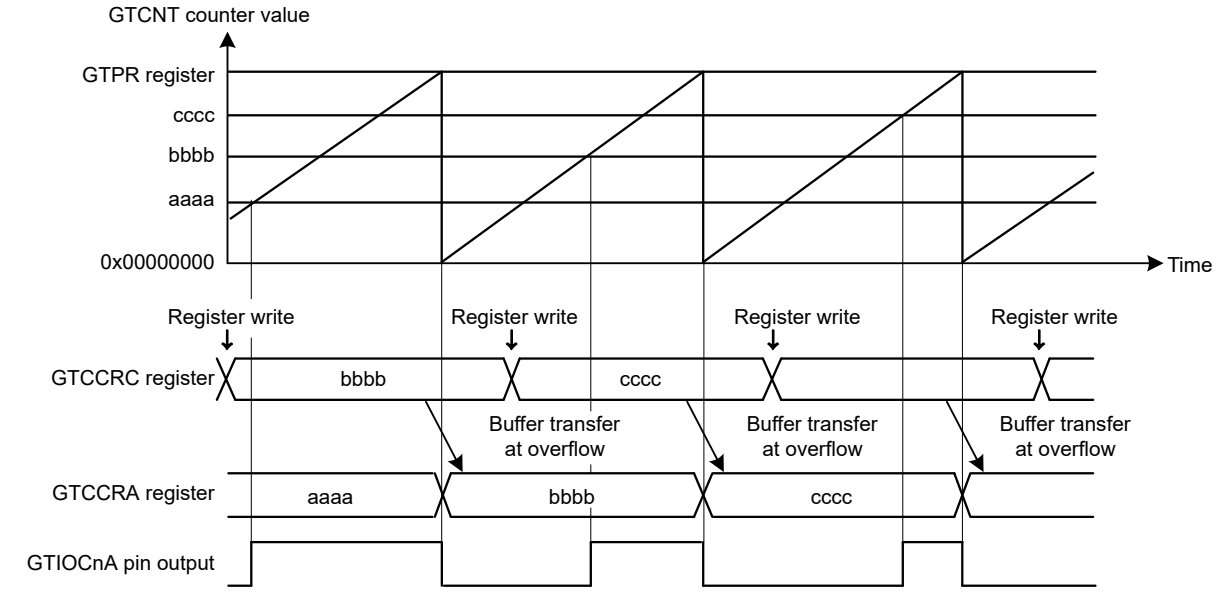
(1) When GTCCRA or GTCCRB Functions as Output Compare Register

In saw-wave one-shot pulse mode, triangle-wave PWM mode 3, the buffer operations that specific each PWM output operation mode are performed regardless of the setting of GTBER.CCRA [1:0] bits and GTBER.CCRB [1:0] bits. For details, see section 21.3.3. PWM Output Operating Mode. Other than the specified PWM output operation mode, buffer transfer occurs in the following situations:

Buffer transfer occurs in the following situations:

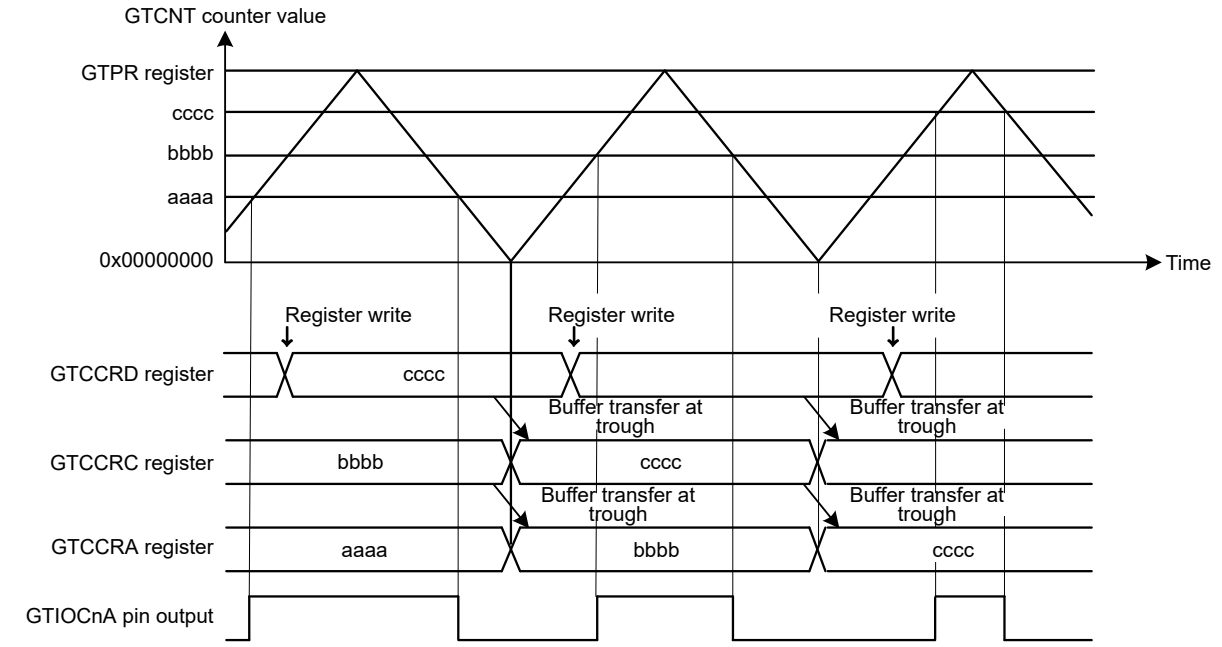
- Buffer transfer by overflow or underflow
  Buffer transfer is performed at an overflow (during up-counting) or an underflow (during down-counting) in saw-wave mode or in event count operation. In triangle-wave mode, buffer transfer is performed at a trough (triangle-wave PWM mode 1) or a crest and trough (triangle-wave PWM mode 2).
- Buffer transfer by counter clear
  In saw-wave mode or in event count operation, during counting, buffer transfer (which is the same as an overflow during up-counting or an underflow during down-counting) is performed by the counter clear sources similar to the case shown in section 21.3.2.1. GTPR Register Buffer Operation.
  In triangle-wave mode, buffer transfer is not performed by the counter clear.
- Forcible buffer transfer
  When GTBER.CCRSWT bit is set to 1 while the count operation is stopped, the GTCCRA and the GTCCRB register buffer transfer are performed forcibly in saw-wave mode, in event count operation and in triangle-wave mode. Additionally buffer transfer from the GTCCRD register to temporary register A and from the GTCCRF register to temporary register B are performed in saw-wave one-shot pulse mode or triangle-wave PWM mode 3.

Figure 21.13 to Figure 21.15 show examples of GTCCRA and GTCCRB buffer operation and Table 21.13 shows an example for setting GTCCRA and GTCCRB buffer operation.



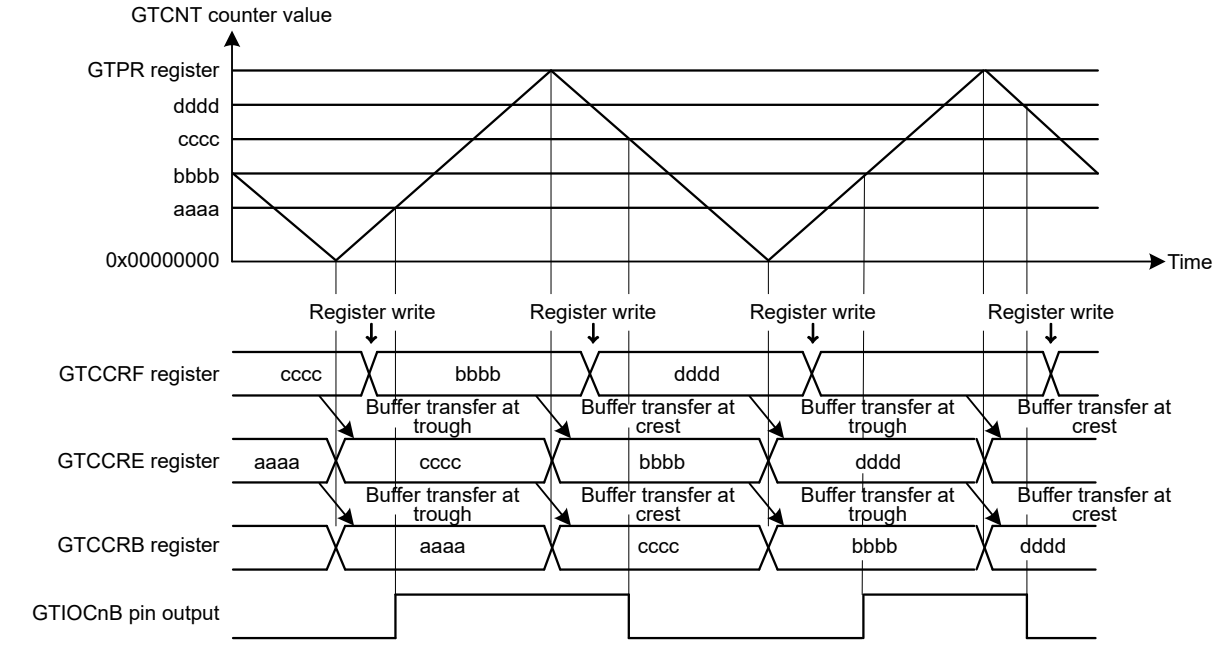
Note: The base address of registers are the same.

Figure 21.13 Example of GTCCRA buffer operation with output compare, saw waves in up-counting, high output at GTCCRA compare match, and low output at cycle end



Note: The base address of registers are the same.

Figure 21.14 Example of GTCCRA double buffer operation with output compare, triangle waves, buffer operation at trough, output toggled at GTCCRA compare match, and output retained at cycle end



Note: The base address of registers are the same.

Figure 21.15 Example of GTCCRB double buffer operation with output compare, triangle waves, buffer operation at both troughs and crests, output toggled at GTCCRB compare match, and output retained at cycle end

Table 21.13 Example for setting GTCCRA and GTCCRB buffer operation for output compare

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.13, 000b (saw-wave PWM mode) is set, in Figure 21.14, 100b (triangle-wave PWM mode 1) is set, and in Figure 21.15, 101b (triangle-wave PWM mode 2) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.13, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.13, GTIOA[4:0] = 00110b, in Figure 21.14, GTIOA[4:0] = 00011b, and in Figure 21.15, GTIOB[4:0] = 00011b.
7	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTBER register. In Figure 21.13, CCRA[1:0] = 01b, in Figure 21.14, CCRA[1:0] = 1xb, and in Figure 21.15, CCRB [1:0] = 1xb.
9	Set compare match value	Set the GTIOCnA pin transition in the GTCCRA register and the GTIOCnB pin transition in the GTCCRB register.
10	Set buffer value	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRD and GTCCRF registers, respectively.
11	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
12	Set buffer value for each cycle	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRD and GTCCRF registers, respectively.

Note: n: 0 to 13
m: A, B

(2) When GTCCRA or GTCCRB Functions as Input Capture Register

When an input capture is generated, the GTCNT counter value is transferred to GTCCRA and GTCCRB and the stored GTCCRA and GTCCRB register values are transferred to the buffer registers. In input capture operation, the buffer transfer is not performed by the counter clear.

Figure 21.16 and Figure 21.17 show examples of GTCCRA and GTCCRB buffer operation and Table 21.14 shows an example for setting GTCCRA and GTCCRB buffer operation.

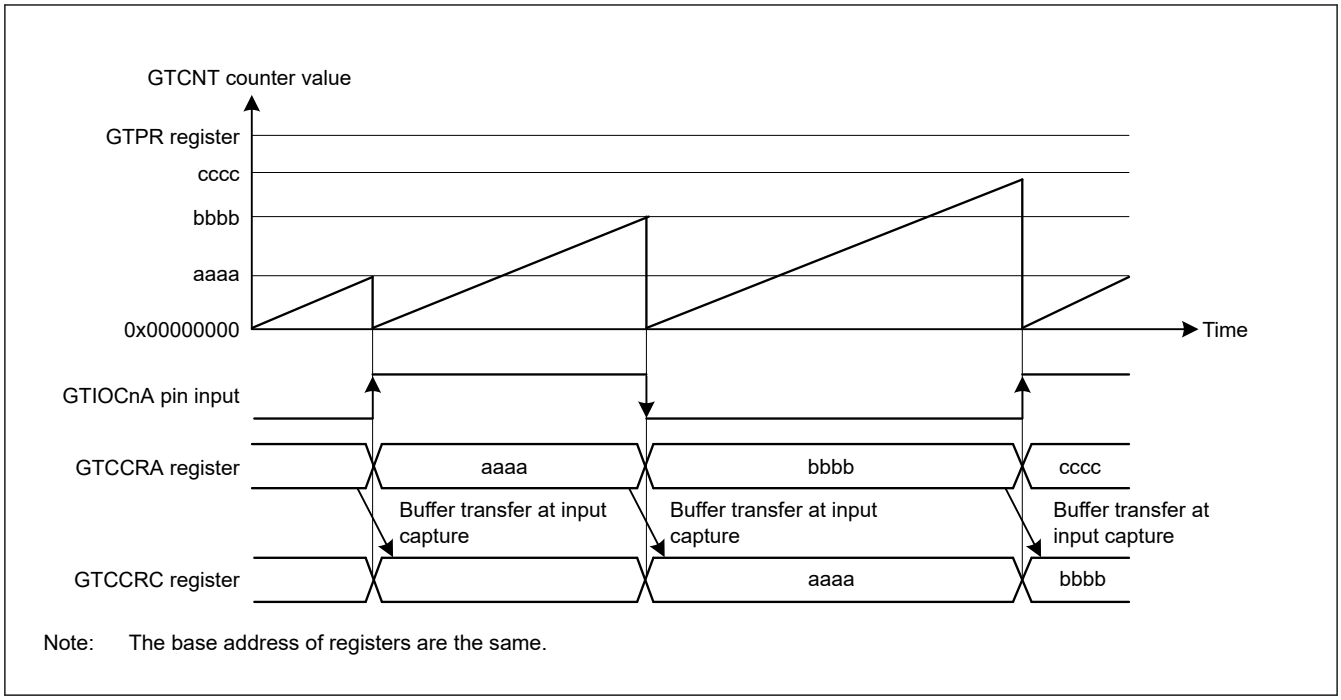


Figure 21.16 Example of GTCCRA buffer operation with input capture at both edges of GTIOCnA input, saw waves in up-counting, and GTCNT counter cleared at both edges of GTIOCnA input

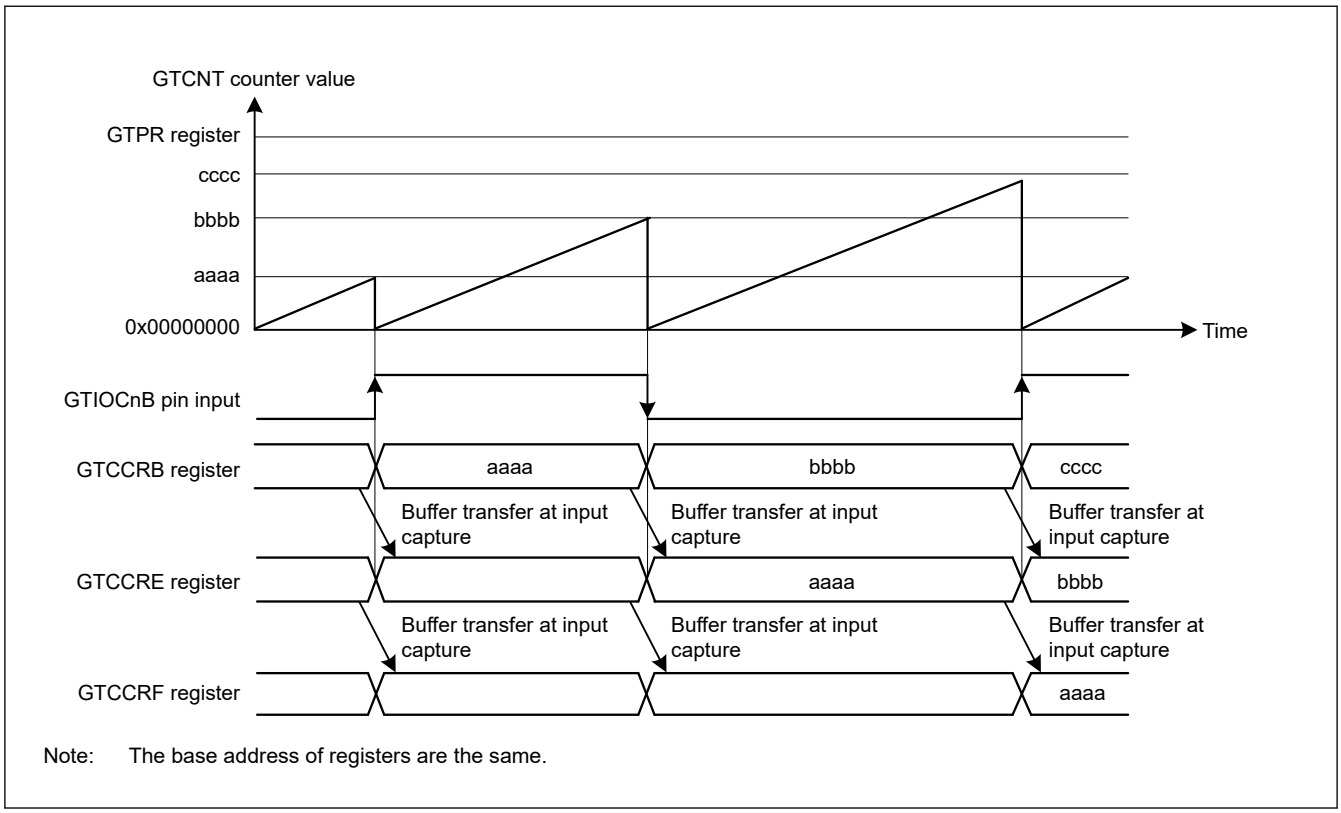


Figure 21.17 Example of GTCCRB double buffer operation with input capture at both edges of GTIOCnB input, saw waves in up-counting, and GTCNT counter cleared at both edges of GTIOCnB input

Table 21.14 Example for setting GTCCRA and GTCCRB buffer operation for input capture

No.	Step Name	Description
1	Set operating mode and counter clear sources	Set the operating mode with the GTCR.MD[2:0] bits and count clear source with the GTCSR register. In Figure 21.16, MD[2:0] = 000b (saw-wave PWM mode) and GTCSR = 0x00000F00, and in Figure 21.17, MD[2:0] = 000b (saw-wave PWM mode) and GTCSR = 0x0000F000.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.16, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Select input capture source	Select input capture source in the GTICASR register and GTICBSR register. In Figure 21.16, GTICASR = 0x00000F00, and in Figure 21.17, GTICBSR = 0x0000F000.
7	Set buffer operation	Set buffer operation with the CCRA and CCRB bits in the GTBER register. In Figure 21.16, CCRA[1:0] = 01b, and in Figure 21.17, CCRB[1:0] = 1xb.
8	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

21.3.2.3 Buffer Operation for GTADTRA and GTADTRB Registers

The GTADTBRA register can function as the GTADTRA buffer register and the GTADTDBRA register can function as the GTADTBRA buffer register (double buffer register for the GTADTRA register). Similarly, the GTADTBRB register can function as the GTADTRB buffer register and the GTADTDBRB register can function as the GTADTBRB buffer register (double buffer register for the GTADTRB register).

To set the GTADTRA or GTADTRB register to function as a double buffer, set the GTBER.ADTDA or ADTDB bit to 1. For single buffer operation, set 0. Not to function as buffer, set the GTBER.ADTTA[1:0] or ADTTB[1:0] bits to 00b.

The buffer transfer timing can be set with the ADTTA[1:0] and ADTTB[1:0] bits to an overflow (in up-counting) or an underflow (in down-counting) in saw-wave mode, with ADTTA[1:0] and ADTTB[1:0] bits to 01b for a crest, to 10b for a trough, or to 11b for both crest and trough in triangle-wave mode.

In saw-wave mode, when the ADTTA[1:0] and ADTTB[1:0] bits are set to value other than 00b and in count operation, the buffer transfer, by similar counter clearing sources in section 21.3.2.1. GTPR Register Buffer Operation, is performed in the same way at an overflow (in up-counting) or an underflow (in down-counting).

Figure 21.18 to Figure 21.20 show examples of buffer operation of the GTADTRA and GTADTRB registers, and Table 21.15 shows an example for setting buffer operation of the GTADTRA and GTADTRB registers.

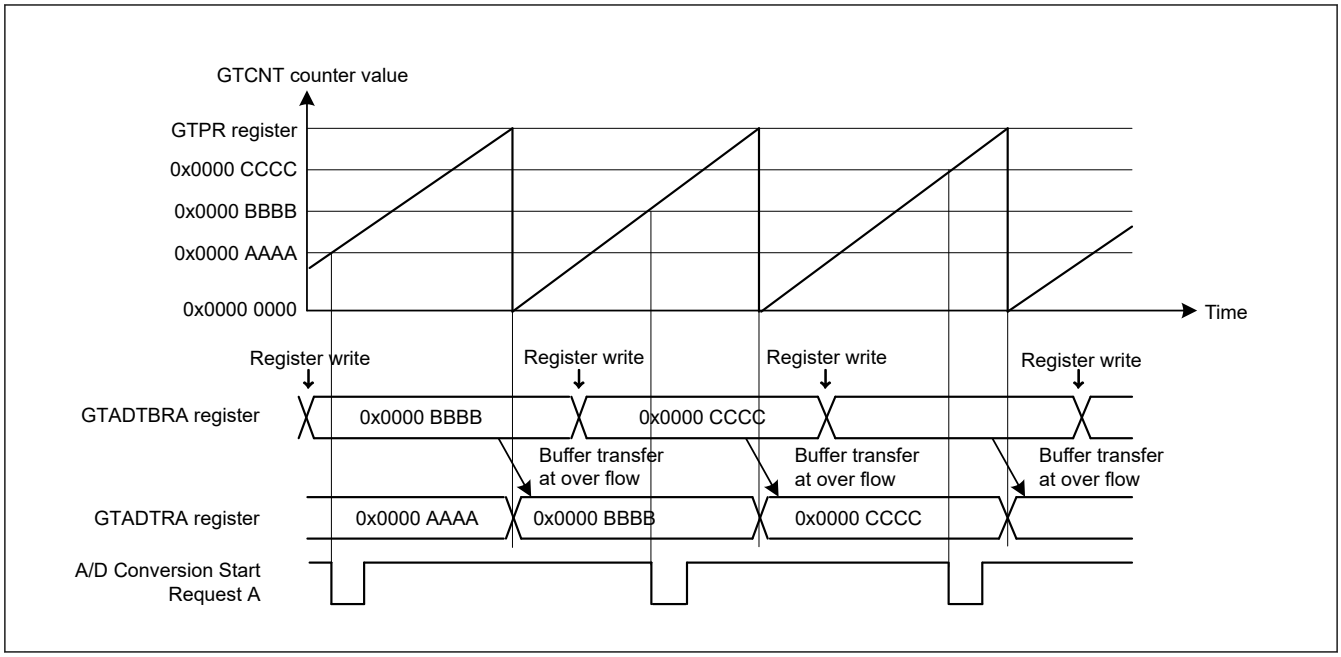


Figure 21.18 Example of buffer operation of the GTADTRA registers (saw waves in up-counting, A/D conversion start request generated by up-counting)

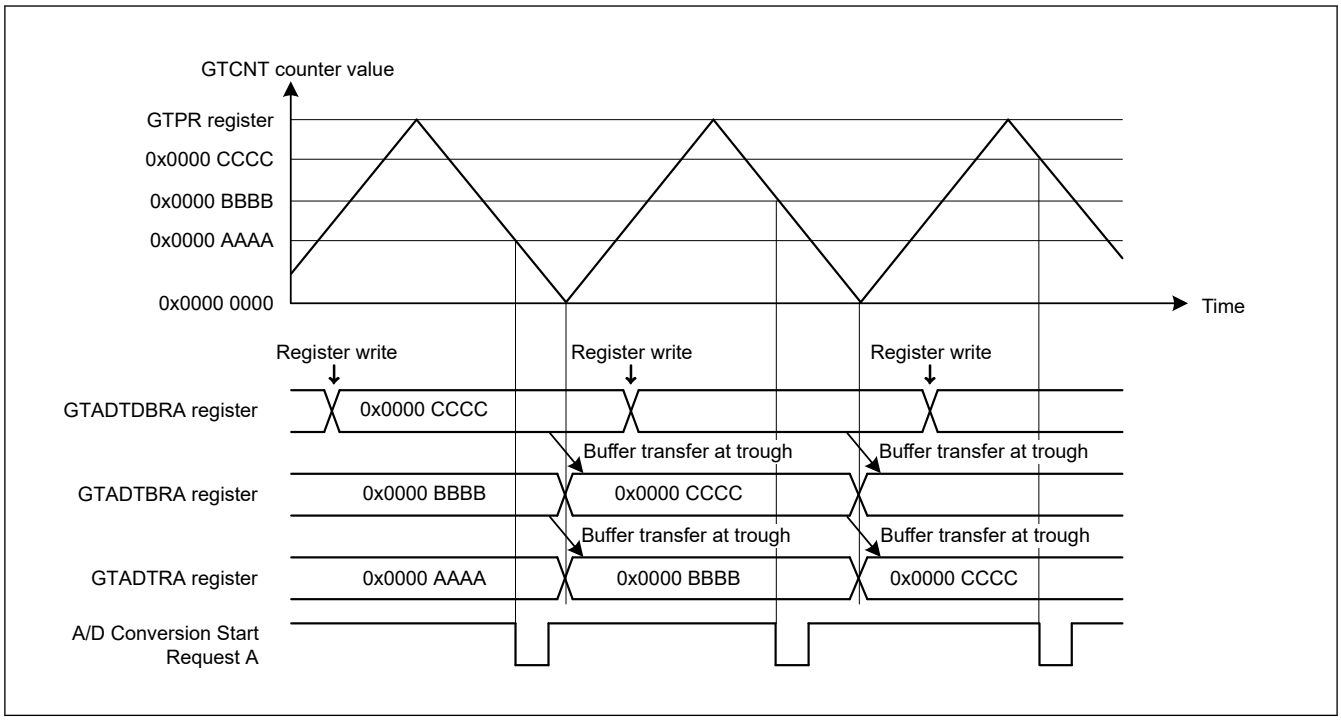


Figure 21.19 Example of double buffer operation of the GTADTRA registers (triangle waves, buffer transfer at troughs, A/D conversion start request generated by down-counting)

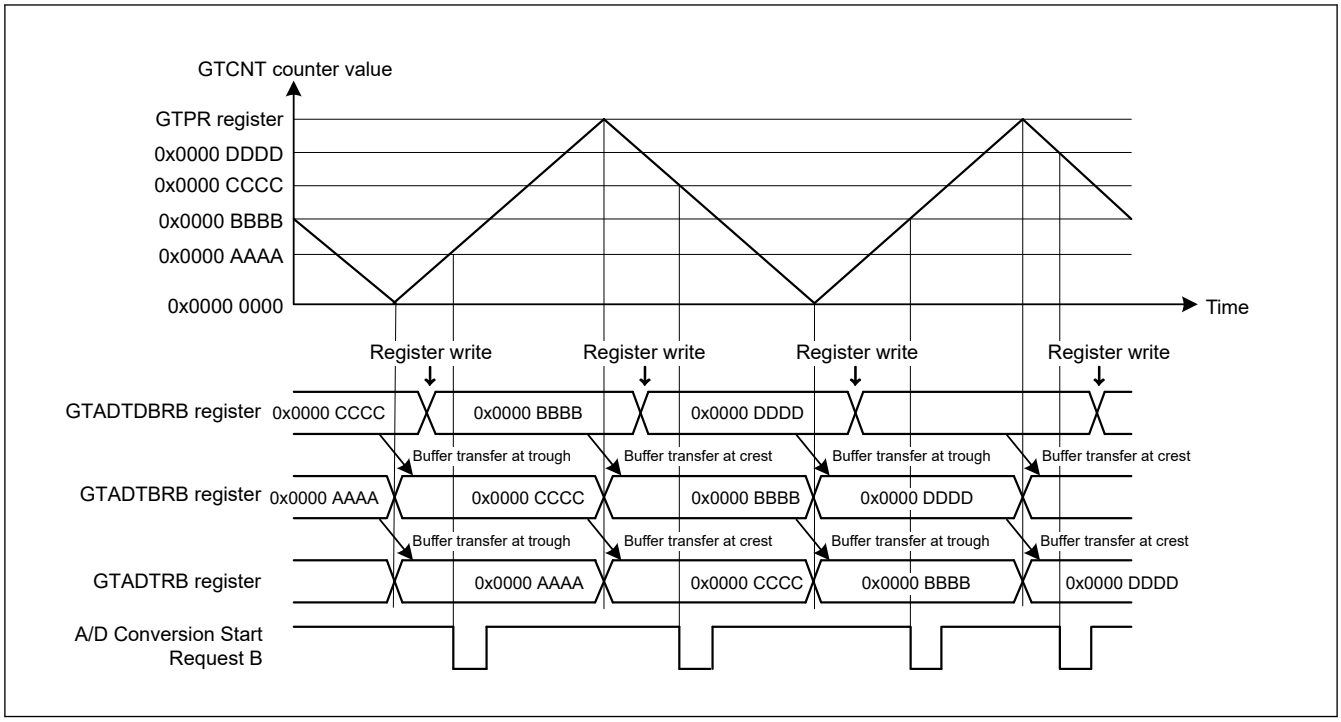


Figure 21.20 Example of double buffer operation of the GTADTRB registers (triangle waves, buffer transfer at both troughs and crests, A/D conversion start request generated by both up- and down-counting)

Table 21.15 Example for Setting Buffer Operation of the GTADTRA and GTADTRB registers (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. (In Figure 21.18 000b (saw-wave PWM mode) is set, in Figure 21.19 and Figure 21.20, 100b, 101b, 110b (triangle-wave PWM mode) is set.)
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. (In Figure 21.18, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).)
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set buffer operation	Set buffer operation with the ADTTA[1:0], ADTTB[1:0], ADTDA, and ADTDB bits in GTBER register. (In Figure 21.18, ADTTA[1:0] bits = 01b, 10b, or 11b and ADTDA bit = 0, in Figure 21.19, ADTTA [1:0] bits = 10b and ADTDA bit = 1, and in Figure 21.20, ADTTB[1:0] bits = 11b and ADTDB bit = 1.)
7	Set compare match value	Set the A/D conversion start request point in the GTADTRA and GTADTRB registers.
8	Set buffer value	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.
9	Enable A/D conversion start request	Set to enable A/D conversion start request with the ADTRAUEN, ADTRADEN, ADTRBUEN, and ADTRBDEN bits in the GTINTAD register. (In Figure 21.18, ADTRAUEN bit = 1, in Figure 21.19, ADTRADEN bit = 1, and in Figure 21.20, ADTRBUEN bit = 1 and ADTRBDEN bit = 1.)
10	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

Table 21.15 Example for Setting Buffer Operation of the GTADTRA and GTADTRB registers (2 of 2)

No.	Step Name	Description
11	Set buffer value of each cycle	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.

21.3.3 PWM Output Operating Mode

The GPT can output PWM waveforms to the GTIOCnA or GTIOCnB pin (n = 0 to 13) by a compare match between the GTCNT counter and GTCCRA or GTCCRB.

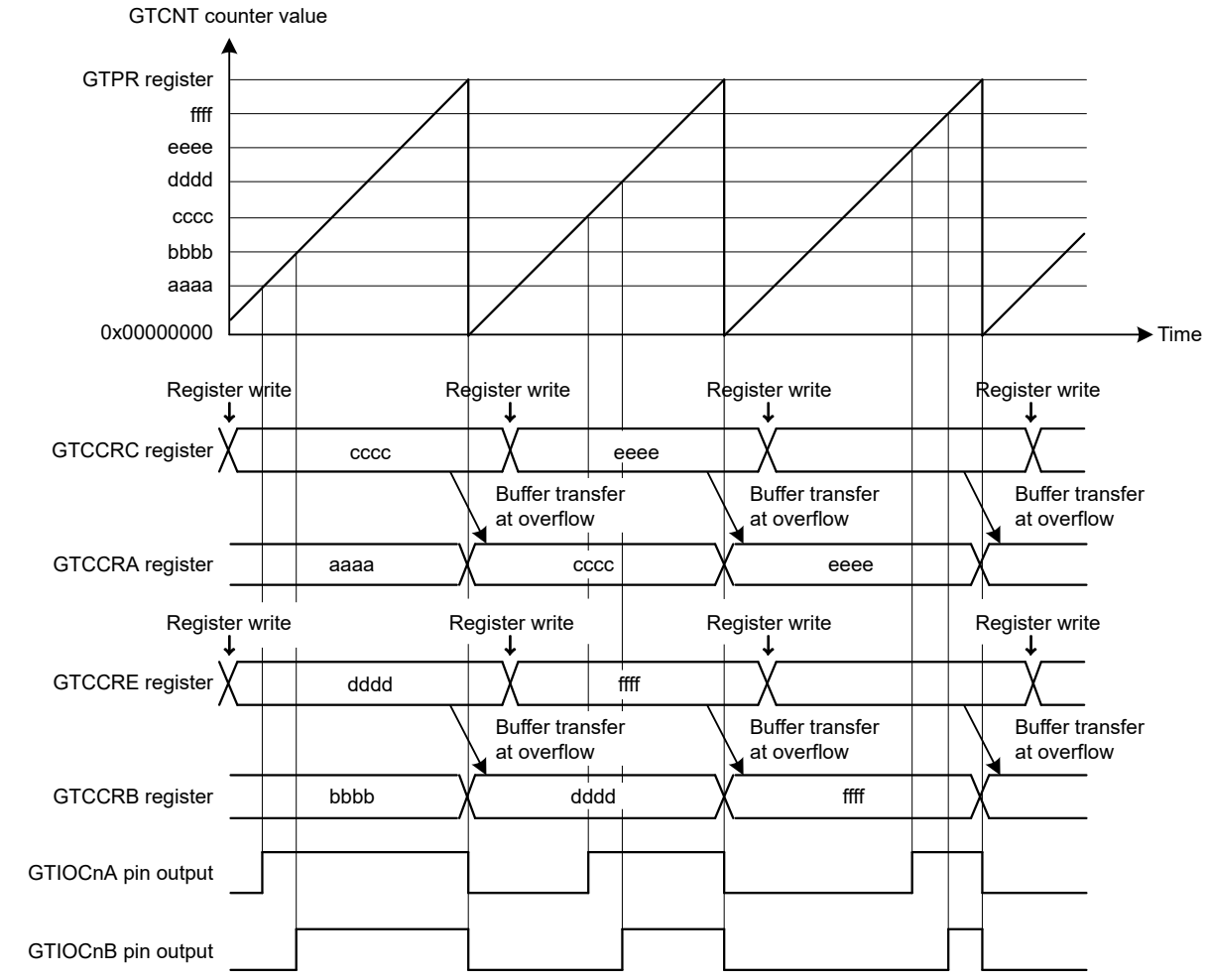
By setting GTDTCR and GTDVU, the compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

21.3.3.1 Saw-Wave PWM Mode

In saw-wave PWM mode, GTCNT performs saw-wave (half-wave) operation by setting the cycle in GTPR and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 13) when a GTCCRA or GTCCRB compare match occurs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the cycle end according to the GTIOR setting.

When the timing of the end of cycle and the timing of GTCCRk register compare match are the same time, the output pin performs along the PWM output setting for the end of cycle set by the GTIOR.GTIOx[3:2] bits (x = A, B).

Figure 21.21 shows an example of saw-wave PWM mode operation, and Table 21.16 shows an example for setting saw-wave PWM mode.



Note: The base address of registers are the same.

Figure 21.21 Example of saw-wave PWM mode operation with up-counting, buffer operation, high output at GTCCRA/GTCCRB compare match, and low output at cycle end

Table 21.16 Example for setting saw-wave PWM mode (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.21, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.21, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.21, GTIOA[4:0] = 00110b and GTIOB[4:0] = 00110b.
7	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTBER register. In Figure 21.21, CCRA[1:0] = 01b and CCRB[1:0] = 01b.
9	Set compare match value	Set the GTIOCnA pin transition in the GTCCRA register and the GTIOCnB pin transition in the GTCCRB register.

Table 21.16 Example for setting saw-wave PWM mode (2 of 2)

No.	Step Name	Description
10	Set buffer value	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.
11	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
12	Set buffer value for each cycle	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.

Note: n: 0 to 13
m: A, B

21.3.3.2 Saw-Wave One-Shot Pulse Mode

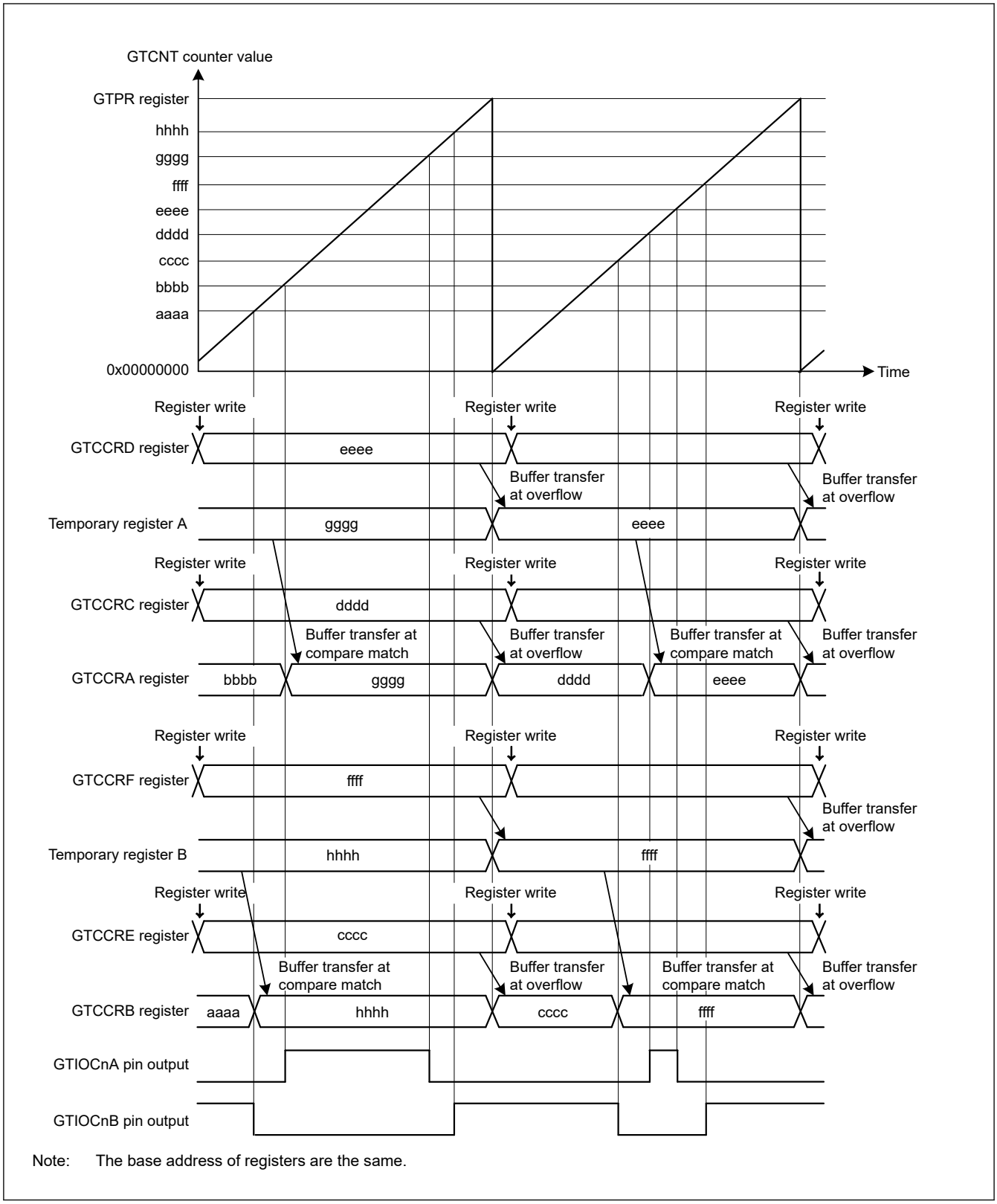
The saw-wave one-shot pulse mode is a mode in which the cycle is set in GTPR, the GTCNT counter performs saw-wave (half-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 13) at a compare match of GTCCRA or GTCCRB with buffer operation fixed.

Buffer operation in saw-wave one-shot pulse mode is different from the usual buffer operation. Buffer transfer is performed from:

- GTCCRC to GTCCRA at the cycle end
- GTCCRE to GTCCRB at the cycle end
- GTCCRD to temporary register A at the cycle end
- GTCCRF to temporary register B at the cycle end
- Temporary register A to GTCCRA at a GTCCRA compare match
- Temporary register B to GTCCRB at a GTCCRB compare match.

The pin output value can be selected from low output, high output, or toggled output separately for a compare match and the cycle end according to the GTIOR setting. When the GTBER.CCRSWT bit is set to 1 while count operation is stopped, the buffer is transferred forcibly from the GTCCRD register to temporary register A and from the GTCCRF register to temporary register B. By setting GTDTCR and GTDVU, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 21.22 shows an example of saw-wave one-shot pulse mode operation, and Table 21.17 shows an example for setting saw-wave one-shot pulse mode.



Note: The base address of registers are the same.

Figure 21.22 Example of saw-wave one-shot pulse mode operation with up-counting, low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/ GTCCRB compare match, and output retained at cycle end

Table 21.17 Example setting for saw-wave one-shot pulse mode

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.22, 001b (saw-wave one-shot pulse mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.22, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.22, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
7	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set compare match value	Set the GTIOCnA pin transition immediately after the count start in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.
9	Set forcible buffer transfer	Set the GTBER.CCRSWT bit to 1 to transfer buffer register data forcibly.
10	Set buffer value	For buffer operation, set the GTIOCnA pin transition in one cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.
11	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
12	Set buffer value for each cycle	For buffer operation, set the GTIOCnA pin transition in one cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.

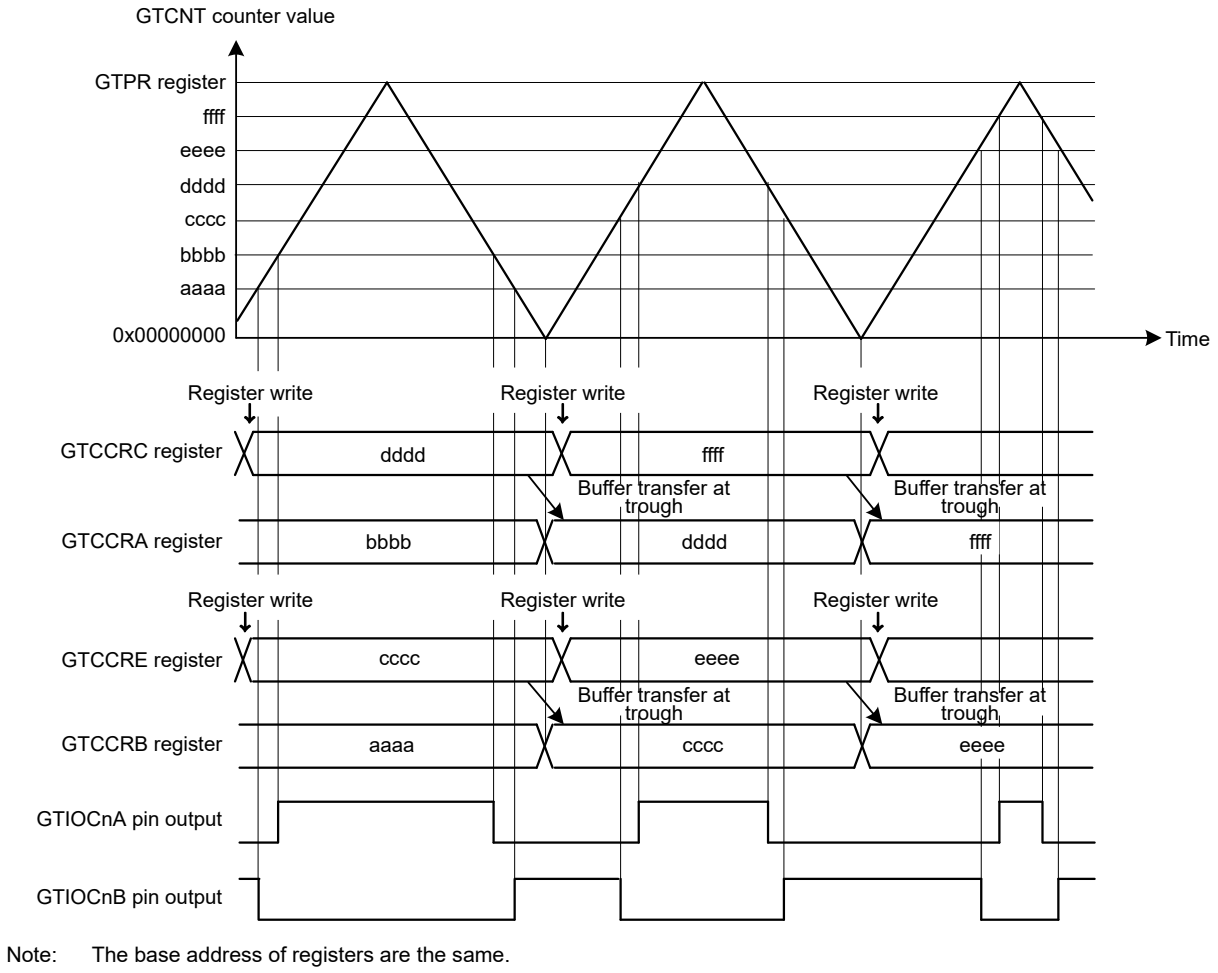
Note: n: 0 to 13
m: A, B

21.3.3.3 Triangle-Wave PWM Mode 1 (32-Bit Transfer at Trough)

The triangle-wave PWM mode 1 is a mode in which the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 13) when a GTCCRA or GTCCRB compare match occurs. Buffer transfer is performed at the trough. The pin output value can be selected from low output, high output, or toggled output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR and GTDVU, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 21.23 shows an example of a triangle-wave PWM mode 1 operation, and Table 21.18 shows an example for setting a triangle-wave PWM mode 1.



Note: The base address of registers are the same.

Figure 21.23 Example of triangle-wave PWM mode 1 operation with buffer operation, low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/ GTCCRB register compare match, and output retained at cycle end

Table 21.18 Example setting for triangle-wave PWM mode 1 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.23, 100b (triangle-wave PWM mode 1) is set.
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.23, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
7	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTBER register. In Figure 21.23, CCRA[1:0] = 01b and CCRB[1:0] = 01b.
8	Set compare match value	Set the GTIOCnA and GTIOCnB pins transitions in the GTCCRA and GTCCRB registers, respectively.
9	Set buffer value	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.

Table 21.18 Example setting for triangle-wave PWM mode 1 (2 of 2)

No.	Step Name	Description
10	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
11	Set buffer value for each cycle	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.

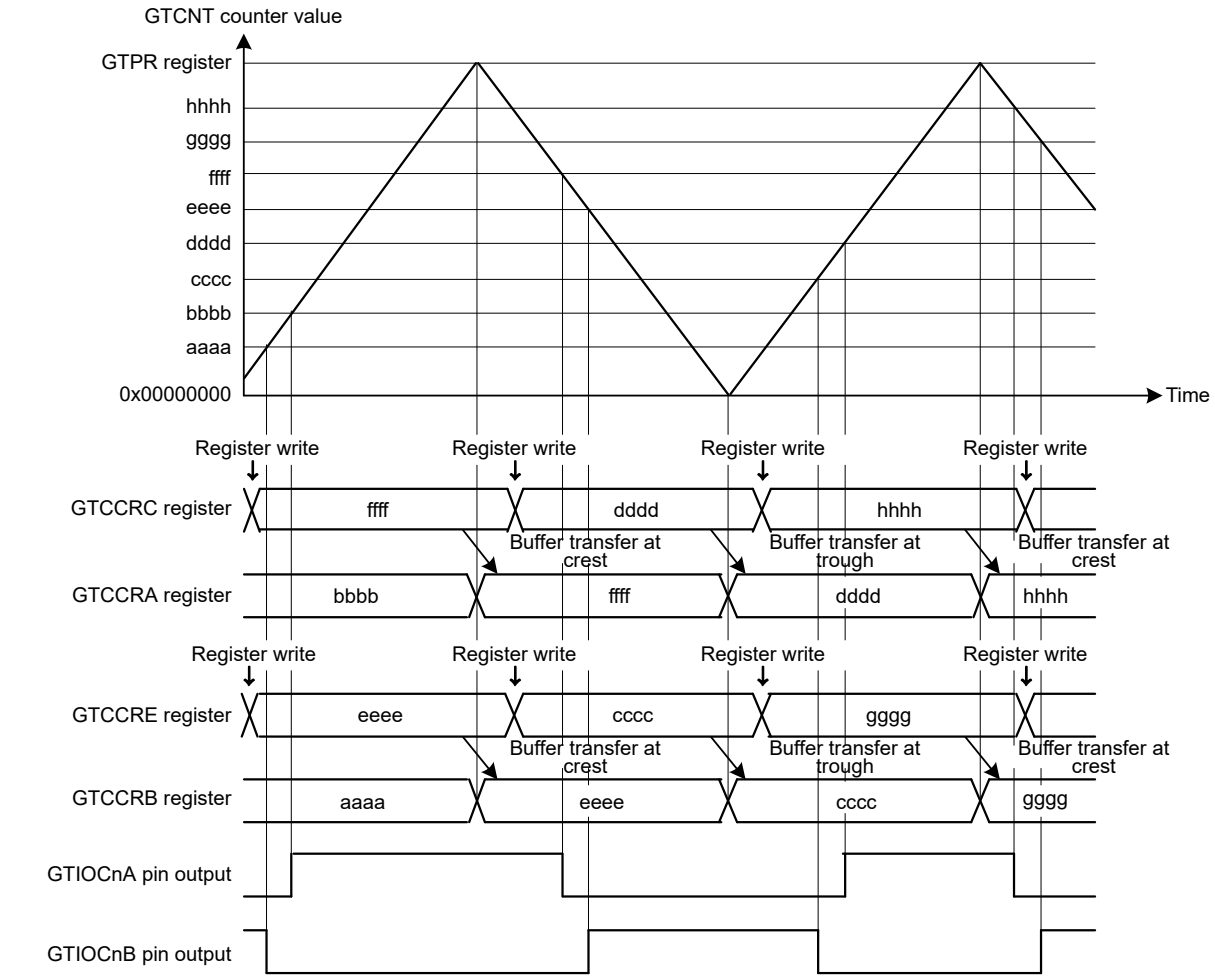
Note: n: 0 to 13
m: A, B

21.3.3.4 Triangle-Wave PWM Mode 2 (32-Bit Transfer at Crest and Trough)

Similarly to triangle-wave PWM mode 1, in triangle-wave PWM mode 2 the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 13) when a GTCCRA or GTCCRB compare match occurs. The buffer transfer is performed at both crests and troughs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR and GTDVU, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 21.24 shows an example of triangle-wave PWM mode 2 operation, and Table 21.19 shows an example for setting triangle-wave PWM mode 2.



Note: The base address of registers are the same.

Figure 21.24 Example of triangle-wave PWM mode 2 operation with buffer operation, low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/ GTCCRB compare match, and output retained at cycle end

Table 21.19 Example for setting triangle-wave PWM mode 2 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.24, 101b (triangle-wave PWM mode 2) is set.
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.24, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
7	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTBER register. In Figure 21.24, CCRA[1:0] = 01b and CCRB[1:0] = 01b.
8	Set compare match value	Set the GTIOCnA and GTIOCnB pins transitions in the GTCCRA and GTCCRB registers, respectively.

Table 21.19 Example for setting triangle-wave PWM mode 2 (2 of 2)

No.	Step Name	Description
9	Set buffer value	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in half cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRD and GTCCRF registers, respectively.
10	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
11	Set buffer value for each half cycle	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in half cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in GTCCRD and GTCCRF registers, respectively.

Note: n: 0 to 13
m: A, B

21.3.3.5 Triangle-Wave PWM Mode 3 (64-Bit Transfer at Trough)

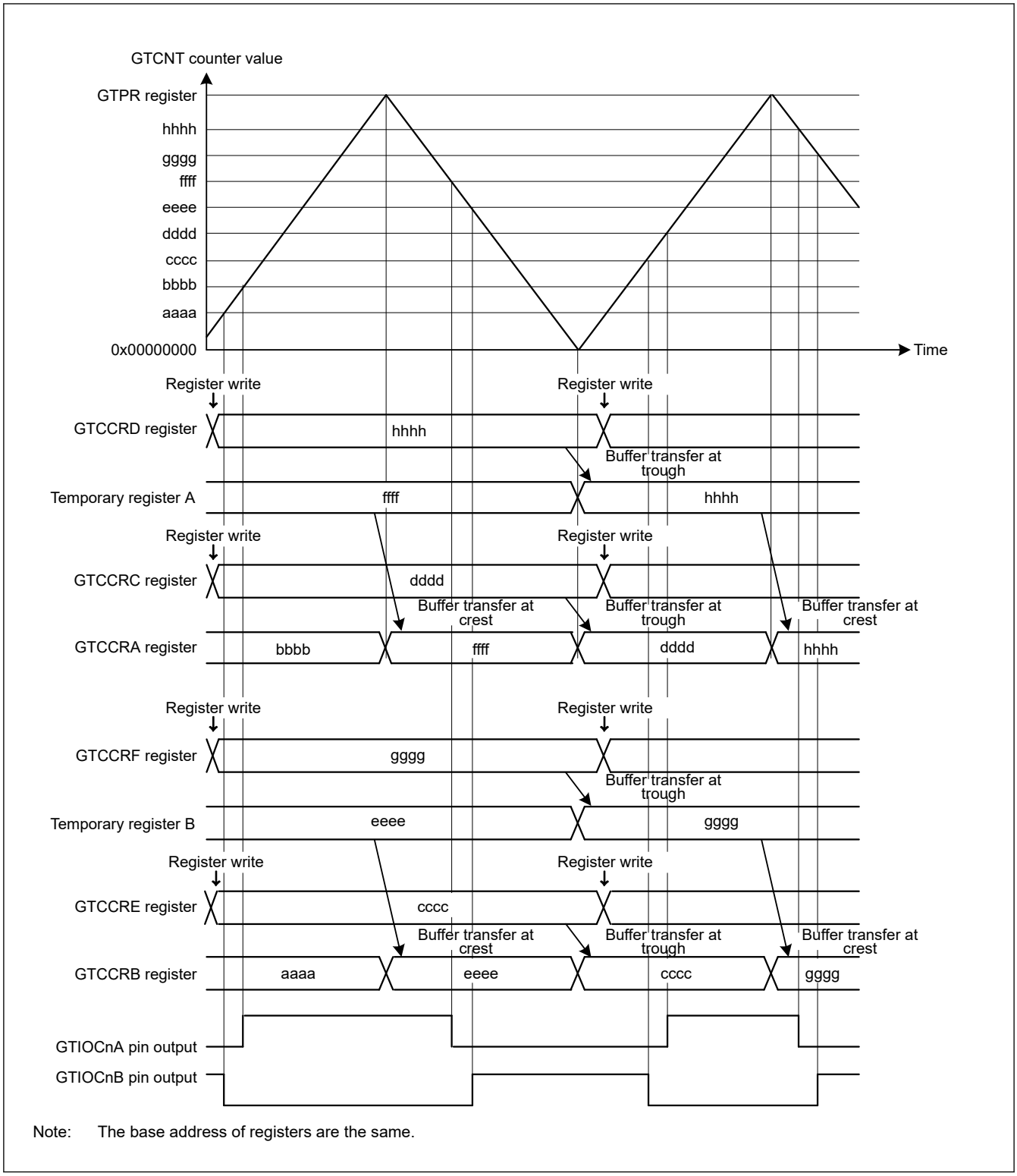
The triangle-wave PWM mode 3 is a mode in which the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 13) at a compare match of GTCCRA or GTCCRB with buffer operation fixed. Buffer operation in triangle-wave PWM mode 3 is different from the usual buffer operation. Buffer transfer is performed from the following:

- GTCCRC to GTCCRA at the trough
- GTCCRE to GTCCRB at the trough
- GTCCRD to temporary register A at the trough
- GTCCRF to temporary register B at the trough
- Temporary register A to GTCCRA at the crest
- Temporary register B to GTCCRB at the crest.

The pin output value can be selected from low output, high output, or toggled output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR and GTDVU, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 21.25 shows an example of triangle-wave PWM mode 3 operation, and Table 21.20 shows an example for setting triangle-wave PWM mode 3.



Note: The base address of registers are the same.

Figure 21.25 Example of triangle-wave PWM mode 3 operation with low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/GTCCRB compare match, and output retained at cycle end

Table 21.20 Example setting for triangle-wave PWM mode 3 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.25, 110b (triangle-wave PWM mode 3) is set.
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.

Table 21.20 Example setting for triangle-wave PWM mode 3 (2 of 2)

No.	Step Name	Description
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.25, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
7	Set compare match value	Set the GTIOCnA pin transition immediately after the count start in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.
8	Set forcible buffer transfer	Set the GTBER.CCRSWT bit to 1 to transfer buffer register data forcibly.
9	Set buffer value	Set the GTIOCnA pin transition in 1 cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.
10	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
11	Set buffer value for each cycle	Set the GTIOCnA pin transition in 1 cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.

Note: n: 0 to 13
m: A, B

21.3.4 Automatic Dead Time Setting Function

By setting GTDTCR, a compare match value for a negative waveform with dead time obtained by a compare match value for a positive waveform (GTCCRA value) and specified dead time value (GTDVU value) can automatically be set to GTCCRB.

The automatic dead time setting function can be used in saw-wave one-shot pulse mode and all the triangle PWM modes.

Dead time for the changing point of a negative waveform is set in the GTDVU register.

The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. Writing to GTCCRB is prohibited when the automatic dead time setting function is used.

Do not set the dead time that makes the change point of the waveform exceeding the count period. When any dead-time setting which would generate a dead-time error is made, adjust the change points of the positive- and negative-phase waveforms to generate waveforms with secured dead-time as shown inTable 21.21. The adjusted change point of the negative-phase waveform is automatically set in the GTCCRB register. An internal signal is used to judge the change point of the positive-phase waveform, thus the value of the GTCCRA register is not updated by the adjusted value.

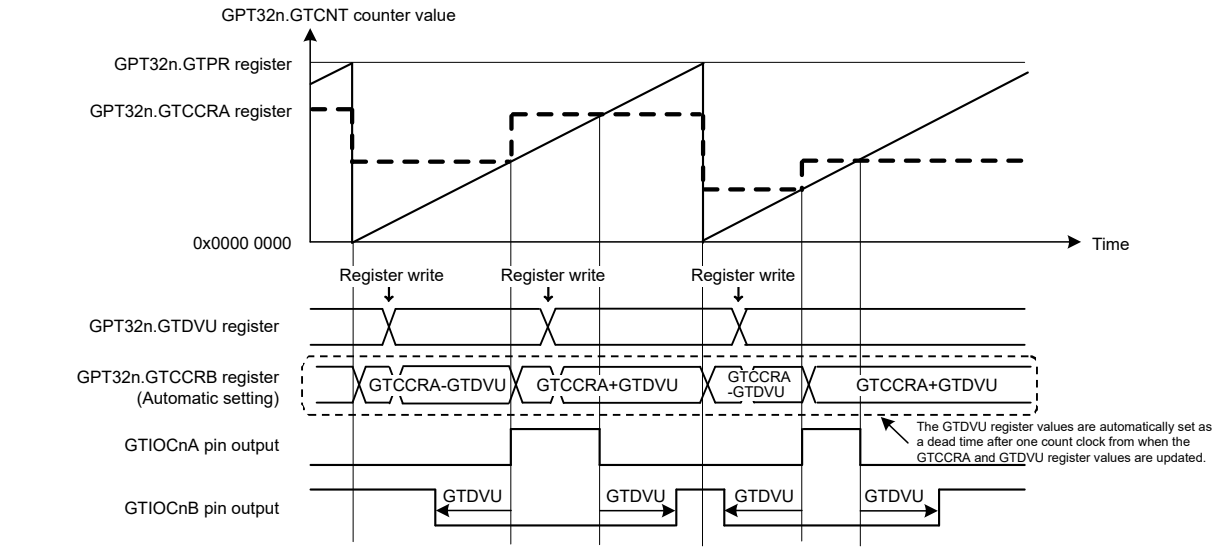
In saw-wave one-shot pulse mode, if the order of the change point becomes inconsistent by adjustment of the waveform change point due to occurrence of dead time errors, or if the change point exceeds the count period even after the adjustment, the complementary relation between the positive- and negative-phases cannot be guaranteed.

Automatic setting for a dead time value to the GTCCRB register is performed at the next count clock after the register value for calculating the automatic setting value is updated. In triangle-wave mode, it also can be done at the next count clock from the current crest.

Table 21.21 Adjustment of the Waveform Change Point When a Dead-Time Error Occurs

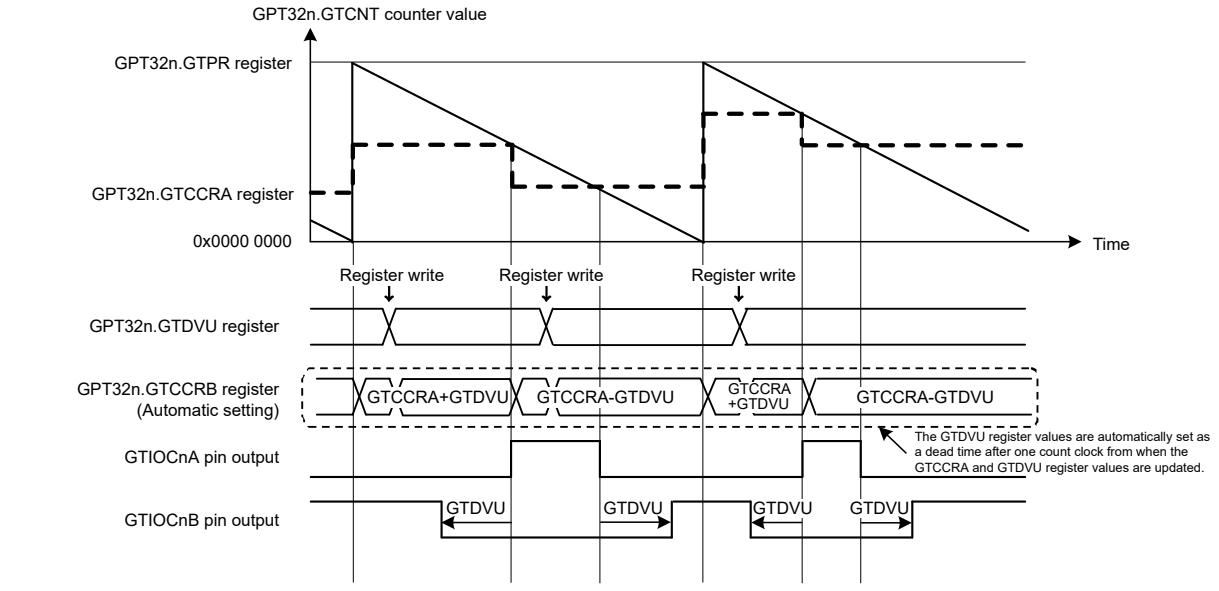
Mode	Count Direction	Period	Condition for Dead Time Error	Change Point of the Positive-Phase Waveform after Adjustment	Change Point of the Negative-Phase Waveform after Adjustment
Sawtooth-wave one-shot pulse mode	Up-counting	First half	GTCCRA - GTDVU < 0	GTDVU	0
		Second half	GTCCRA + GTDVU > GTPR	GTPR - GTDVU	GTPR
	Down-counting	First half	GTCCRA + GTDVU > GTPR	GTPR - GTDVU	GTPR
		Second half	GTCCRA - GTDVU < 0	GTDVU	0
Triangle-wave PWM mode 1/2/3	Up-counting	(First half)	GTCCRA - GTDVU ≤ 0	GTDVU + 1	1
	Down-counting	(Second half)	GTCCRA - GTDVU < 0	GTDVU	0

Figure 21.26 to Figure 21.29 show examples of automatic dead time setting function operation. Table 21.22 and Table 21.23 show the setting examples.



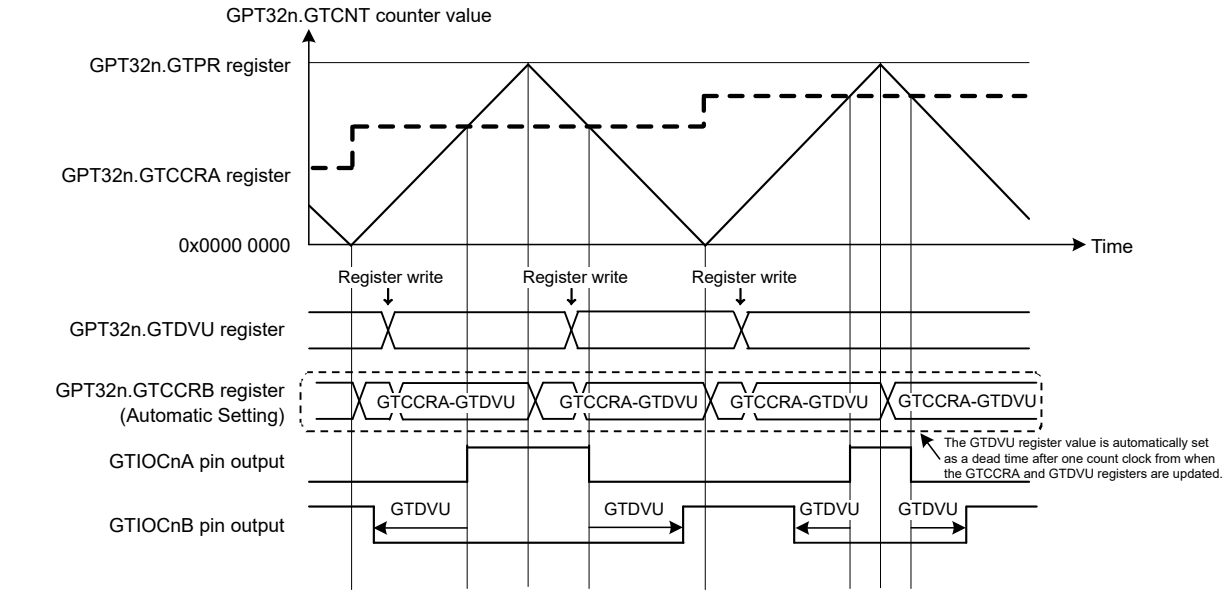
Note: The base address of registers are the same.

Figure 21.26 Example of automatic dead time setting function operation in saw-wave one-shot pulse mode, up-counting, and active-high



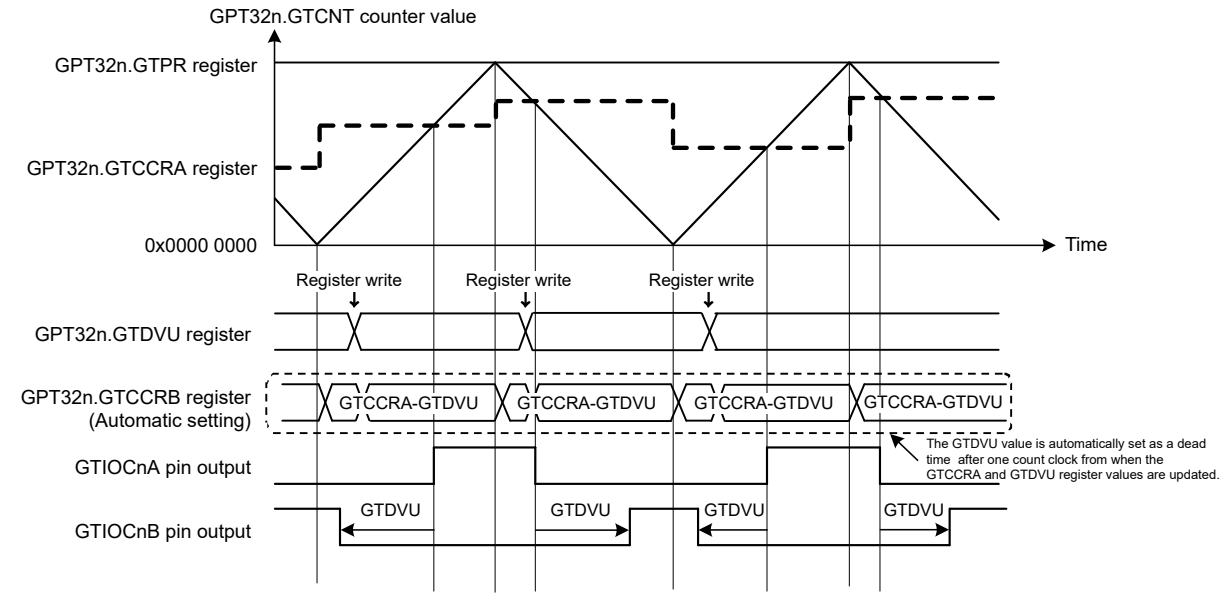
Note: The base address of registers are the same.

Figure 21.27 Example of automatic dead time setting function operation in saw-wave one-shot pulse mode, down-counting, and active-high



Note: The base address of registers are the same.

Figure 21.28 Example of automatic compare-match value setting function with dead time in triangle-wave PWM mode 1, and active-high



Note: The base address of registers are the same.

Figure 21.29 Example of automatic compare-match value setting function with dead time in triangle-wave PWM mode 2 or 3, and active-high

Table 21.22 Example setting for automatic dead time setting function in saw-wave one-shot pulse mode, and triangle-wave PWM mode 3 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.26 and Figure 21.27, 001b (saw-wave one-shot pulse mode) is set. In Figure 21.29, 110b (triangle-wave PWM mode 3) is set.

Table 21.22 Example setting for automatic dead time setting function in saw-wave one-shot pulse mode, and triangle-wave PWM mode 3 (2 of 2)

No.	Step Name	Description
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.26, 01b is set after 11b is set in the GTUDDTYC[1:0] bits (up count). In Figure 21.27, 00b is set after 10b is set in the GTUDDTYC[1:0] bits (down count).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.26, Figure 21.27, and Figure 21.29, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
7	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set buffer value for compare match	Set the GTIOCnA pin transition immediately after the count start in the GTCCRC and GTCCRD registers.
9	Set forcible buffer transfer for compare match	Set the GTBER.CCRSWT bit to 1 to transfer buffer register data forcibly to the GTCCRA register.
10	Set buffer value for compare match	Set the GTIOCnA pin transition in 1 cycle after the current cycle in the GTCCRC and GTCCRD registers.
11	Set automatic dead time setting function	Set the GTDTCR.TDE bit to 1 to enable the automatic dead time setting function.
12	Set dead time value	Set the first half dead time value in GTDVU.
13	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
14	Set buffer value for each cycle	Set the GTIOCnA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD.

Note: n: 0 to 13
m: A, B

Table 21.23 Example setting for automatic dead time setting function in triangle-wave PWM mode 1 or 2 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.28, 100b (triangle-wave PWM mode 1) is set. In Figure 21.29, 101b (triangle-wave PWM mode 2) is set.
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.28 and Figure 21.29, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
7	Set buffer operation for compare match	Set buffer operation with the CCRA[1:0] bits in the GTBER register.
8	Set compare match value	Set the GTIOCnA pin transition in the GTCCRA register.
9	Set buffer value for compare match	For buffer operation, set the GTIOCnA pin transition in 1 cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in the GTCCRC register. For double buffer operation, also set the GTIOCnA pin transition in 2 cycles after the current cycle (in triangle-wave PWM mode 1) or 1 cycle after the current cycle (in triangle-wave PWM mode 2) in the GTCCRD registers.
10	Set automatic dead time setting function	Set the GTDTCR.TDE bit to 1 to enable the automatic dead time setting function.
11	Set dead time value	Set the first half dead time value in GTDVU.
12	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

Table 21.23 Example setting for automatic dead time setting function in triangle-wave PWM mode 1 or 2 (2 of 2)

No.	Step Name	Description
13	Set buffer value for each cycle	When the compare match register is used for buffer operation, set the GTIOCnA pin transition in 1 cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRC. When the compare match register is used for double-buffered operation, set the GTIOCnA pin changing point in two cycles after the current cycle (in triangle-wave PWM mode 1) or one cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRD.

Note: n: 0 to 13
m: A, B

21.3.5 Count Direction Changing Function

The count direction of the GTCNT counter can be changed by modifying the UD bit in GTUDDTYC.

In saw-wave mode, if the UD bit in GTUDDTYC is modified during count operation, the count direction is changed at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the GTUDDTYC.UD bit is modified while the count operation stops and the GTUDDTYC.UDF bit is 0, the GTUDDTYC.UD bit modification is not reflected at the start of counting and the count direction is changed at an overflow or an underflow. If the UDF bit is set to 1 while the count operation stops, the GTUDDTYC.UD bit value at that time is reflected at the start of counting.

In triangle-wave mode, the count direction does not change even though the UD bit in GTUDDTYC is modified during the count operation. Similarly, even though the GTUDDTYC.UD bit is modified while the count operation stops and GTUDDTYC.UDF bit is 0, the GTUDDTYC.UD bit value is not reflected to the count operation. If the GTUDDTYC.UDF bit is set to 1 while the count operation is stopped, the GTUDDTYC.UD bit value at that time is reflected at the start of counting.

If the count direction changes during a saw-wave count operation, the GTPR value after the start of up-counting is reflected in the count cycle during up-counting and the GTPR value after the start of down-counting is reflected in the count cycle during down-counting.

Figure 21.30 shows an example of count direction changing function operation.

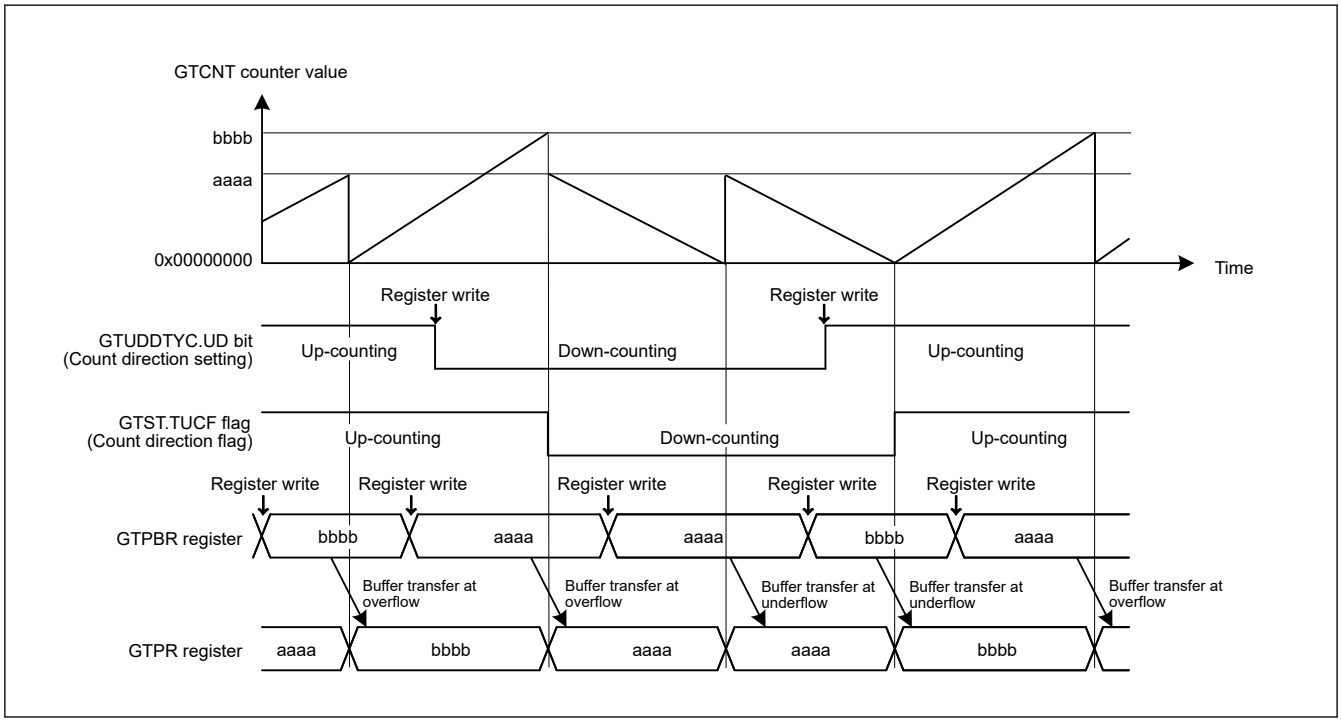


Figure 21.30 Example of a count direction changing function operation during buffer operation

21.3.6 Function of Output Duty 0% and 100%

The output duty of the GTIOCnA pin and the GTIOCnB pin (n = 0 to 13) are set to 0% or 100% by changing the GTUDDTYC.OADTY bit or GTUDDTYC.OBDTY bit.

In saw-wave mode, if the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified during the count operation, the output duty setting is reflected at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation is stopped and the GTUDDTYC.OADTYF bit or the GTUDDTYC.OBDTYF bit is 0, the output duty modification is not reflected at the start of counting. The output duty changes at an overflow or an underflow. If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation is stopped and the GTUDDTYC.OADTYF bit or the GTUDDTYC.OBDTYF bit is 1, the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit value at that time is reflected at the start of counting.

In triangle-wave mode, if the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified during the count operation, the output duty setting is reflected an underflow.

If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation is stopped and the GTUDDTYC.OADTYF bit or the GTUDDTYC.OBDTYF bit is 0, the output duty modification is not reflected at the start of counting. The output duty changes at an underflow. If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation stops and the GTUDDTYC.OADTYF bit or the GTUDDTYC.OBDTYF bit is 1, the output duty modification is reflected at the start of counting.

In performing 0% or 100% duty operation, GPT internally continues to:

- Perform compare match operation
- Set compare match flag
- Output interrupt
- Perform buffer operation.

When the control is changed from 0% or 100% duty setting to compare match, the output value of GTIOCnA pin at cycle end is decided by GTIOR.GTIOA[3:2] and GTUDDTYC.OADTYR. The output value of GTIOCnB pin at cycle end is decided by GTIOR.GTIOB[3:2] and GTUDDTYC.OBDTYR.

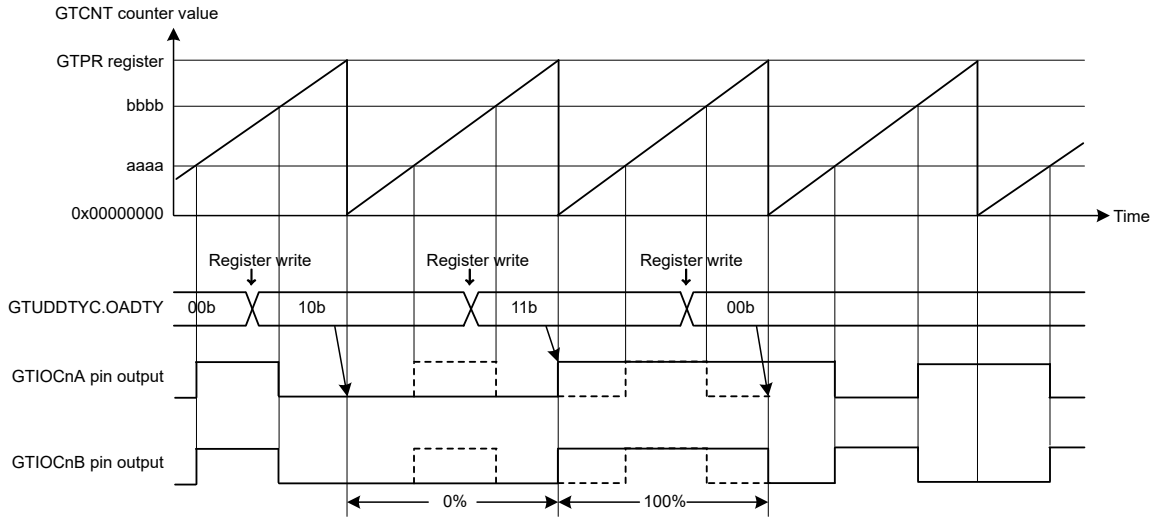
When GTIOR.GTIOA[3:2] and GTIOR.GTIOB[3:2] are set to 01b, the output pins output low at cycle end. When GTIOR.GTIOA[3:2] and GTIOR.GTIOB[3:2] are set to 10b, the output pins output high at cycle end.

GTUDDTYC.OADTYR selects the value that is the object of output retained/toggled at cycle end, when GTIOR.GTIOm[3:2] are set to 00b (output retained at cycle end) or when GTIOR.GTIOm[3:2] are set to 11b (output toggled at cycle end). Table 21.24 shows the values of GTIOCnA and GTIOCnB pin output at cycle end.

Table 21.24 Output values after releasing 0% or 100% duty setting (m = A, B)

GTIOR.GTIOm[3:2]	Compare match value at cycle end masked by 0% or 100% duty setting	GTUDDTYC.OmDTYR in duty 0% setting		GTUDDTYC.OmDTYR in duty 100% setting	
		0	1	0	1
00 (output retained at cycle end)	0	0	0	1	0
	1	0	1	1	1
01 (low output at cycle end)	—	0	0	0	0
10 (high output at cycle end)	—	1	1	1	1
11 (output toggled at cycle end)	0	1	1	0	1
	1	1	0	0	0

Figure 21.31 shows an example of output duty 0% and 100% function.



Note: The base address of registers are the same.

Note: [Setting examples]

GTIOR.GTIOA[4:0] bits:	00011b initial low output, output toggled at compare match, output retained at cycle end
GTUDDTYC.OADTYR bit:	0b The function selected by GTIOA[3:2] is applied to the output value when the duty cycle is set after release from 0 or 100% duty-cycle setting.
GTIOR.GTIOB[4:0] bits:	00011b initial low output, output toggled at compare match, output retained at cycle end
GTUDDTYC.OBDTYR bit:	1 The function of GTIOB[3:2] is applied to the compare match output value which is masked after 0 or 100% duty setting release.

Figure 21.31 Example of output duty 0% and 100% function

21.3.7 Hardware Count Start/Count Stop and Clear Operation

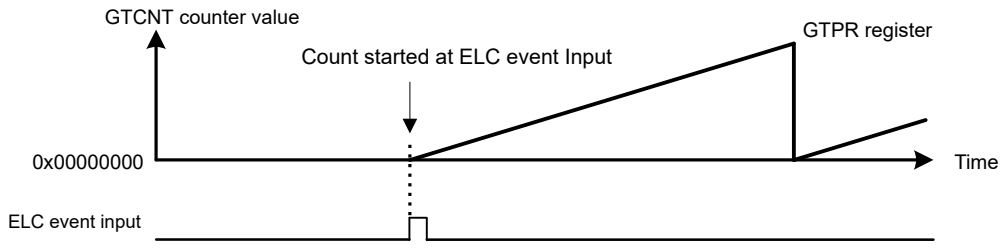
The GTCNT counter can be started, stopped, or cleared by the following hardware sources:

- External trigger input
- ELC event input
- GTIOCnA and GTIOCnB pin input (n = 0 to 13).

21.3.7.1 Hardware Start Operation

The GTCNT counter can be started by selecting a hardware source using GTSSR.

Figure 21.32 shows an example of a count start operation by a hardware source. Table 21.25 shows the setting example.



Note: ELC event input: ELCA event input

Figure 21.32 Example of count start operation by a hardware source started at the input of the signal from the ELCA event

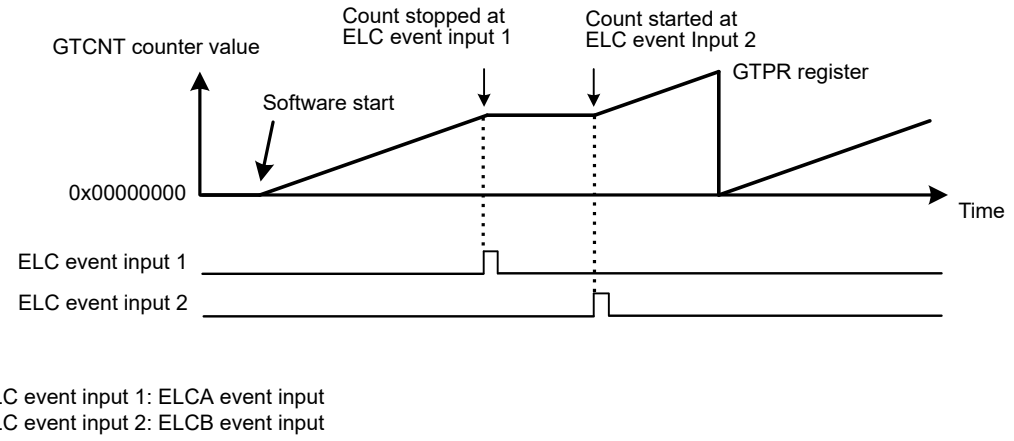
Table 21.25 Example setting for count start operation by a hardware source

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.32, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.32, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 21.32, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation in the GTSSR register. In Figure 21.32, GTSSR.SSELCA = 1
7	Set hardware source operation	Set operation of the hardware source selected by the GTSSR register and start counting. In Figure 21.32, the ELCA event input operation is set.

21.3.7.2 Hardware Stop Operation

The GTCNT counter can be stopped by selecting a hardware source using GTPSR.

Figure 21.33 shows an example of a count stop operation by a hardware source. Table 21.26 shows the setting example. In this example, the count operation stops at the ELCA event input and restarts at the ELCB event input.



Note: ELC event input 1: ELCA event input
ELC event input 2: ELCB event input

Figure 21.33 Example of count stop operation by hardware source started by software, stopped at ELCA input, and restarted at ELCB input

Table 21.26 Example setting for count stop operation by a hardware source

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0] bits. In Figure 21.33, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.33, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[3:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 21.33, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation in GTSSR register, and wait for count start by the hardware source. In Figure 21.33, GTSSR.SSELCB = 1.
7	Set hardware count stop	Select a hardware source for stopping count operation in GTPSR register and wait for count stop by the hardware source. In Figure 21.33, GTPSR.PSELCA = 1.
8	Set hardware source operation	Set operation of the hardware source selected in GTSSR register or GTPSR register, and start or stop counting. In Figure 21.33, ELCA input operation and ELCB input operation are set.

Figure 21.34 shows an example of a count start/stop operation by a hardware source. Table 21.27 shows the setting example. In this example, the counter operates during the high-level periods of the external trigger input GTETRGA.

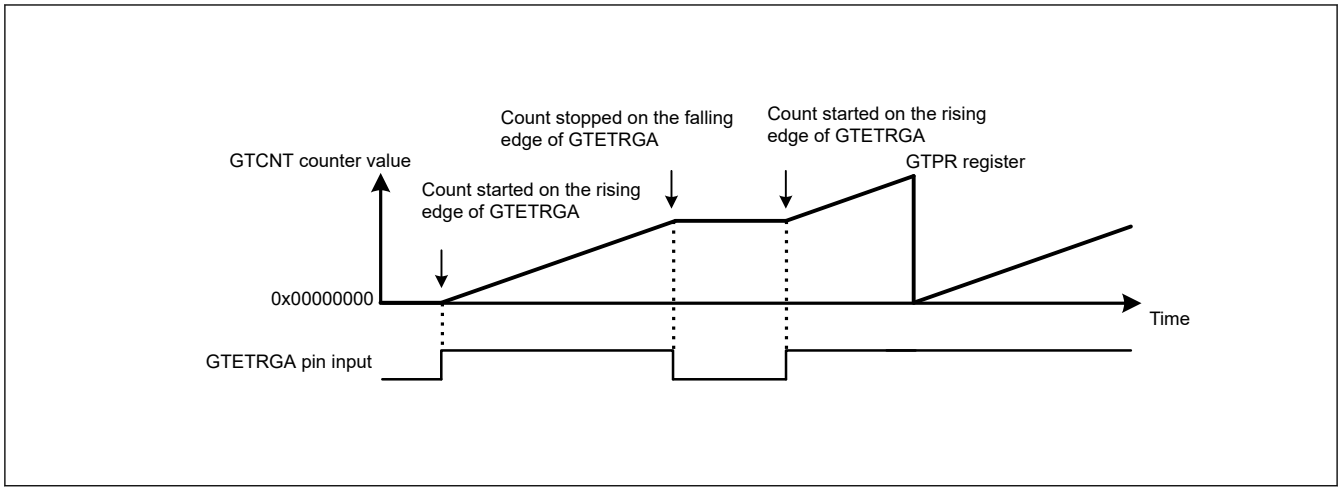


Figure 21.34 Example of count start/stop operation by a hardware source started on the rising edge of GTETRGA pin input, and stopped on the falling edge of GTETRGA pin input

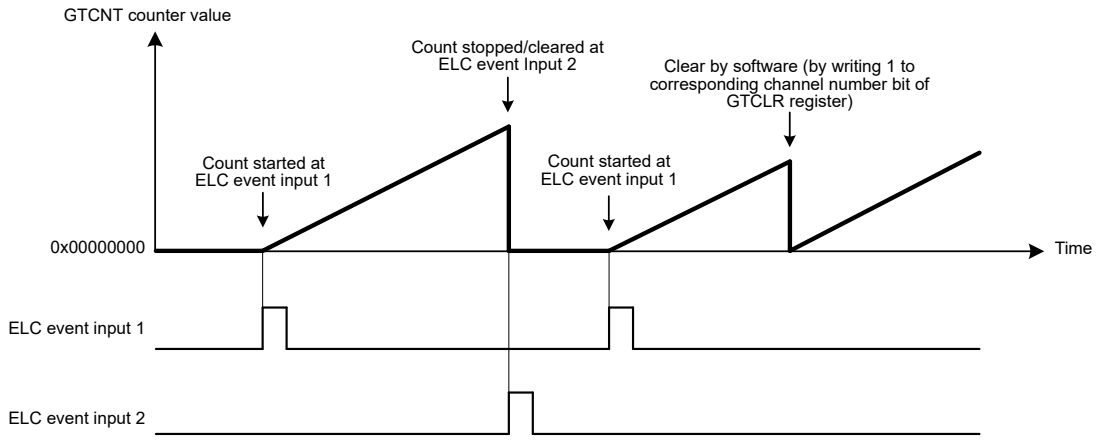
Table 21.27 Example setting for count start/stop operation by a hardware source

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.34, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.34, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 21.34, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation with the GTSSR register, and wait for count start by the hardware source. In Figure 21.34, GTSSR.SSGTRGAR = 1.
7	Set hardware count stop	Select a hardware source for stopping count operation with the GTPSR register, and wait for count stop by the hardware source. In Figure 21.34, GTPSR.PSGTRGAF = 1.
8	Set hardware source operation	Set operation of the hardware source selected in the GTSSR register or GTPSR register and start or stop counting. In Figure 21.34, the GTETRGA pin operation is set.

21.3.7.3 Hardware Clear Operation

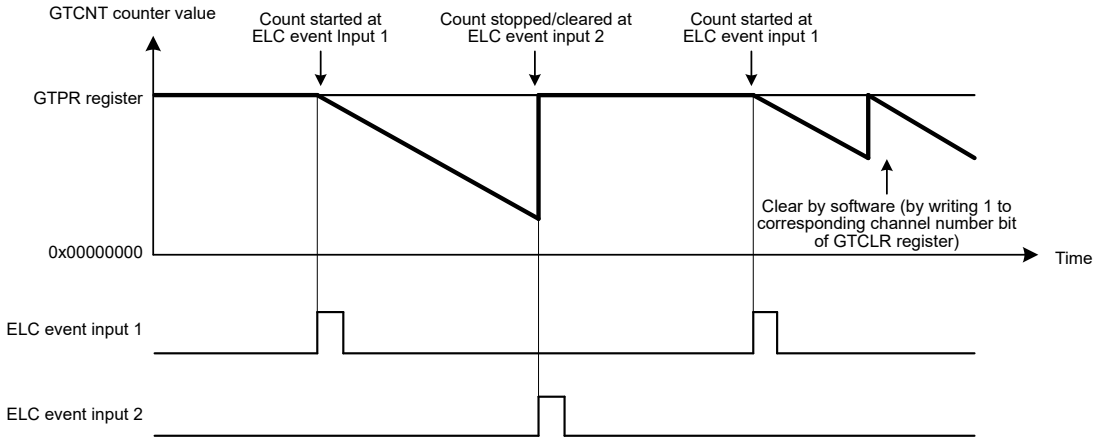
The GTCNT counter can be cleared by selecting a hardware source using GTCSR. The GPTn_OVF/GPTn_UDF (n = 0 to 13) interrupt (overflow/underflow interrupt) is not generated when the GTCNT counter is cleared by a hardware source or by software.

Figure 21.35 and Figure 21.36 show examples of the GTCNT counter clearing operation by a hardware source. Table 21.28 shows the setting example. In this example, the GTCNT counter starts at the ELCA input, and the counter stops and clears at the ELCB input.



Note: ELC event input 1: ELCA event input
ELC event input 2: ELCB event input

Figure 21.35 Examples of count clearing operation by hardware source in saw wave up-counting, started at ELCA input, and stopped/cleared at ELCB input



Note: ELC event input 1: ELCA event input
ELC event input 2: ELCB event input

Figure 21.36 Examples of count clearing operation by hardware source in saw wave down-counting, started at ELCA input, and stopped/cleared at ELCB input

Table 21.28 Example setting for count clearing operation by a hardware source (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.35 and Figure 21.36, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.35, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting). In Figure 21.36, after 10b is set in the GTUDDTYC[1:0] bits, 00b is set in the GTUDDTYC[1:0] bits (down-counting).

Table 21.28 Example setting for count clearing operation by a hardware source (2 of 2)

No.	Step Name	Description
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 21.35, 0x00000000 is set. In Figure 21.36, the GTPR register value is set.
6	Set hardware count start	Select a hardware source for starting count operation in the GTSSR register, and wait for count start by the hardware source. In Figure 21.35 and Figure 21.36, GTSSR.SSELCA = 1.
7	Set hardware count stop	Select a hardware source for stopping count operation in the GTPSR register, and wait for count stop by the hardware source. In Figure 21.35 and Figure 21.36, GTPSR.PSELCB = 1.
8	Set hardware count clear	Select a hardware source for clearing count operation in the GTCSR register, and wait for count clear by the hardware source. In Figure 21.35 and Figure 21.36, GTCSR.CSELCB = 1.
9	Set hardware source operation	Set operation of the hardware source selected in the GTSSR register, GTPSR register or GTCSR register and start, stop or clear counting. In Figure 21.35 and Figure 21.36, the ELCA input and ELCB input are set.

The GPTn_OVF/GPTn_UDF (n = 0 to 13) interrupt (overflow/underflow interrupt) is not generated when the counter is cleared by a hardware source or by software.

Figure 21.37 shows the relationship between the counter clearing by a hardware source and the GPTn_OVF (n = 0 to 13) interrupt.

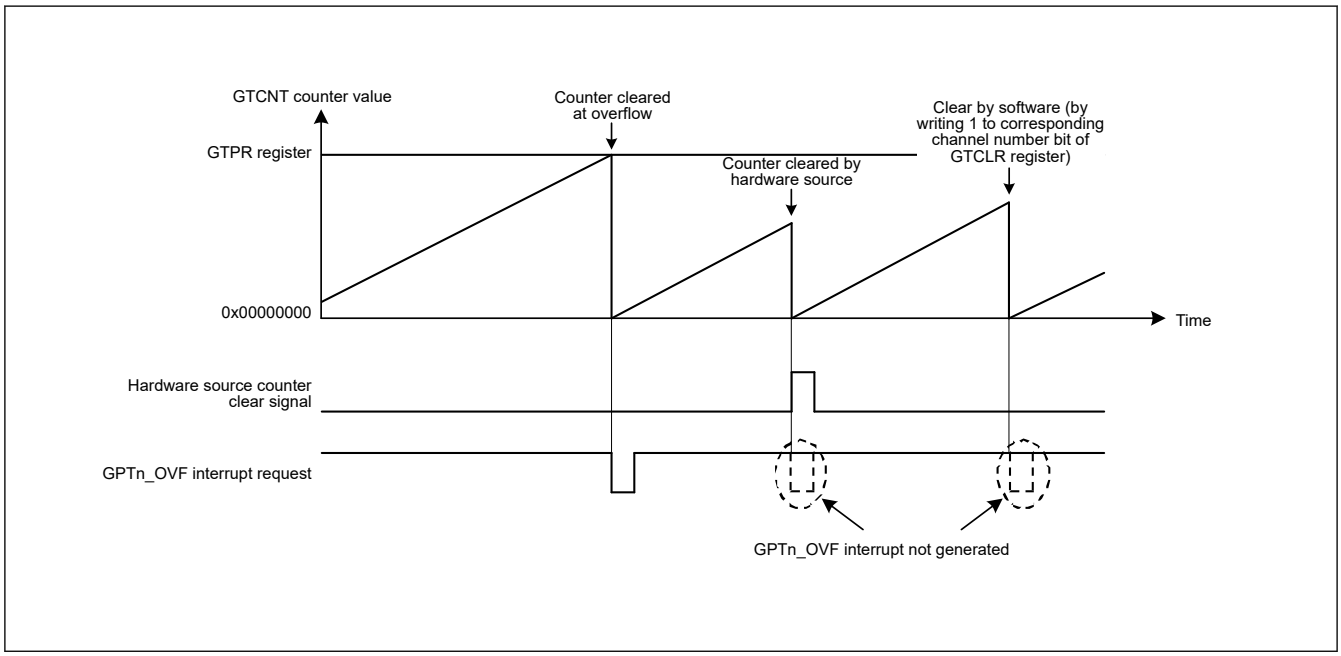


Figure 21.37 Relationship between counter clearing by hardware source and GPTn_OVF (n = 0 to 13) interrupt

21.3.8 Synchronized Operation

Synchronized operation on channels such as a synchronized start, stop, and clear operation can be performed.

21.3.8.1 Synchronized Operation by Software

The GTCNT counters can be started, stopped, and cleared on multiple channels by setting the associated GTSTR, GTSTP, or GTCLR bits simultaneously to 1.

Count start with a phase difference is possible by setting the initial value in the GTCNT counter and setting the associated GTSTR bits simultaneously to 1.

Because the clock of count operation is selected by GTCR.TPCS[3:0] bits in respective channels, if the clock period of each channel that performs synchronous operation (count start/stop/clear) is different from others, the synchronous operation timings of every channels are not exact same.

Figure 21.38 shows an example of a simultaneous start, stop, and clear by software. Figure 21.39 shows an example of phase start operation by software. Figure 21.40, Figure 21.41, Figure 21.42 show an example of simultaneous start/stop/clearing with different count period.

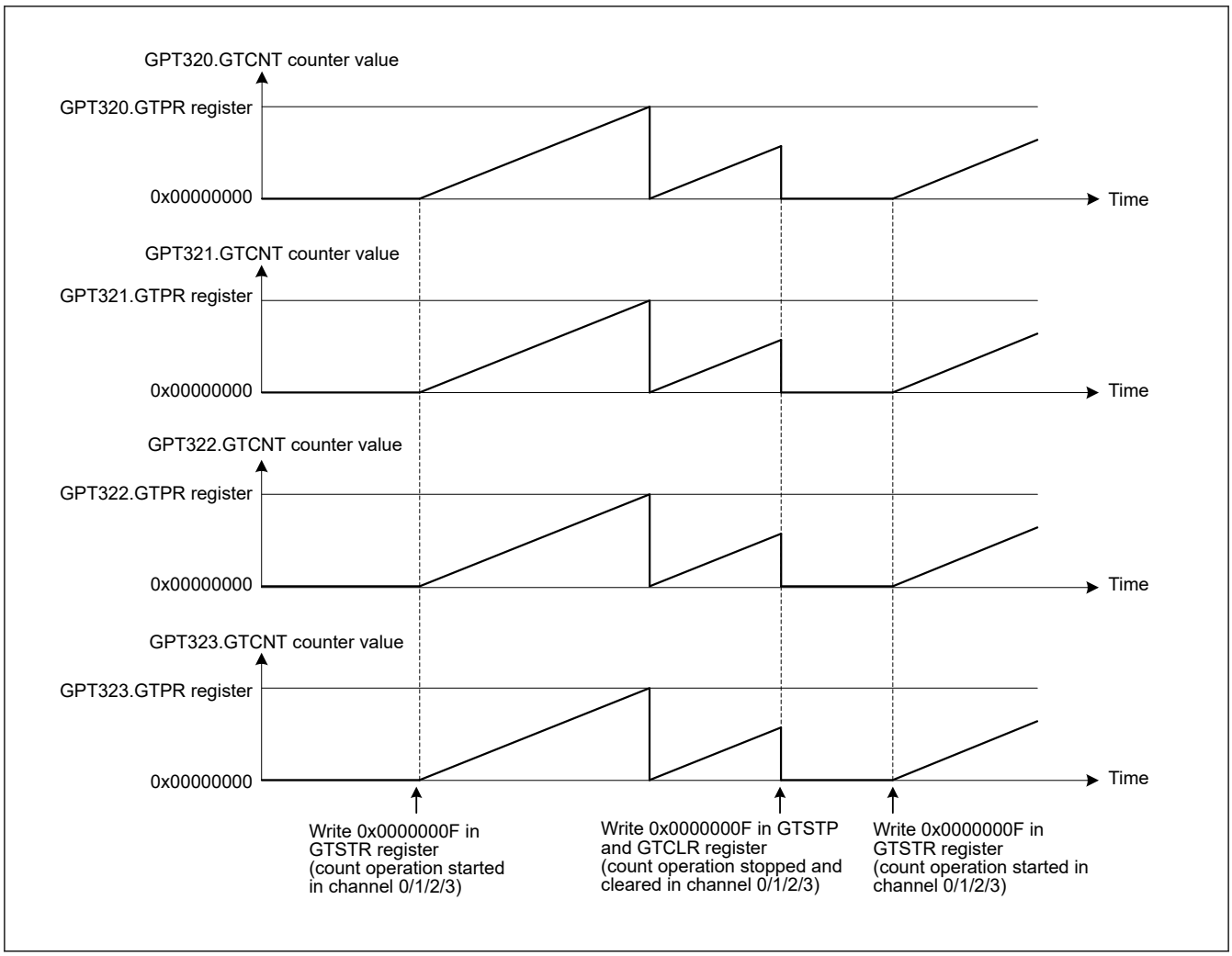


Figure 21.38 Example of a simultaneous start, stop, and clear by software with the same count cycle (GTPR register value)

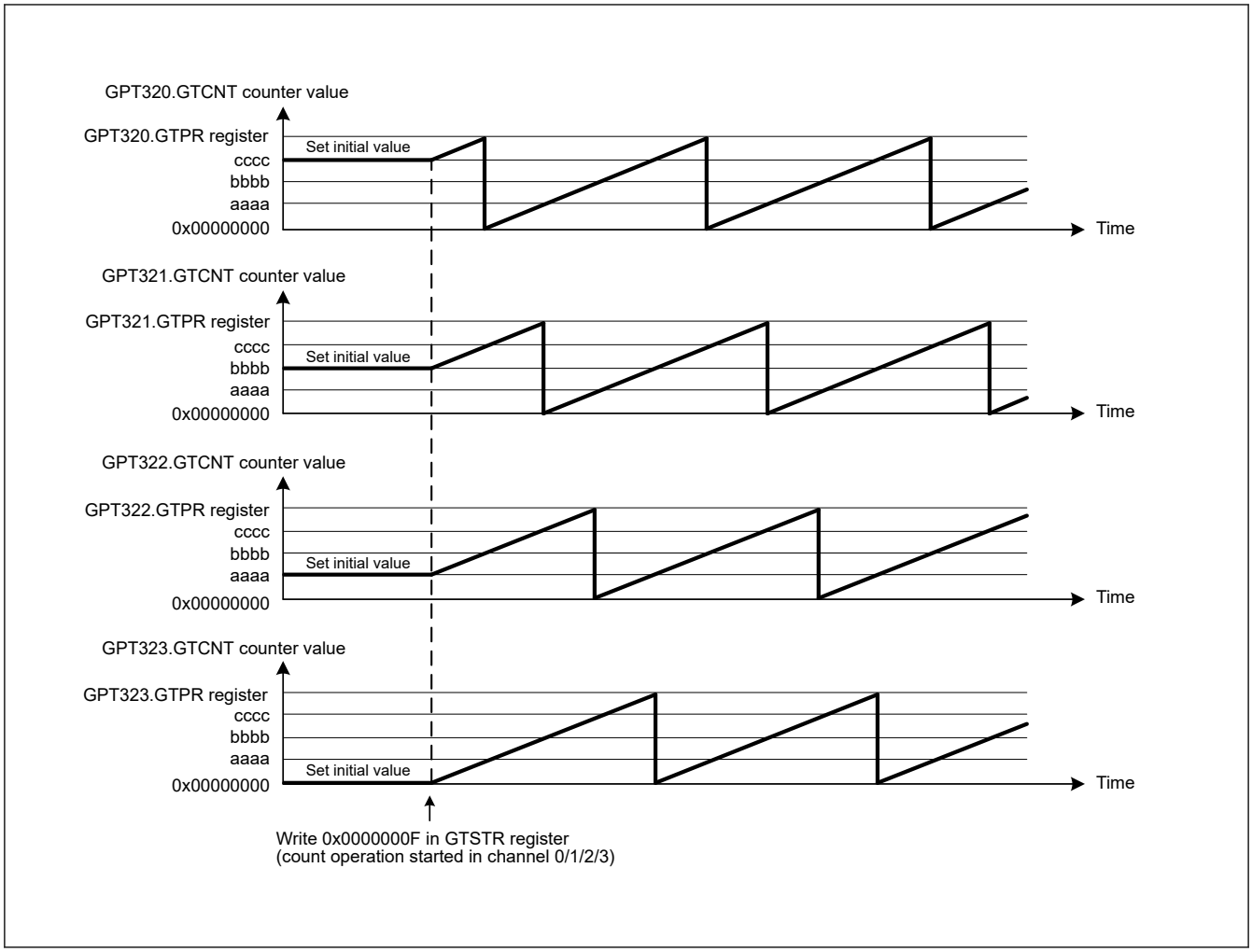


Figure 21.39 Example of software phase start with the same count cycle (GTPR register value)

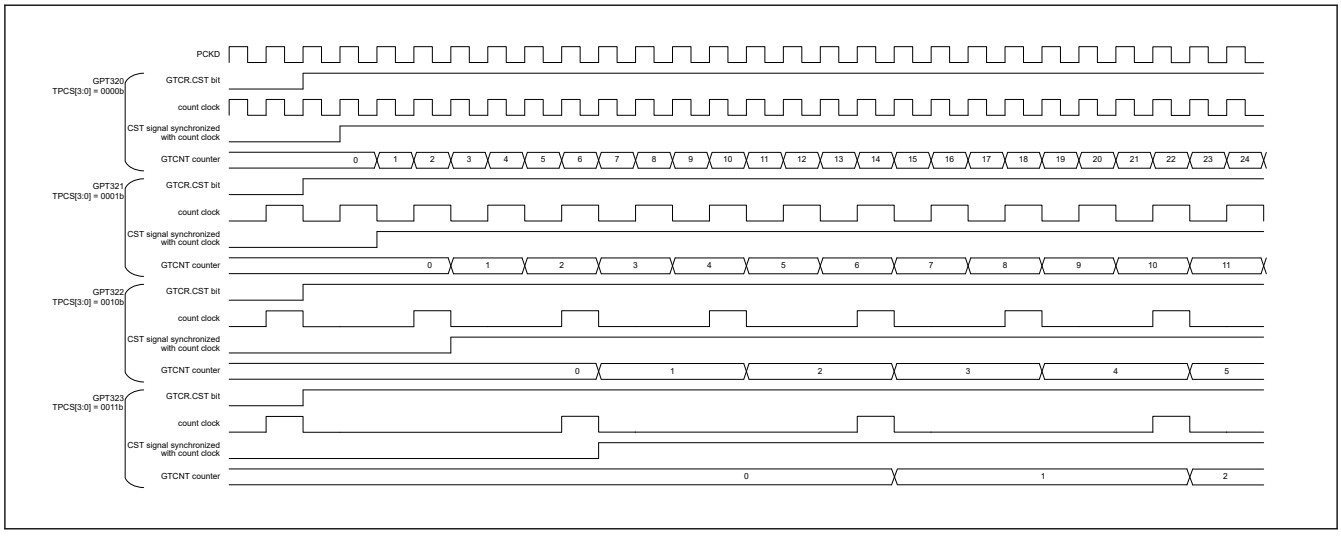


Figure 21.40 Example of Simultaneous Start Operation by Software (with Different Count Period)

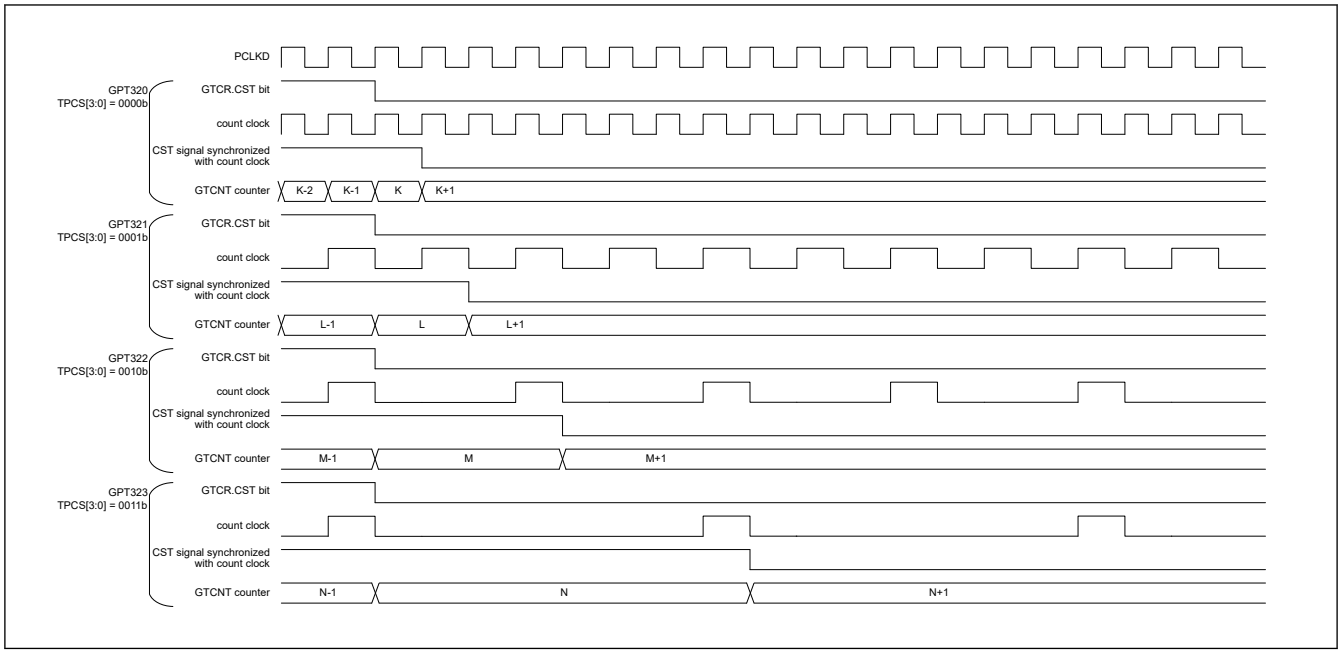


Figure 21.41 Example of Simultaneous Stop Operation by Software (with Different Count Period)

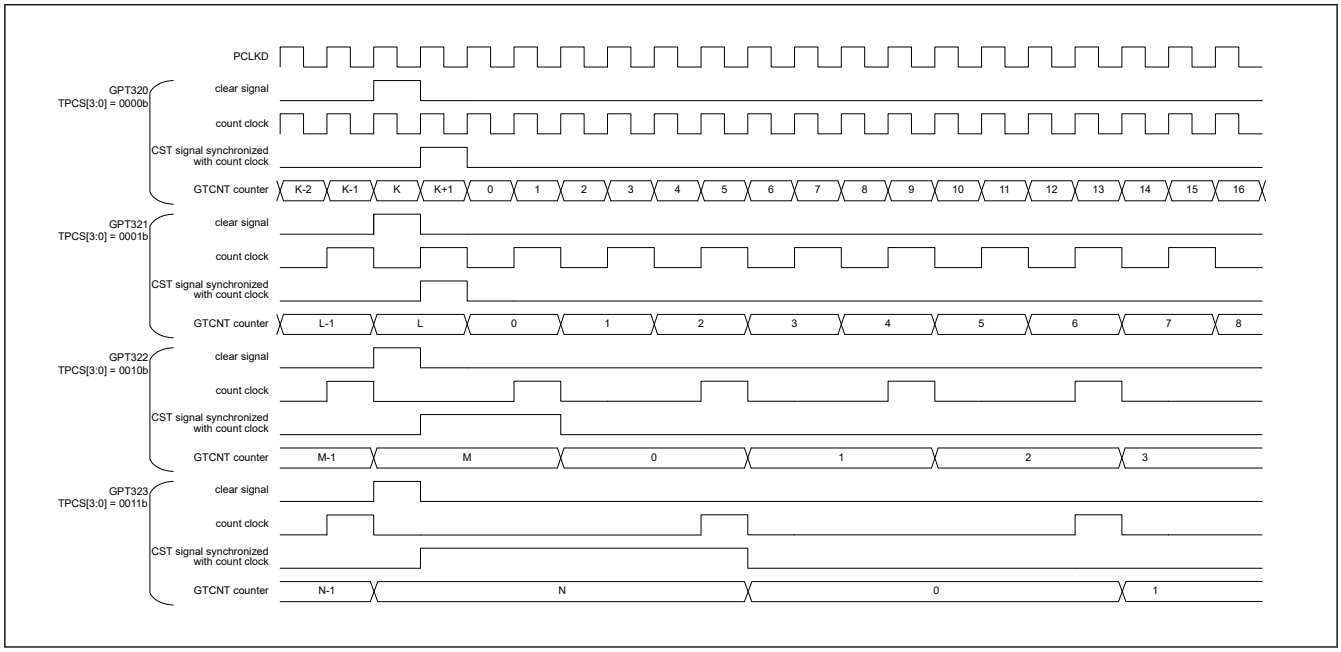
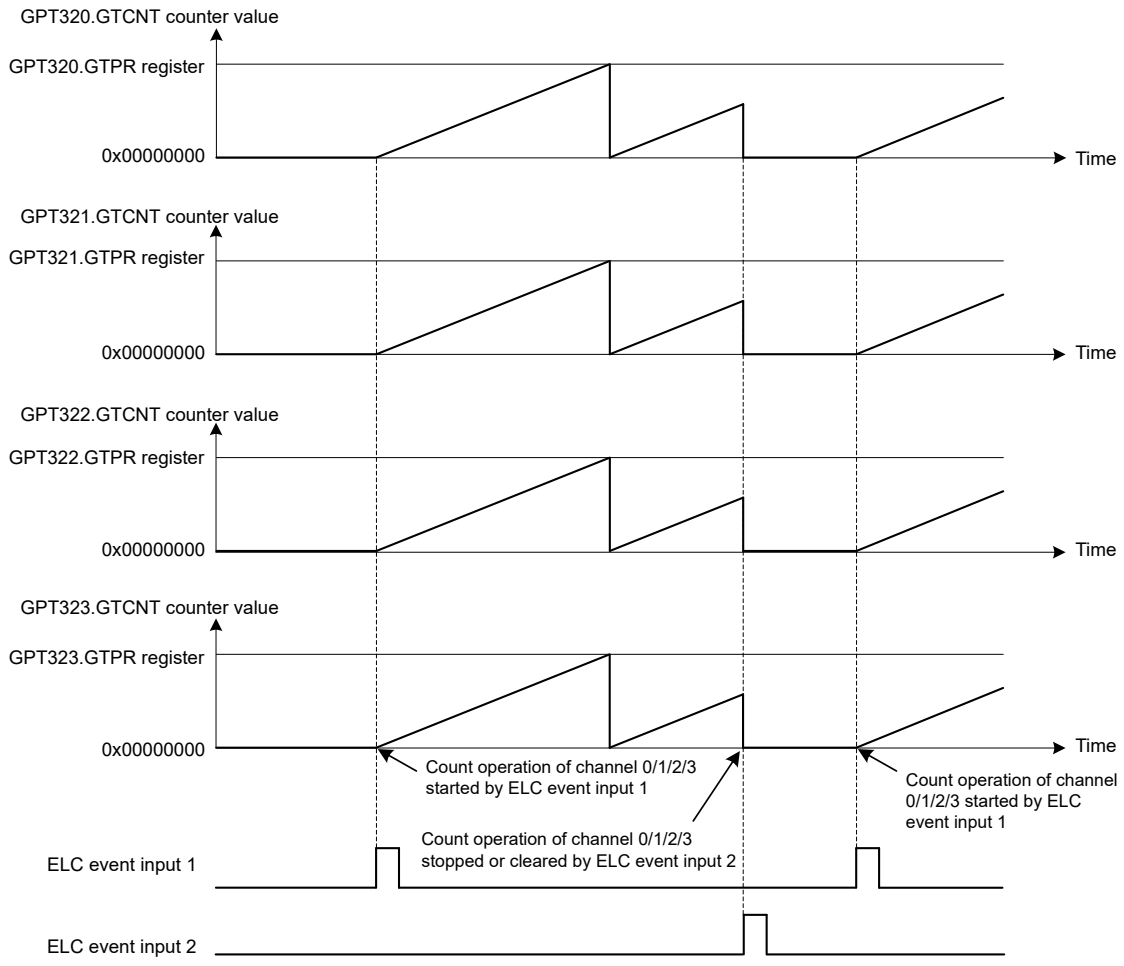


Figure 21.42 Example of Simultaneous Clearing Operation by Software (with Different Count Period)

21.3.8.2 Synchronized Operation by Hardware

The counters for multiple channels can be started, stopped, and cleared simultaneously by ELC event input.

Figure 21.43 shows an example of a simultaneous start, stop, and clear operation by a hardware source. Table 21.29 shows the setting example.



Note: ELC event input 1: ELCA event input
ELC event input 2: ELCB event input

Figure 21.43 Example of a simultaneous start, stop, and clear by a hardware source with the same count cycle (GTPR register value)

Table 21.29 Example setting for simultaneous start by a hardware source (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.43, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.43, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 21.43, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation with the GTSSR register, and wait for count start by the hardware source. In Figure 21.43, GTSSR.SSELCA = 1.

Table 21.29 Example setting for simultaneous start by a hardware source (2 of 2)

No.	Step Name	Description
7	Set hardware count stop	Select a hardware source for stopping count operation with the GTPSR register, and wait for count stop by the hardware source. In Figure 21.43, GTPSR.PSELCB = 1.
8	Set hardware count clear	Select a hardware source for clearing count operation with the GTCSR register, and wait for count clear by the hardware source. In Figure 21.43, GTCSR.CSELCB = 1.
9	Set hardware source operation	Set operation of the hardware source selected in the GTSSR, GTPSR, or GTCSR registers, and start, stop, or clear counting. In Figure 21.43, ELCA input and ELCB input are set.

21.3.9 PWM Output Operation Examples

(1) Synchronized PWM output

The GPT outputs 14 × 2 phases of linked PWM waveforms for a maximum of GPT × 14 channels.

Figure 21.44 shows an example in which four channels perform synchronized operation in saw-wave PWM mode and eight phases of PWM waveforms are output. The GTIOCnA is set so that it outputs low as the initial value, high at a GTCCRA compare match, and low at the cycle end. The GTIOCnB is set so that it outputs low as the initial value, high at a GTCCRB compare match, and low at the cycle end.

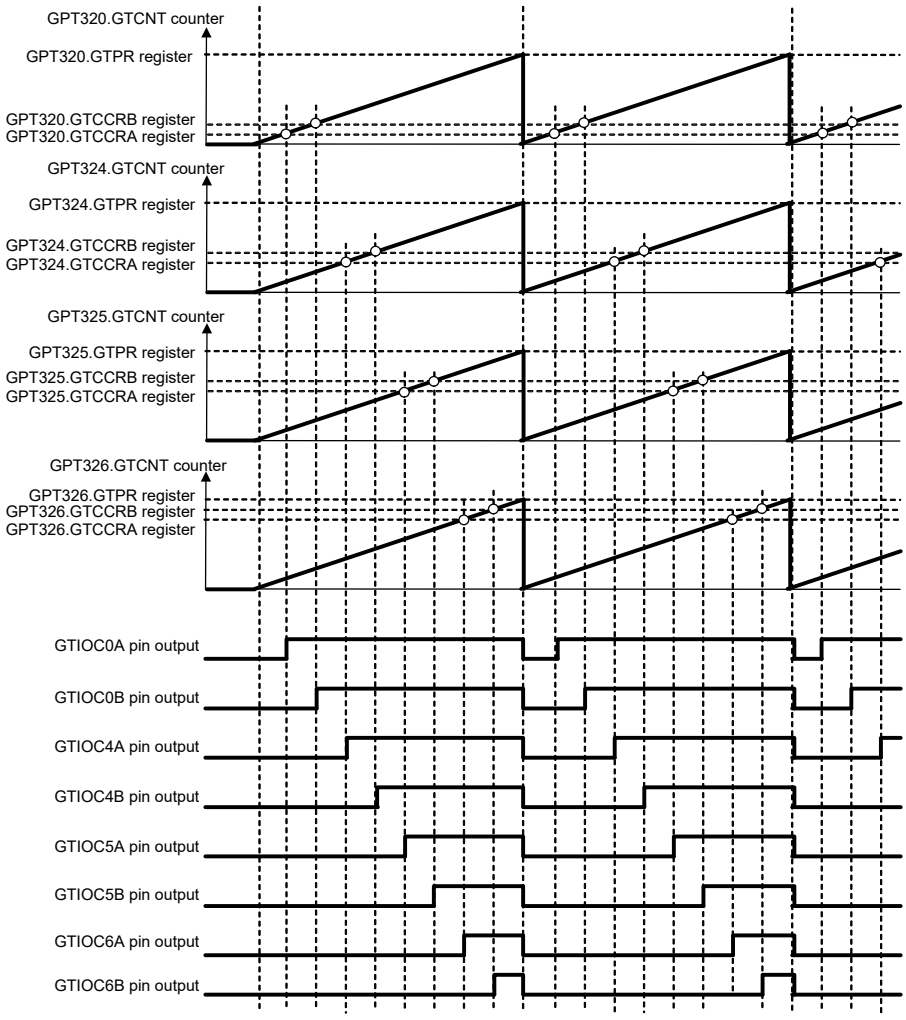


Figure 21.44 Example of synchronized PWM output

(2) 3-phase saw-wave complementary PWM output

Figure 21.45 shows an example in which three channels perform synchronized operation in saw-wave PWM mode and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, high at a GTCCRA compare match, and low at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, low at a GTCCRB compare match, and high at the cycle end.

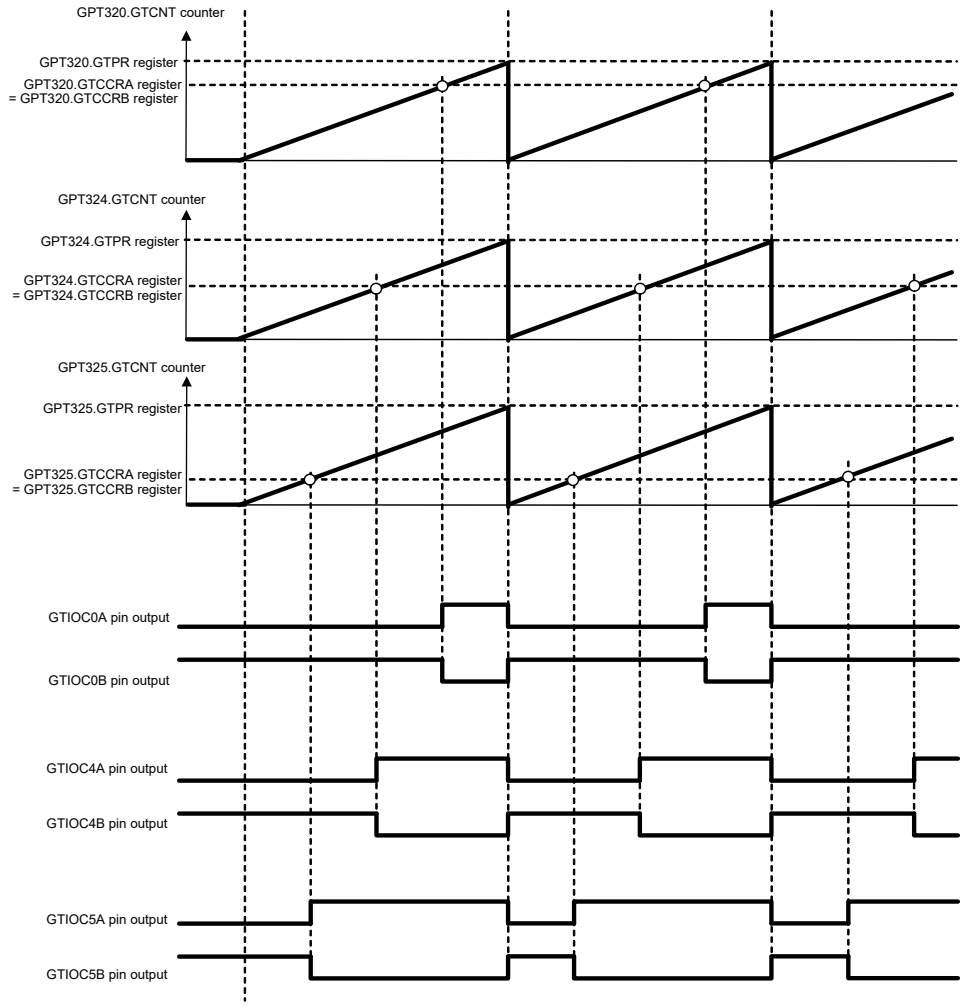


Figure 21.45 Example of 3-phase saw-wave complementary PWM output

(3) 3-phase saw-wave complementary PWM output with automatic dead time setting

Figure 21.46 shows an example in which three channels perform synchronized operation in saw-wave one-shot pulse mode with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, toggles the output at a GTIOCnB compare match, and retains the output at the cycle end.

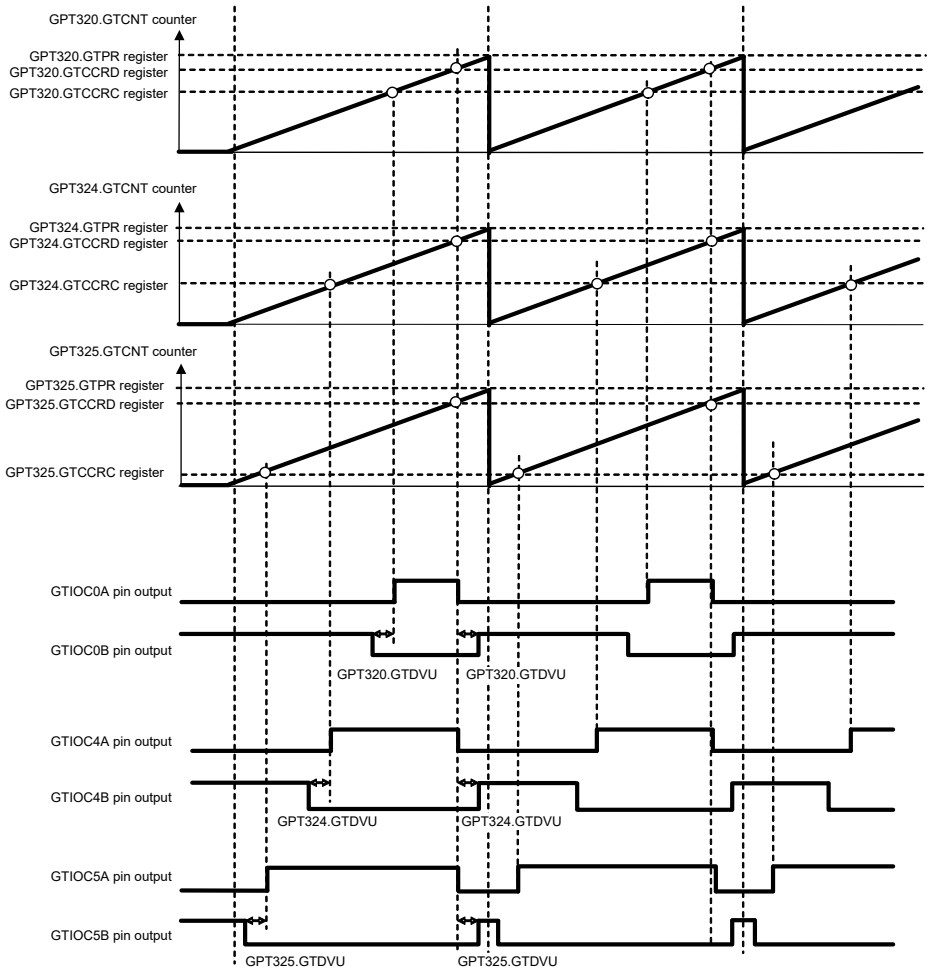


Figure 21.46 Example of 3-phase saw-wave complementary PWM output with automatic dead time setting

(4) 3-phase triangle-wave complementary PWM output

Figure 21.47 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 1 and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

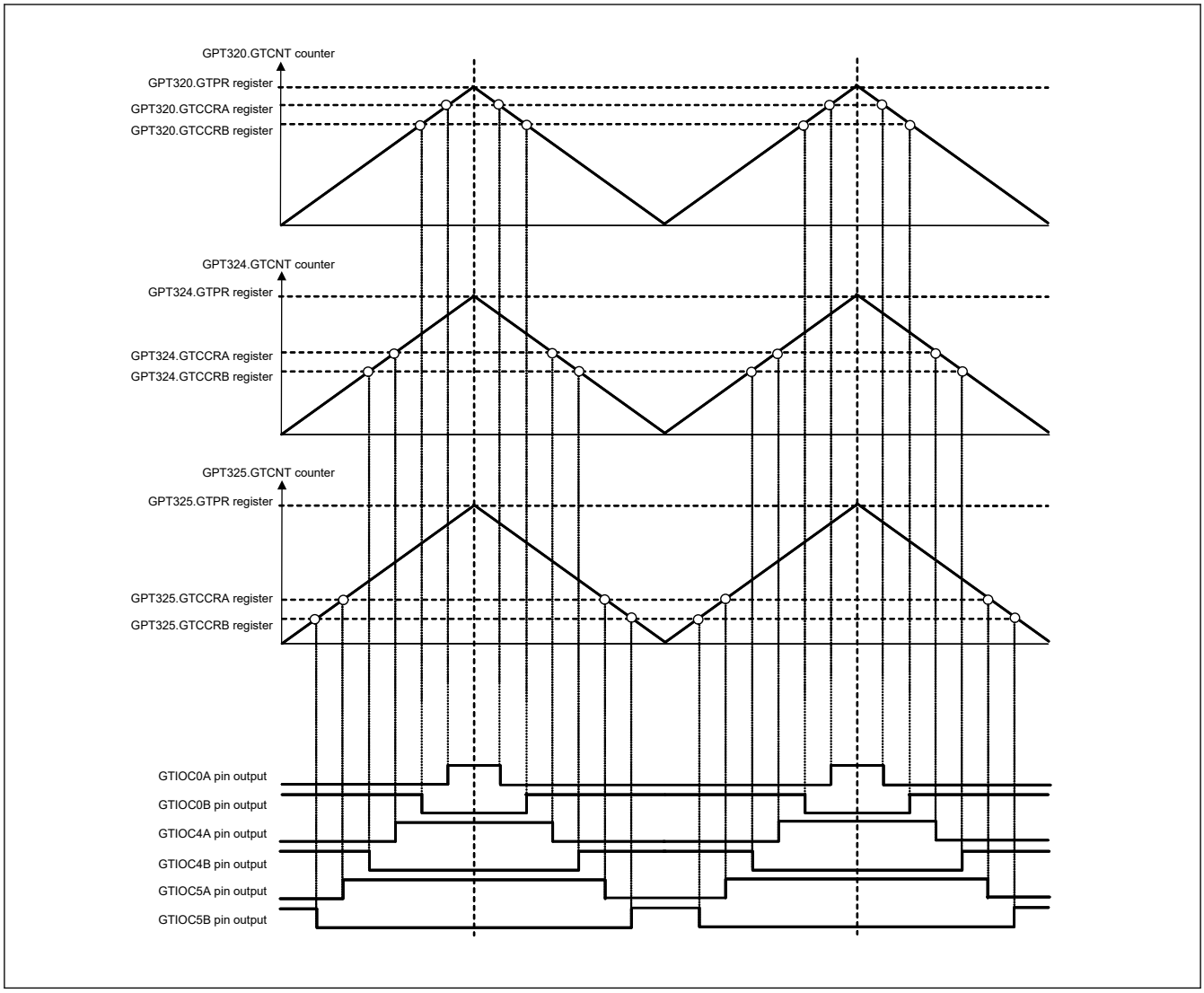


Figure 21.47 Example of 3-phase triangle-wave complementary PWM output

(5) 3-phase triangle-wave complementary PWM output with automatic dead time setting

Figure 21.48 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 1 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

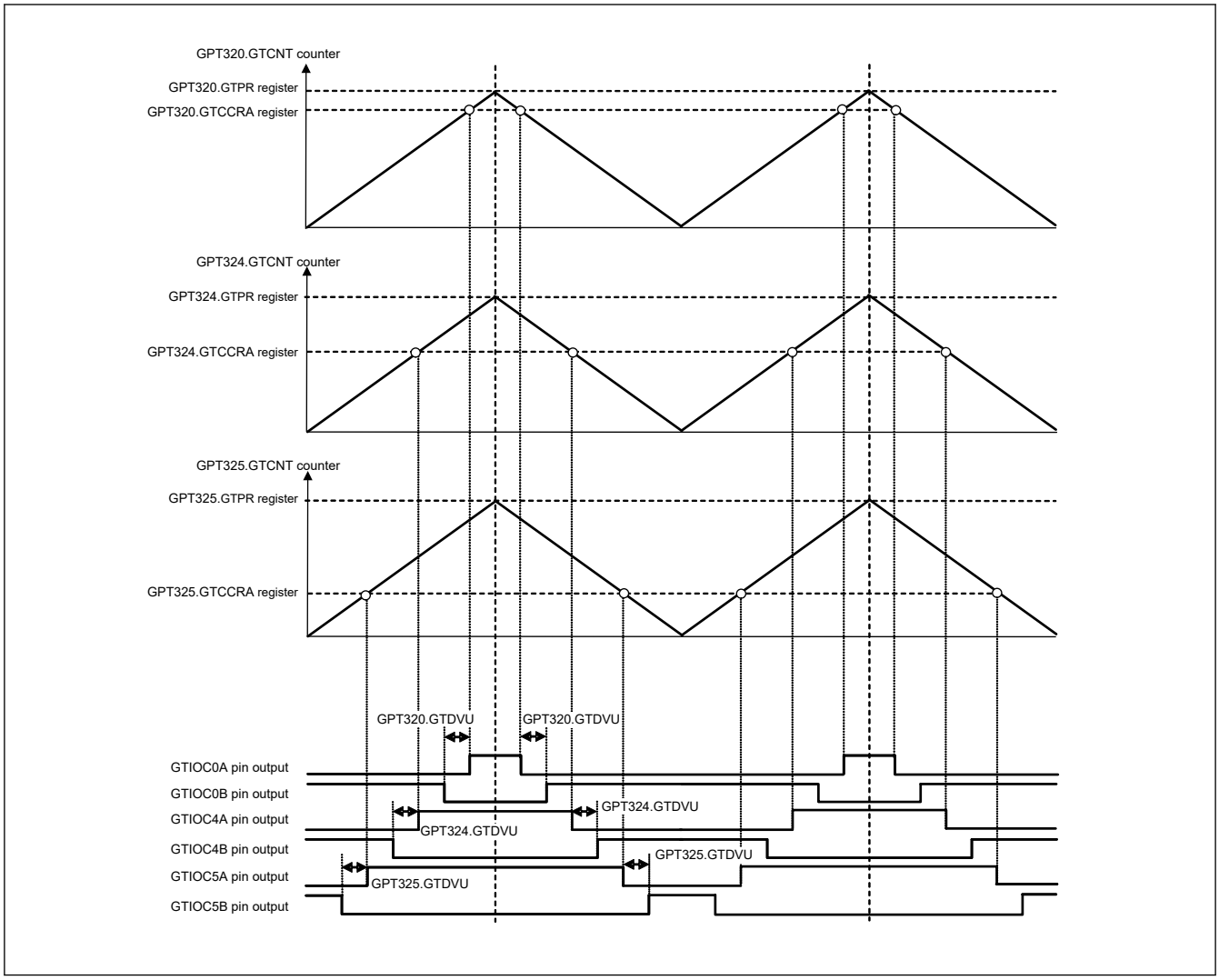


Figure 21.48 Example of 3-phase triangle-wave complementary PWM output with automatic dead time setting

(6) 3-phase asymmetric triangle-wave complementary PWM output with automatic dead time setting

Figure 21.49 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 3 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

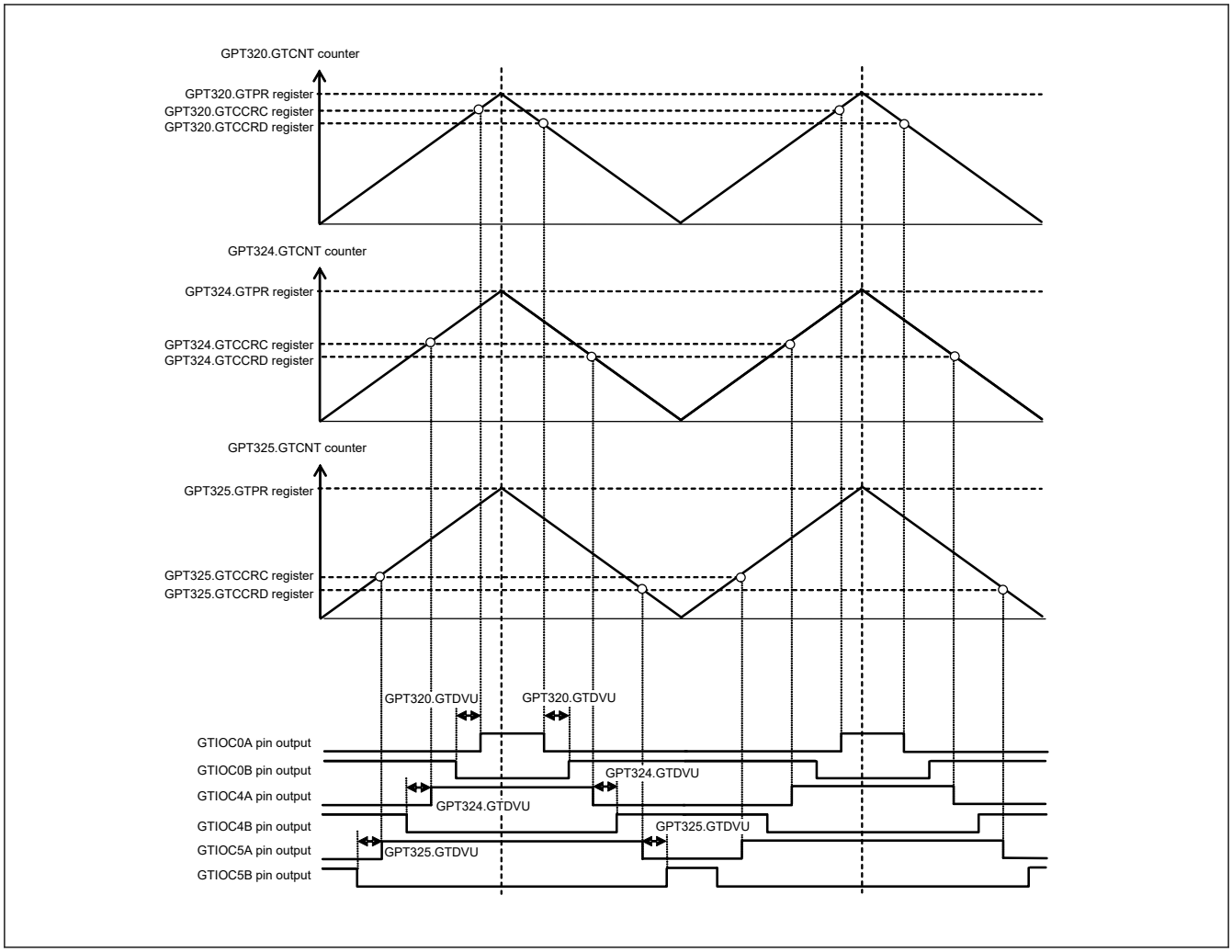


Figure 21.49 Example of 3-phase asymmetric triangle-wave complementary PWM output with automatic dead time setting

21.3.10 Period Count Function

By setting the GTPC register, the end of period can be counted.

The number of period to be counted should be set into the GTPC.PCNT counter when the GTPC.PCEN bit is 0. When the PCEN bit is 1, the PCNT counter can be read, but writing is disabled. When the PCEN bit is 1, down-counting is performed at the end of period. When the PCNT counter is 1 at the end of period, it becomes 0 and counting is stopped to finish the period count function. At that time, the GTST.PCF flag is set, and the period count function finish interrupt request GPTn_PC is generated. When the GTPC.ASTP bit is 1, the GTCNT counter is also stopped at the same time that the period count function is finished.

When the GTCNT counter is stopped while period count function is enabled, the PCNT counter keeps its value. When the GTCNT counter restarts counting and the PCEN bit is 1, the PCNT counter restarts down-counting from the hold value.

When the PCEN bit is changed from 0 to 1 while the PCNT counter is 0 and the ASTP bit is 1, the GTCNT counter is stopped at the count clock immediately after that.

When either GTSECR.SPCE bit or GTSECR.SPCD bit is set to 1, the PCEN bit in the channels set to 1 by the GTSECSR register is simultaneously set the value to enable or disable the period count function for multiple channels.

Figure 21.50 and Figure 21.51 show examples of PWM cycle count function.

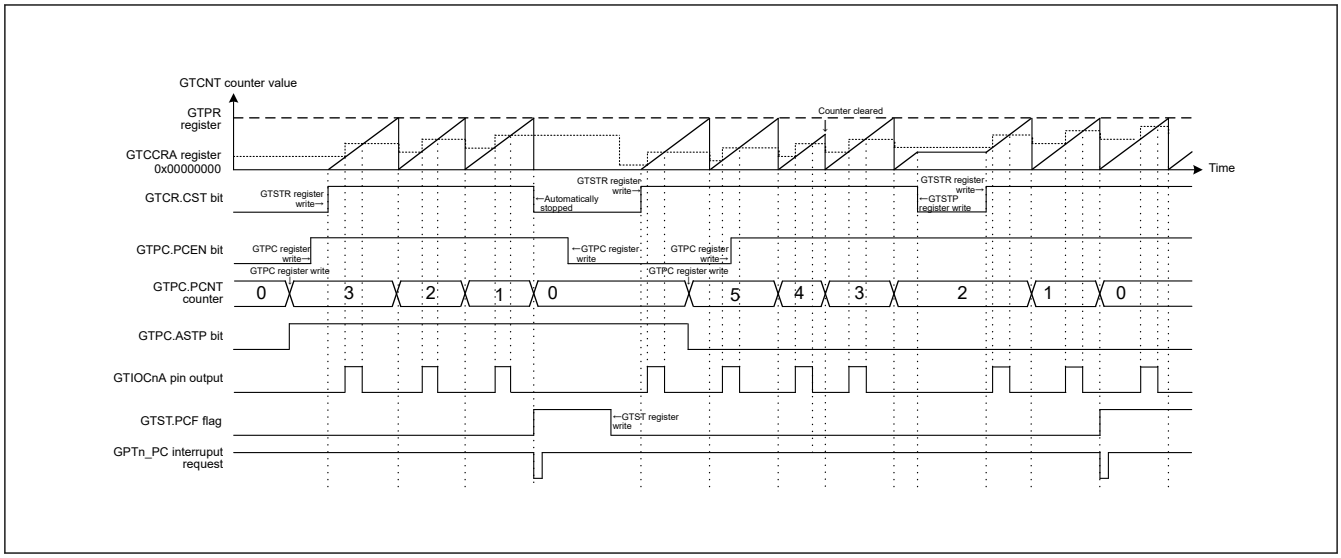
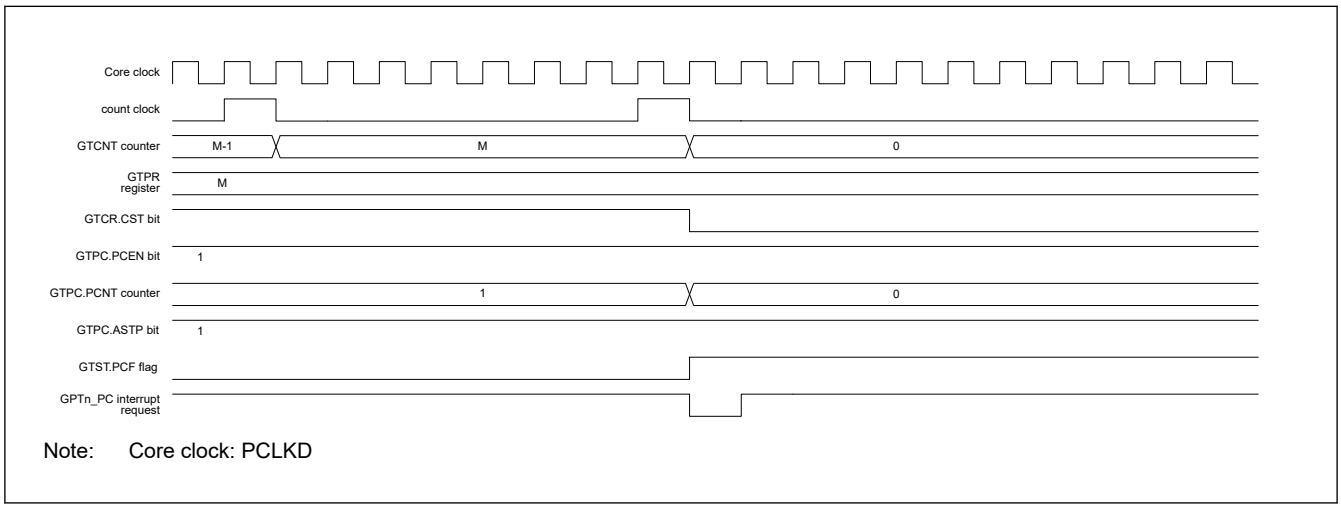


Figure 21.50 Example of PWM cycle count function (saw-wave one-shot pulse mode)



Note: Core clock: PCLKD

Figure 21.51 Example of the timing of operations for PWM cycle count function (saw-wave one-shot pulse mode, up-counting)

21.3.11 Phase Counting Function

The phase difference between the GTIOCnA and GTIOCnB pin (n = 0 to 13) inputs is detected and the associated GTCNT counts up or counts down. The detectable phase difference is available in any combination with the relationship between the edge and the level of GTIOCnA and GTIOCnB pin inputs being set in the GTUPSR and GTDNSR registers. For details on count operation, see section 21.3.1.1. Counter Operation.

Figure 21.52 to Figure 21.61 show an example of phase counting modes 1 to 5 operation when the GTIOCnA, GTIOCnB pins are used. Table 21.30 to Table 21.39 show conditions of up-counting or down-counting and list settings for the GTUPSR and GTDNSR registers which is corresponding to Figure 21.52 to Figure 21.61.

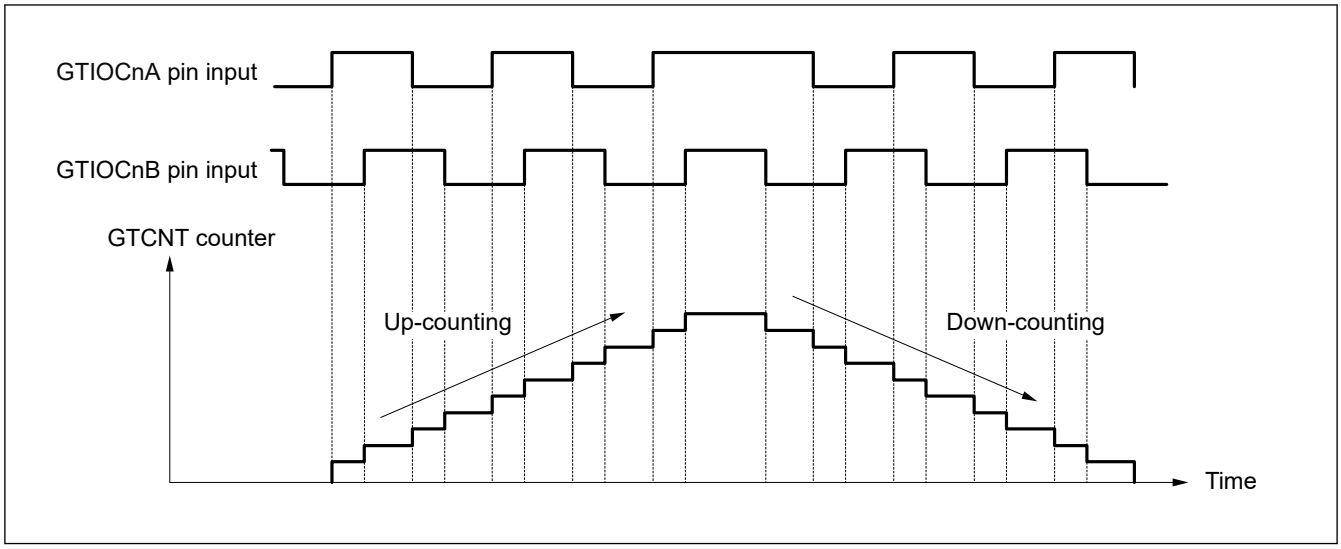


Figure 21.52 Example of phase counting mode 1

Table 21.30 Conditions of up-counting/down-counting in phase counting mode 1

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Up-counting	GTUPSR = 0x00006900 GTDNSR = 0x00009600
Low	↓		
↑	Low		
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High		
↓	Low		

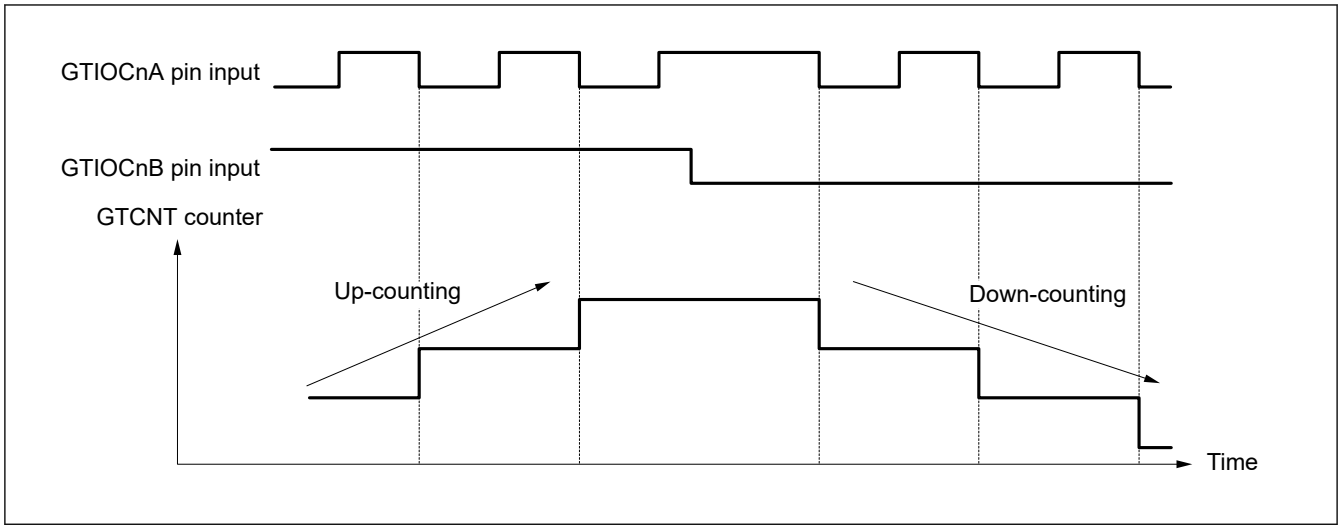


Figure 21.53 Example of phase counting mode 2 (A)

Table 21.31 Conditions of up-counting/down-counting in phase counting mode 2 (A)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Not counting	GTUPSR = 0x00000800 GTDNSR = 0x00000400
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Not counting	
Low	↑		
↑	High		
↓	Low	Down-counting	

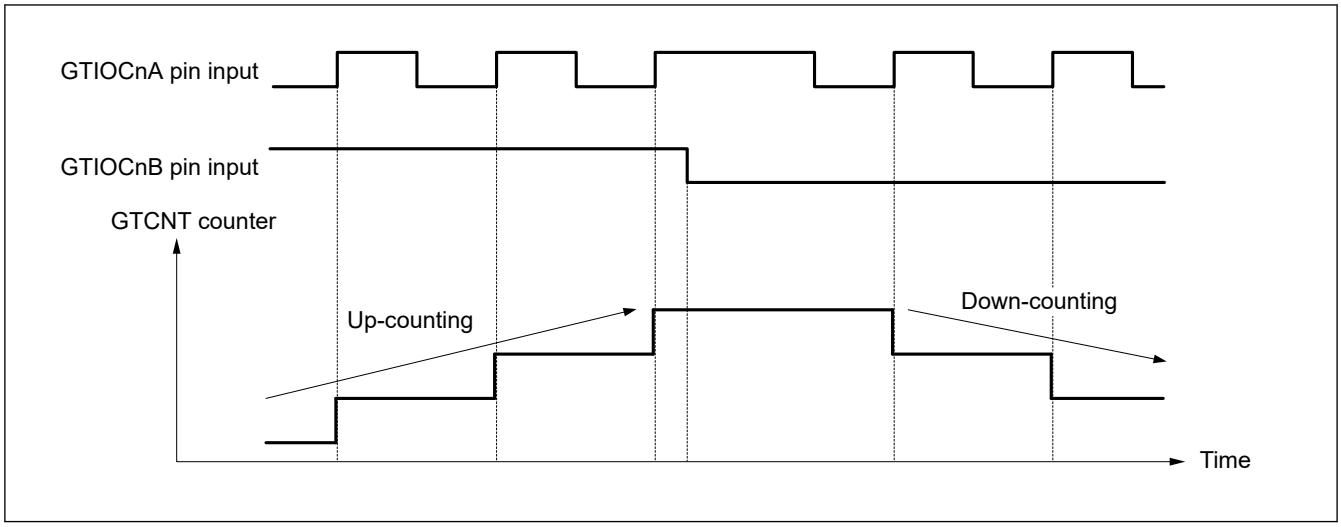


Figure 21.54 Example of phase counting mode 2 (B)

Table 21.32 Conditions of up-counting/down-counting in phase counting mode 2 (B)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Not counting	GTUPSR = 0x00000200 GTDNSR = 0x00000100
Low	↓		
↑	Low	Down-counting	
↓	High	Not counting	
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Not counting	

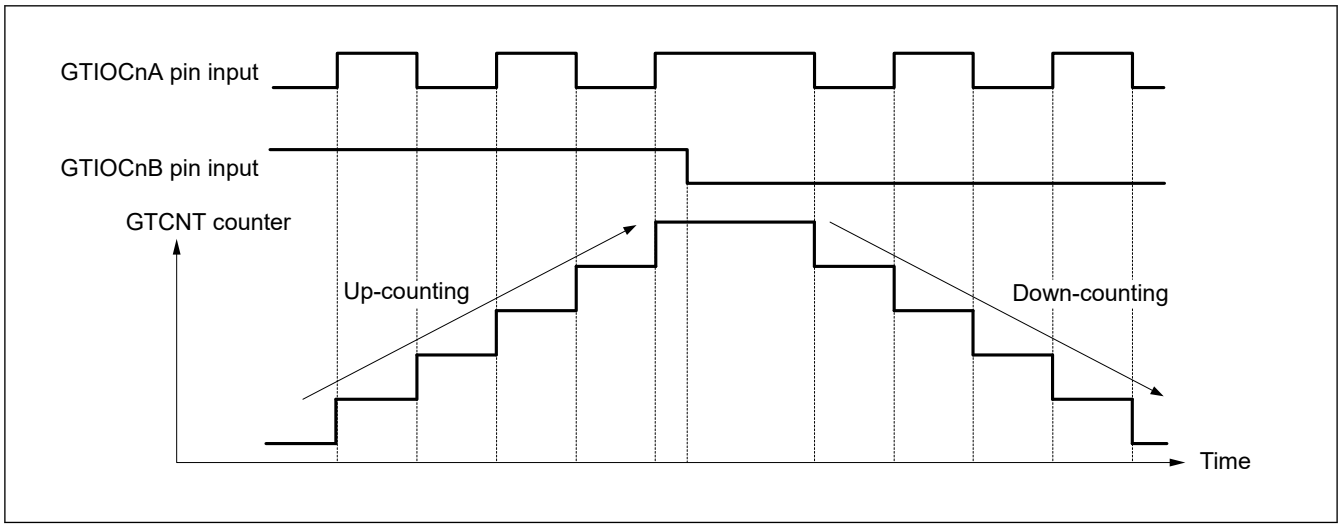


Figure 21.55 Example of phase counting mode 2 (C)

Table 21.33 Conditions of up-counting/down-counting in phase counting mode 2 (C)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Not counting	GTUPSR = 0x00000A00 GTDNSR = 0x00000500
Low	↓		
↑	Low	Down-counting	
↓	High	Up-counting	
High	↓	Not counting	
Low	↑		
↑	High	Up-counting	
↓	Low	Down-counting	

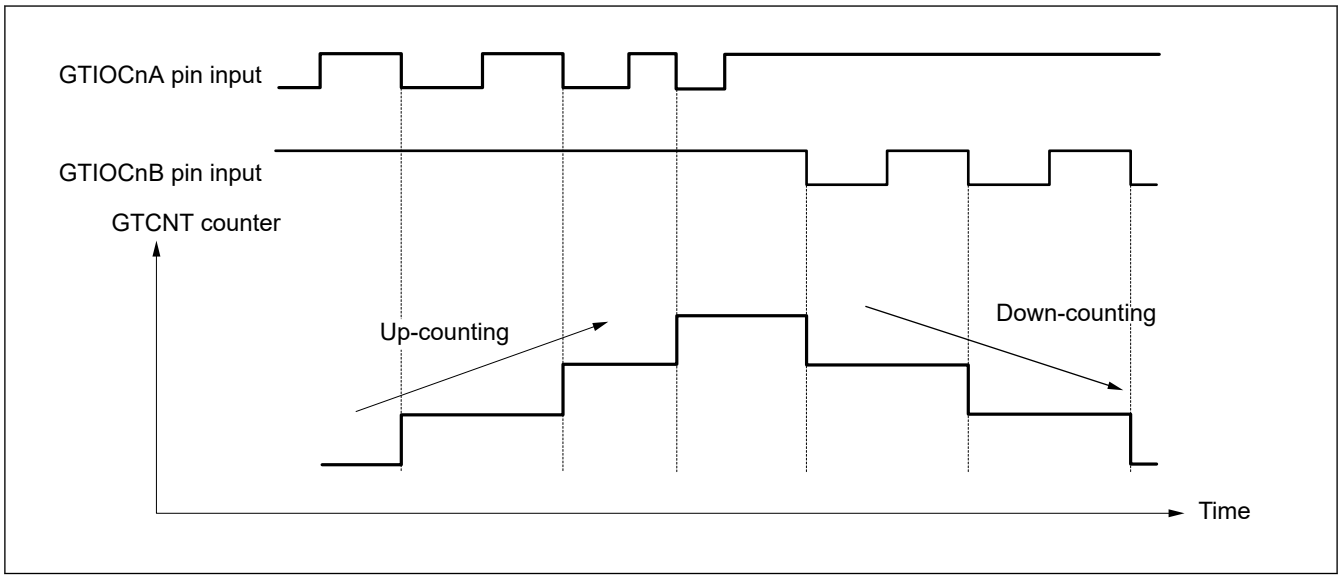


Figure 21.56 Example of phase counting mode 3 (A)

Table 21.34 Conditions of up-counting/down-counting in phase counting mode 3 (A)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Not counting	GTUPSR = 0x00000800 GTDNSR = 0x00008000
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Down-counting	
Low	↑	Not counting	
↑	High		
↓	Low		

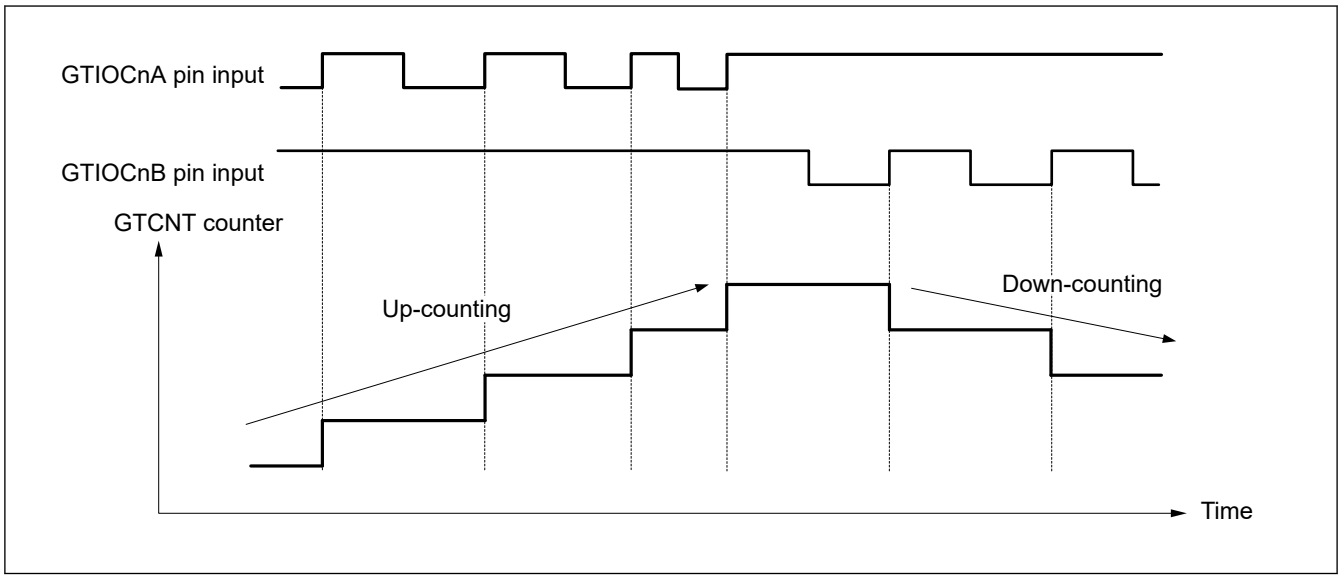


Figure 21.57 Example of phase counting mode 3 (B)

Table 21.35 Conditions of up-counting/down-counting in phase counting mode 3 (B)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Down-counting	GTUPSR = 0x00000200 GTDNSR = 0x00002000
Low	↓	Not counting	
↑	Low		
↓	High		
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Not counting	

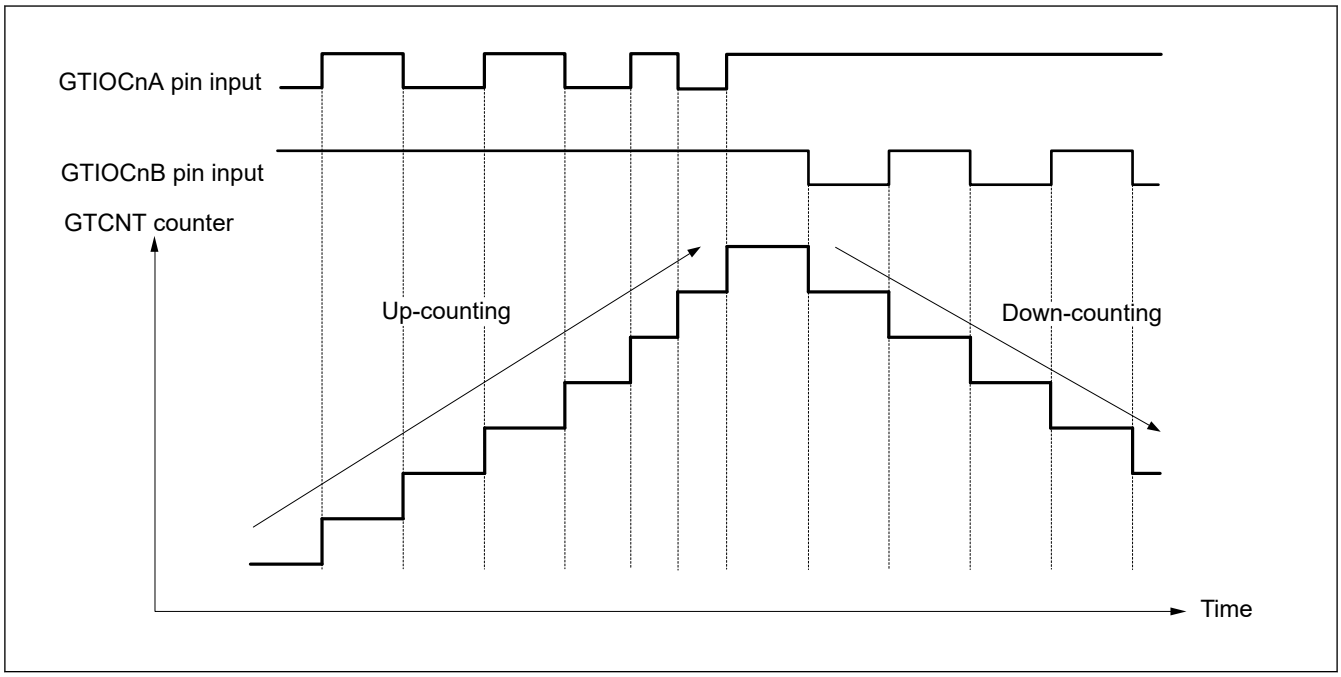


Figure 21.58 Example of phase counting mode 3 (C)

Table 21.36 Conditions of up-counting/down-counting in phase counting mode 3 (C)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Down-counting	GTUPSR = 0x00000A00 GTDNSR = 0x0000A000
Low	↓	Not counting	
↑	Low		
↓	High	Up-counting	
High	↓	Down-counting	
Low	↑	Not counting	
↑	High	Up-counting	
↓	Low	Not counting	

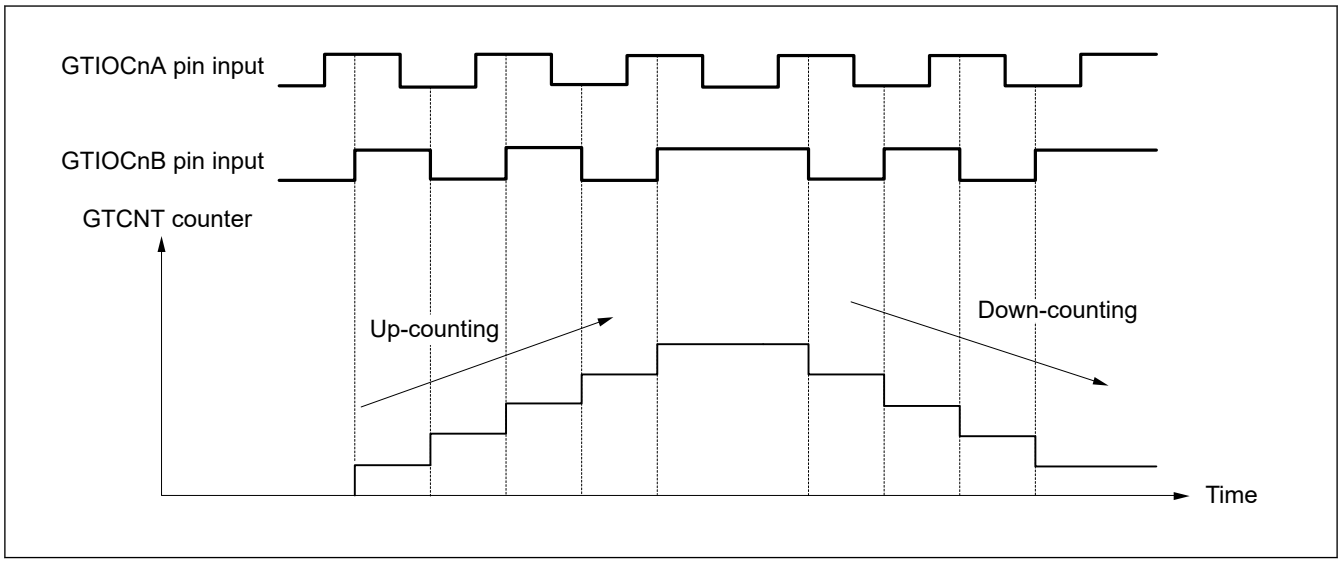


Figure 21.59 Example of phase counting mode 4

Table 21.37 Conditions of up-counting/down-counting in phase counting mode 4

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Up-counting	GTUPSR = 0x00006000 GTDNSR = 0x00009000
Low	↓		
↑	Low	Not counting	
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High	Not counting	
↓	Low		

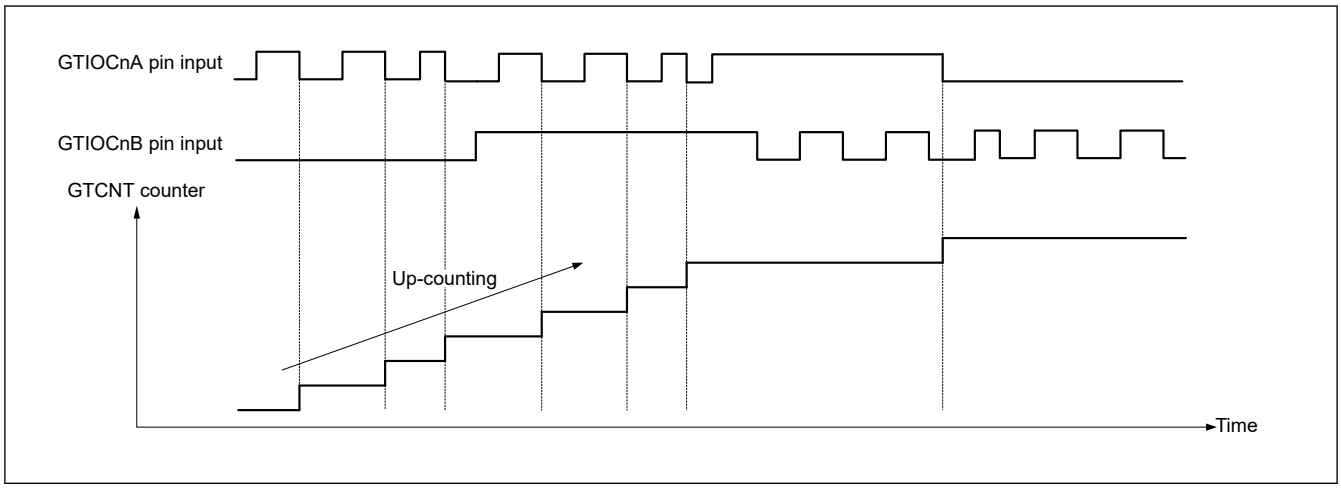


Figure 21.60 Example of phase counting mode 5 (A)

Table 21.38 Conditions of up-counting/down-counting in phase counting mode 5 (A)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x00000C00 GTDNSR = 0x00000000
Low			
	Low		
	High	Up-counting	
High		Not counting	
Low			
	High		
	Low	Up-counting	

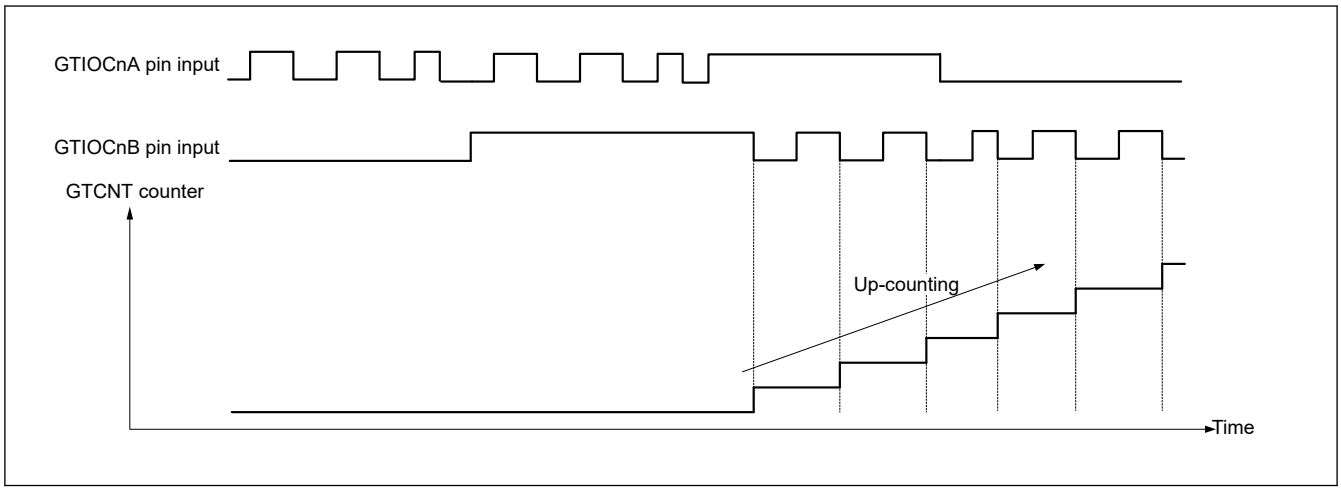


Figure 21.61 Example of phase counting mode 5 (B)

Table 21.39 Conditions of up-counting/down-counting in phase counting mode 5 (B)

: Rising edge

: Falling edge

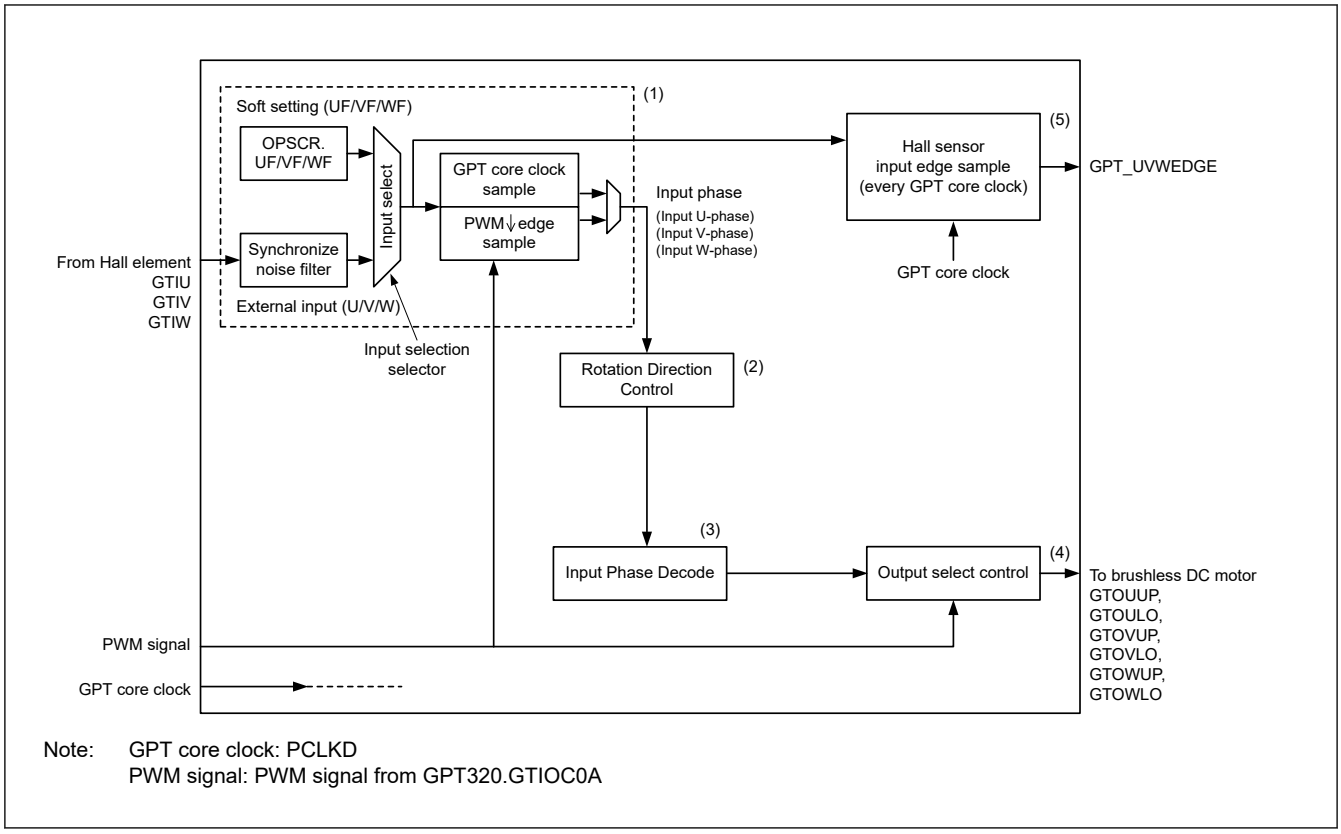
GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x0000C000 GTDNSR = 0x00000000
Low		Up-counting	
	Low	Not counting	
	High		
High		Up-counting	
Low		Not counting	
	High		
	Low		

21.3.12 Output Phase Switching (GPT_OPS)

GPT_OPS provides a function for easy control of brushless DC motor operation using the Output Phase Switching Control Register (OPSCR).

GPT_OPS outputs a PWM signal to be used for chopper control or level signal for each phase (U-positive phase/negative phase, V-positive phase/negative phase, W-positive phase/negative phase) of the 6-phase motor control. This function uses a soft setting value (OPSCR.UF, VF, WF) set by software or external signals detected by the Hall element, a PWM waveform of GPT320.GTIOC0A.

Figure 21.62 shows the conceptual diagram of GPT_OPS control flow.

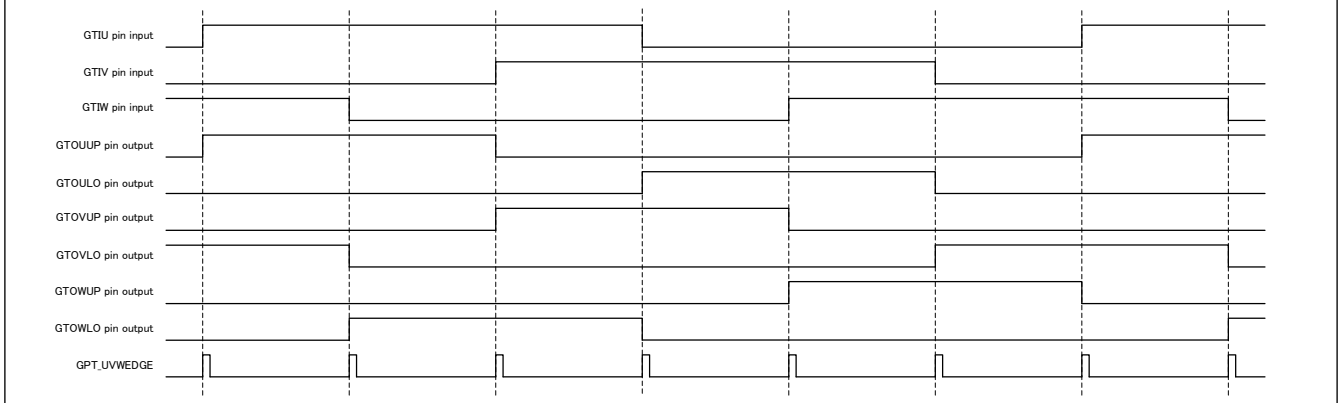


Note: GPT core clock: PCLKD
PWM signal: PWM signal from GPT320.GTIOC0A

Figure 21.62 Conceptual diagram of GPT_OPS control flow

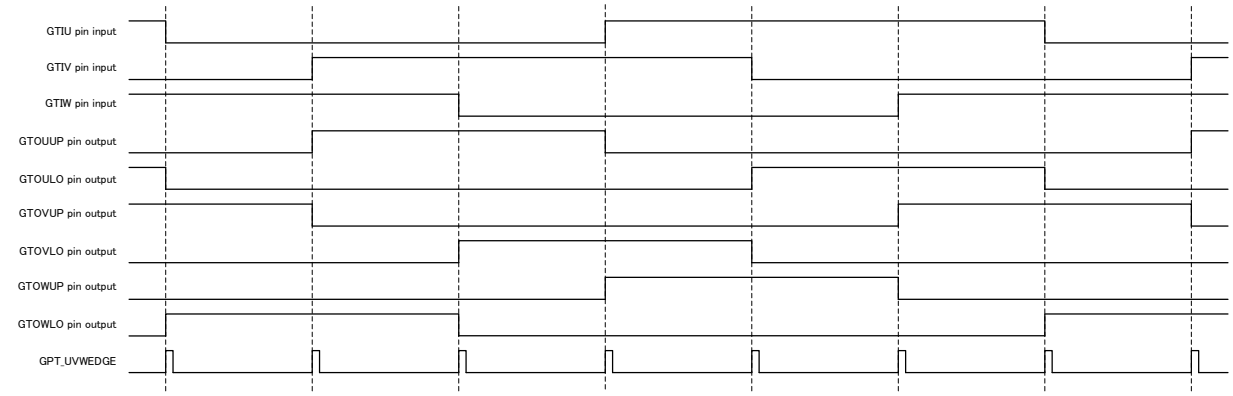
Figure 21.63 and Figure 21.64 shows a 6-phase level signals output example of a GPT_OPS operation.

The GPT_UVWEDGE signal in Figure 21.63 is the Hall sensor input edge that outputs to the ELC.



Note: Register setting: OPSCR.FB = 0, OPSCR.RV = 0, OPSCR.P = 0, OPSCR.N=0, OPSCR.INV = 0

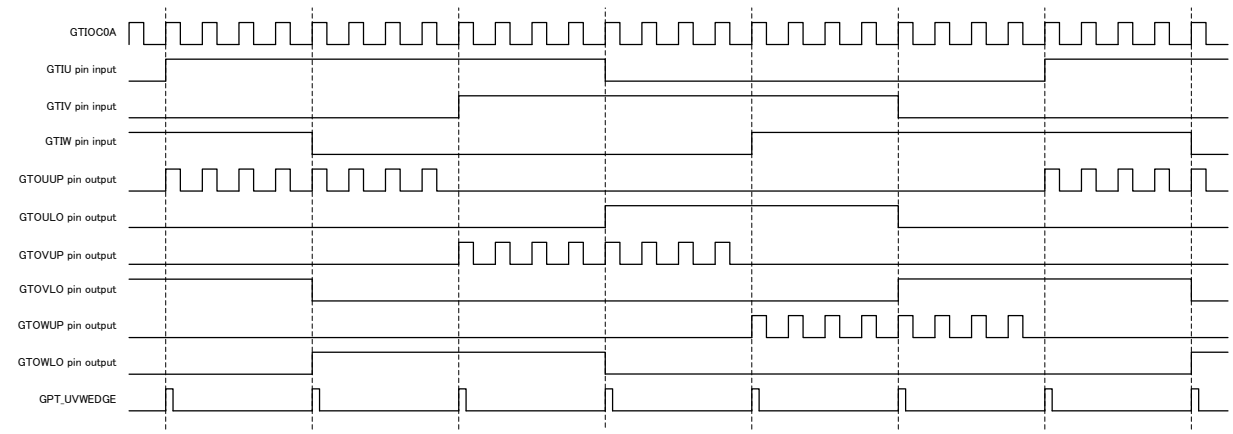
Figure 21.63 Example of GPT_OPS Level Output Operation (Forward Rotation)



Note: Register setting: OPSCR.FB = 0, OPSCR.RV = 1, OPSCR.P = 0, OPSCR.N = 0, OPSCR.INV = 0

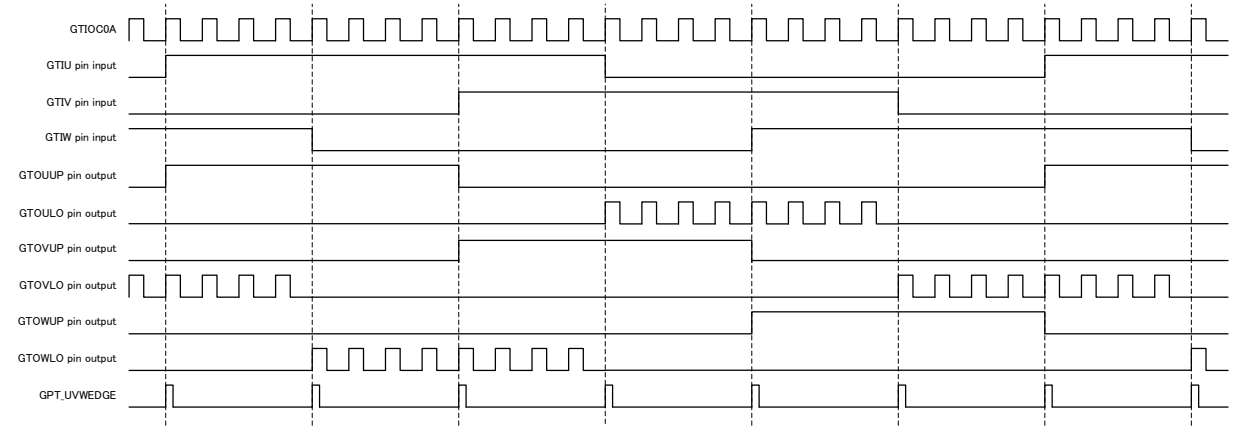
Figure 21.64 Example of GPT_OPS Level Output Operation (Reverse Rotation)

Figure 21.65 and Figure 21.66 shows a 6-phase PWM output example of a GPT_OPS operation with chopper control.



Note: Register setting: OPSCR.FB = 0, OPSCR.RV = 0, OPSCR.P = 1, OPSCR.N = 0, OPSCR.INV = 0

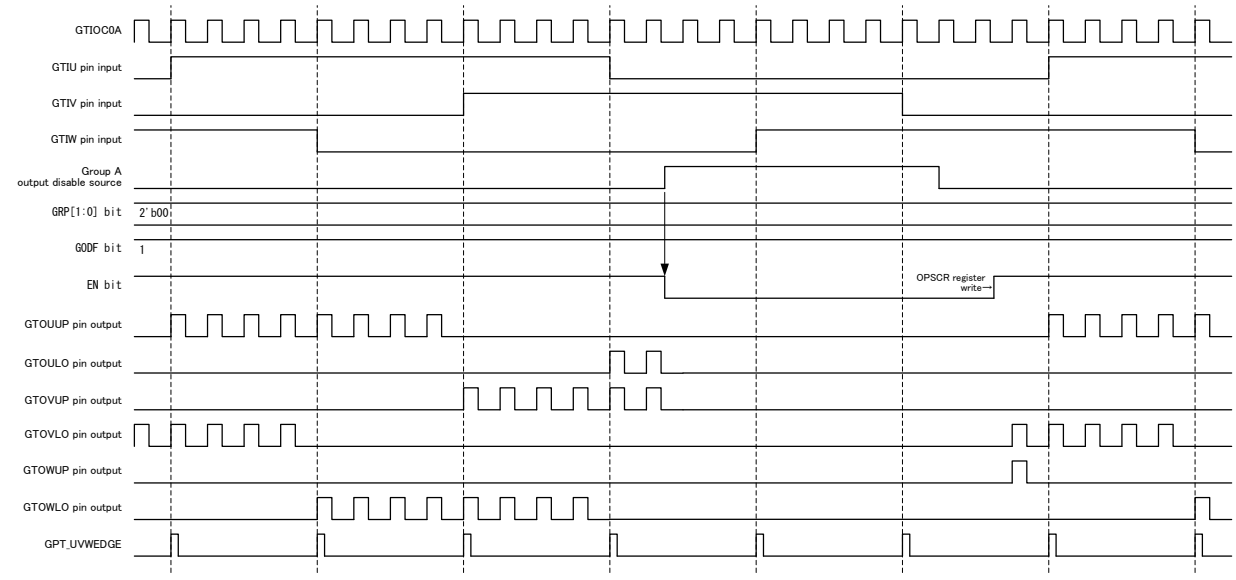
Figure 21.65 Example of GPT_OPS Chopped Output Operation (Positive Phase 120-degree)



Note: Register setting: OPSCR.FB = 0, OPSCR.RV = 0, OPSCR.P = 0, OPSCR.N = 1, OPSCR.INV = 0

Figure 21.66 Example of GPT_OPS Chopped Output Operation (Negative Phase 120-degree)

Figure 21.67 shows a 6-phase PWM output example of an output disable control operation.



Note: Register settings: OPSCR.FB = 0, OPSCR.RV = 0, OPSCR.P = 1, OPSCR.N = 1, OPSCR.INV = 0

Figure 21.67 Example of group output disable control operation

21.3.12.1 Input Selection and Synchronization of External Input Signal

In the GPT_OPS control flow conceptual diagram shown in Figure 21.62, (1) is a selection of input phase from the software settings and external input by the OPSCR.FB bit.

When OPSCR.FB bit is 0, select the external input. Enable the input signal after synchronization with the GPT core clock (PCLKD). After carrying out noise filtering (optional), set the external input to the input phase of PWM (PWM of GPT320.GTIOC0A) using falling edge sampling with OPSCR.ALIGN bit set to 1.

When OPSCR.FB bit is 1, select the soft setting (OPSCR.UF, VF, WF) with the value of the input phase of PWM (PWM of GPT320.GTIOC0A) using falling edge sampling with OPSCR.ALIGN bit set to 1.

When OPSCR.ALIGN bit is 0, GPT_OPS operates with the input phase of PCLKD synchronization with either OPSCR.FB bit set to 0 or OPSCR.FB bit set to 1. However, there are cases where the PWM pulse width of the output U/V/W phases (PWM output mode) of switch timing (just before/just after) is shortened.

Table 21.40 shows the input selection process and setting of associated OPSCR bits.

Table 21.40 Input selection processing method

Register OPSCR		Selection of input phase sampling method (U/V/W-phase)	Synchronization input/output selection process (GPT_OPS internal node name)
FB bit	ALIGN bit		
0	1	External Input at PWM Falling Edge Sampling (PCLKD synchronization + falling edge sample)	Input Phase Input U-Phase (gtu_sync) Input V-Phase (gtv_sync) Input W-Phase (gtw_sync)
	0	External Input at PCLKD Synchronization Output (PCLKD synchronization + through mode)	
1	1	Software Settings at PWM Falling Edge Sampling (OPSCR.UF, VF, WF of falling edge sample)	
	0	Software Setting Value Selection (= OPSCR.UF/VF/WF value) (= PCLKD synchronization)	

21.3.12.2 Input Sampling

The OPSCR.U, V, W bits indicate the PCLKD sampling results of the input selected in the OPSCR.FB bit.

When OPSCR.FB bit is 0 and after synchronization with the GPT core clock (PCLKD) and noise filtering (optional), OPSCR.U, V, W bits indicate the sampling results of the external input. When OPSCR.FB bit is 1, OPSCR.U, V, W bits are the value (OPSCR.UF, VF, WF) of the soft setting.

21.3.12.3 Input Phase Decode

In the GPT_OPS control flow conceptual diagram shown in Figure 21.62, (3) enables the 6-phase signals by decoding the input phase selected in the OPSCR.FB bit.

Table 21.41 shows the decode table of input phase when OPSCR.RV bit is 0.

Table 21.41 Decode table of input phase (OPSCR.RV = 0)

Input phase (U/V/W) (GPT_OPS internal node name)			6-phase enable {U/V/W (Up/Lo)} by decoding input phase (GPT_OPS internal node name)					
Input U-Phase	Input V-Phase	Input W-Phase	U-phase (Up)	U-phase (Lo)	V-phase (Up)	V-phase (Lo)	W-phase (Up)	W-phase (Lo)
(gtu_sync)	(gtv_sync)	(gtw_sync)	(gtuup_en)	(gtulo_en)	(gtvup_en)	(gtvlo_en)	(gtwup_en)	(gtwlo_en)
1	0	1	1	0	0	1	0	0
1	0	0	1	0	0	0	0	1
1	1	0	0	0	1	0	0	1
0	1	0	0	1	1	0	0	0
0	1	1	0	1	0	0	1	0
0	0	1	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0

Table 21.42 Decode table of input phase (OPSCR.RV = 1)

Input phase (U/V/W) (GPT_OPS internal node name)			6-phase enable {U/V/W (Up/Lo)} by decoding input phase (GPT_OPS internal node name)					
Input U-Phase	Input V-Phase	Input W-Phase	U-phase (Up)	U-phase (Lo)	V-phase (Up)	V-phase (Lo)	W-phase (Up)	W-phase (Lo)
(gtu_sync)	(gtv_sync)	(gtw_sync)	(gtuup_en)	(gtulo_en)	(gtvup_en)	(gtvlo_en)	(gtwup_en)	(gtwlo_en)
1	0	1	0	1	1	0	0	0
1	0	0	0	1	0	0	1	0
1	1	0	0	0	0	1	1	0
0	1	0	1	0	0	1	0	0
0	1	1	1	0	0	0	0	1
0	0	1	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0

21.3.12.4 Rotation Direction Control

In the GPT_OPS control flow conceptual diagram shown in Figure 21.62, (2) controls the direction of rotation of a 3-phase motor using the OPSCR.RV bit.

When the rotation direction is reverse (RV bit = 1), the input phase is inverted.

Table 21.43 shows the assigned output phases based on the OPSCR.RV bit setting (before and after rotation direction control).

Table 21.43 Rotation Direction Control Method

Reversal of Direction of Rotation Using Output Phases as Specified in OPSCR Register	Output of Rotation Direction Control [U/V/W (Positive/Negative)] (GPT_OPS Internal Node Name after Control)					
OPSCR.RV bit	(gtuup_ren)	(gtulo_ren)	(gtvup_ren)	(gtvlo_ren)	(gtwup_ren)	(gtwlo_ren)
0	U-phase (Up) (gtuup_en)	U-phase (Lo) (gtulo_en)	V-phase (Up) (gtvup_en)	V-phase (Lo) (gtvlo_en)	W-phase (Up) (gtwup_en)	W-phase (Lo) (gtwlo_en)
1	U-phase (Up) (gtuup_en)	U-phase (Lo) (gtulo_en)	W-phase (Up) (gtwup_en)	W-phase (Lo) (gtwlo_en)	V-phase (Up) (gtvup_en)	V-phase (Lo) (gtvlo_en)

21.3.12.5 Output Selection Control

In the GPT_OPS control flow conceptual diagram in Figure 21.62, (4) represents the selection of the output waveform by setting the OPSCR register bit.

For output selection, the following bits are relevant:

- The OPSCR.EN bit controls whether to output the 6-phase output, or to stop
- The OPSCR.P and OPSCR.N bits can select from the level signal or PWM signal (chopper output) for the output phase
- The polarity of the output phase can be set to a positive logic or negative logic by the OPSCR. INV bit.

Table 21.44 and Table 21.45 show the output selection control method using the OPSCR register bit.

Table 21.44 Output selection control method (positive phase)

Enable-phase output control	Positive-phase output (P) control	Invert-phase output control	Output port name (positive phase = up) (output selection internal node allocation)	
OPSCR.EN	OPSCR.P	OPSCR.INV	GTOUUP GTOVUP GTOWUP	Mode
0	x	x	0	Output Stop (External pin: Hi-Z) GPT_OPS → 0 output
1	0	0	Level signal (gtuup_ren) (gtvup_ren) (gtwup_ren)	Level Output Mode (Positive phase) (Positive logic)
1	0	1	Level signal (~gtuup_ren) (~gtvup_ren) (~gtwup_ren)	Level Output Mode (Positive phase) (Negative logic)
1	1	0	PWM signal (PWM & gtuup_ren) (PWM & gtvup_ren) (PWM & gtwup_ren)	PWM Output Mode (Positive phase) (Positive logic)
1	1	1	PWM signal (~(PWM & gtuup_ren)) (~(PWM & gtvup_ren)) (~(PWM & gtwup_ren))	PWM Output Mode (Positive phase) (Negative logic)

Table 21.45 Output selection control method (negative phase)

Enable-phase output control	Positive-phase output (N) control	Invert-phase output control	Output port name (negative phase = Lo) (output selection internal node allocation)	
OPSCR.EN	OPSCR.N	OPSCR.INV	GTOULO GTOVLO GTOWLO	Mode
0	x	x	0	Output Stop (External pin: Hi-Z) GPT_OPS → 0 output
1	0	0	Level signal (gtulo_ren) (gtvlo_ren) (gtwlo_ren)	Level Output Mode (Negative phase) (Positive logic)
1	0	1	Level signal (~gtulo_ren) (~gtvlo_ren) (~gtwlo_ren)	Level Output Mode (Negative phase) (Negative logic)
1	1	0	PWM signal (PWM & gtulo_ren) (PWM & gtvlo_ren) (PWM & gtwlo_ren)	PWM Output Mode (Negative phase) (Positive logic)
1	1	1	PWM signal (~(PWM & gtulo_ren)) (~(PWM & gtvlo_ren)) (~(PWM & gtwlo_ren))	PWM Output Mode (Negative phase) (Negative logic)

21.3.12.6 Output Selection Control (Group Output Disable Function)

When OPSCR.GODF is 1 and the signal value selected by the OPSCR.GRP bit is high (output disable request), the group output-disable function asynchronously sets the output to Hi-Z. When an output-disable request is generated, the OPSCR.EN bit is cleared to 0. For the return, set the OPSCR.EN bit to 1 after clearing the output disable request by software.

To ensure output-disable control, use the POEG_GROUPn (n = A to D) interrupt to clear the flag in the POE or check that the OPSCR.EN bit is 0 and then clear the flag. For an example of the operation for group output disable control, see Figure 21.67.

21.3.12.7 Event Link Controller (ELC) Output

In the GPT_OPS control flow conceptual diagram shown in Figure 21.62, (5) outputs the Hall sensor input signal edge to the ELC.

The Hall sensor input edge signal is the logical OR of the rising and falling edge signals of each U-phase/V-phase/W-phase input sampled at PCLKD. That is, if the high period of each of the U-phase/V-phase/W-phase of the input phase is short in duration, the Hall sensor edge input signal is not output at that time.

When the OPSCR.FB bit is 0, the Hall sensor input edge signal is the logical OR of the edge signals of the external input phase sampled at PCLKD.

When OPSCR.FB bit is 1, the Hall sensor input edge signal is the logical OR of the edge of the soft setting (OPSCR.UF, VF, WF) sampled at PCLKD.

See Figure 21.63 to Figure 21.67 for examples of the output signal to the ELC.

21.3.12.8 GPT_OPS Start Operation Setting Flow

Table 21.46 Example setting of GPT_OPS start operation (1 of 2)

No.	Step Name	Description
1	GPT32n operation mode setting	GPT320.GTIOC0A set the PWM output operation mode of the saw-wave or triangle-wave. For details, see section 21.3.3. PWM Output Operating Mode.
2	Counting of GPT32n	Start the count operation of GPT32n, and outputs a PWM waveform.

Table 21.46 Example setting of GPT_OPS start operation (2 of 2)

No.	Step Name	Description
3	GPT_OPS input data set (only software setting is selected)	Set software setting to OPSCR.UF, VF, and WF bits.
4	Noise filter settings of GPT_OPS external input (only external input is selected)	When using a noise filter, set the sampling clock of the noise filter by OPSCR.NFCS[1:0] bits. Then the noise filter is enabled if OPSCR.NFEN = 1.
5	GPT_OPS input phase selection setting/input phase alignment setting	Select the input phase from the external input or software setting by OPSCR.FB bit. Select the alignment of the input phase by OPSCR.ALIGN bit.
6	Setting the GPT_OPS output phase	Set the level output/PWM output of the positive/negative phase output by OPSCR.P/OPSCR.N bit. Set the positive logic/negative logic of the output phase by OPSCR.INV bit. Set the rotation direction by OPSCR.RV bit
7	GPT_OPS setting the group output disable function	Set the selection of output disable source by OPSCR.GRP bit. Perform the setting of on/off of the group output disable function by OPSCR.GODF bit.
8	GPT_OPS Working	Setting the OPSCR.EN = 1 outputs the 6-phase output to drive the brushless DC motor from the GPT_OPS.

21.3.13 Inter Channel Logical Operation Function

The logical operation function between compare match outputs can be performed.

Figure 21.68 shows the block diagram of inter channel logical operation.

To prevent hazard to the GPT output, the signal after logical operation is latched with PCLKD. After latching, the output disable control is performed.

When the logical operation function which causes the delay of 1 PCLKD is selected, the output enable signal is also delayed with 1 PCLKD and input to the output disable control.

When the same signal (C = A or D = B) to operate logical function AND, OR, EXOR and NOR is selected, C or D is treated as 1. In the case of GTIOCnA pin output, when A of same channel is selected for C, the result of AND is A, the result of OR is 1, the result of EXOR is NOT A, and the result of NOR is 0.

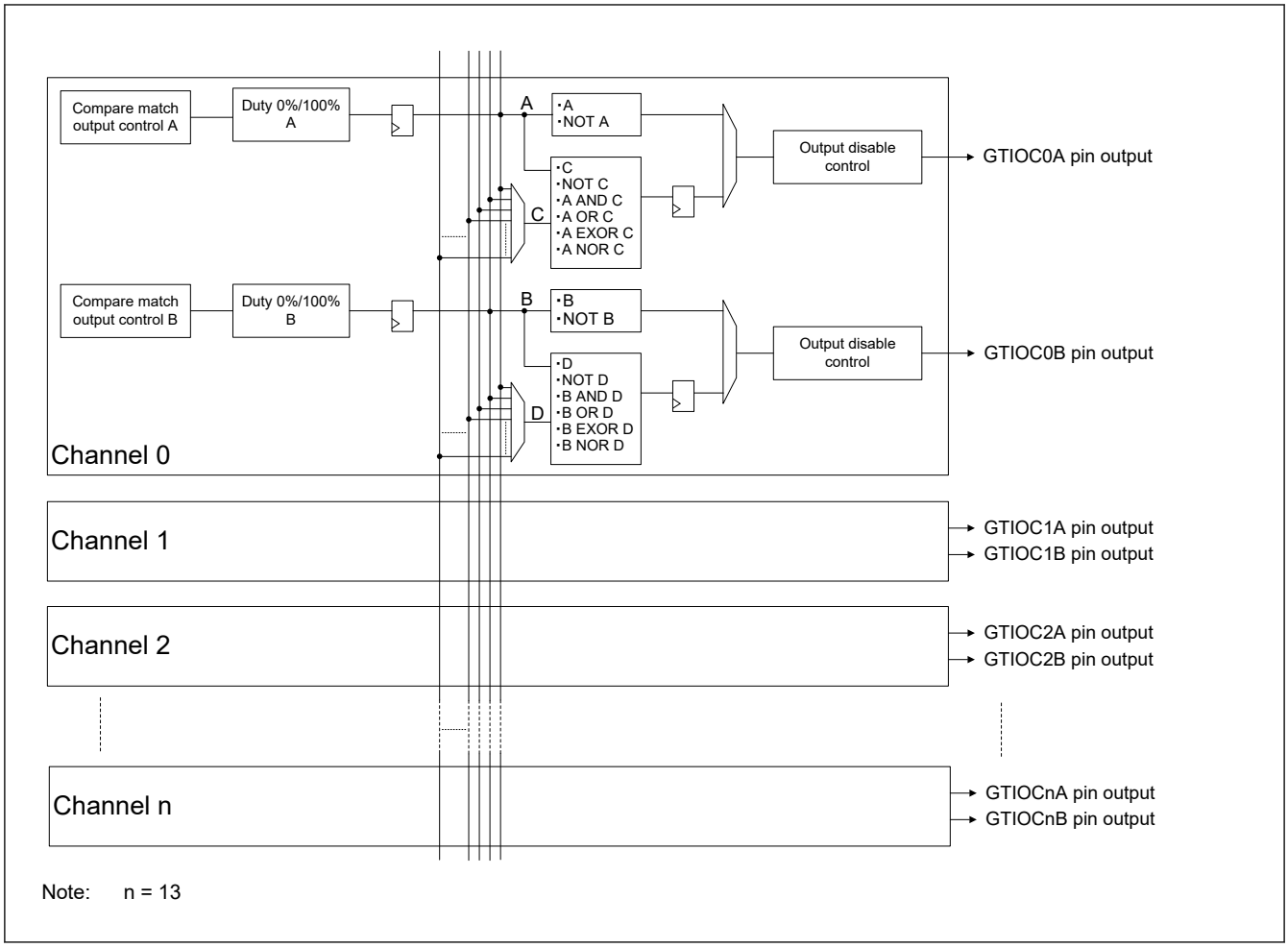


Figure 21.68 Block Diagram of Inter Channel Logical Operation

Figure 21.69 shows an example of inter channel logical operation.

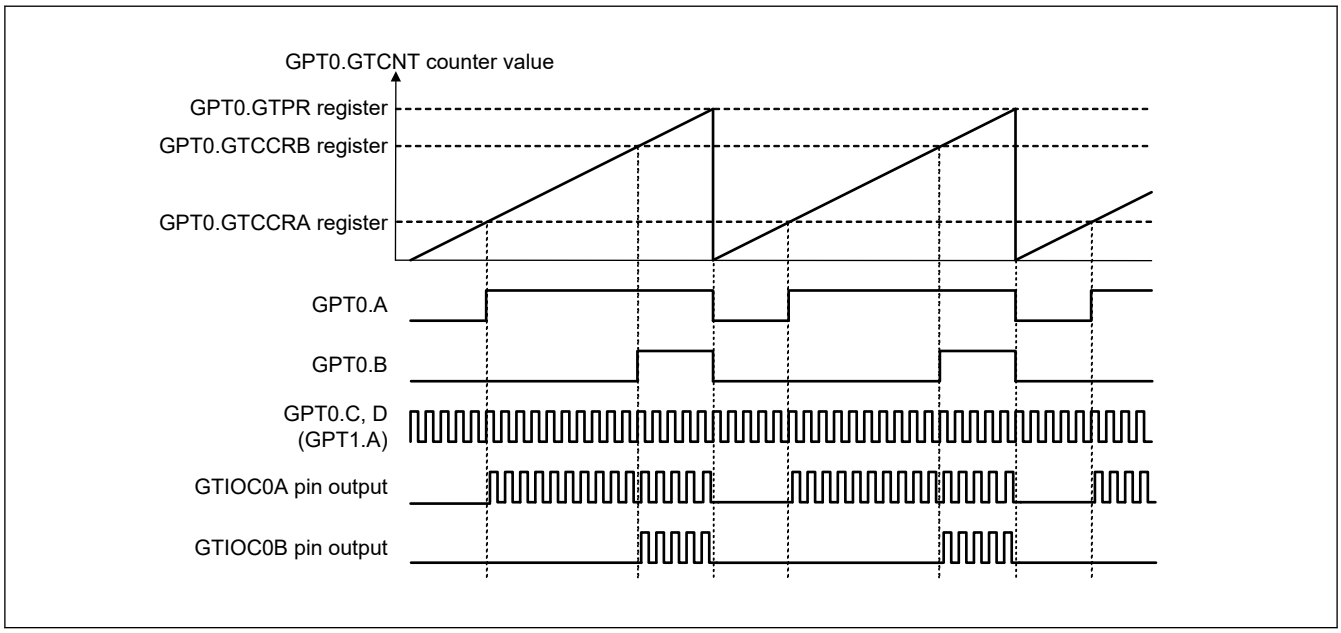


Figure 21.69 Example of Inter Channel Logical Operation

21.4 Interrupt Sources

21.4.1 Interrupt Sources

The GPT provides the following interrupt sources:

- GTCCR input capture/compare match
- GTCNT counter overflow (GTPR compare match)/underflow.
- Period count function finish

Each interrupt source has its own status flag. When an interrupt source signal is generated, the associated status flag in GTST is set to 1, and an interrupt request is generated. The associated status flag in GTST can be cleared by writing 0. If flag set and flag clear occur at the same time, flag clear takes priority over flag set. These flags are automatically updated by the internal state. The Interrupt Controller Unit can change the relative channel priorities. However, the priority within a channel is fixed. For details, see section 13, Interrupt Controller Unit (ICU).

Table 21.47 lists the GPT interrupt sources.

Table 21.47 Interrupt sources

Channel	Name	Interrupt source	Interrupt flag	DTC activation
n = 0 to 7	GPTn_CCMPA	GPT32n.GTCCRA input capture/compare match	GTST[0] (TCFA)	Possible
	GPTn_CCMPB	GPT32n.GTCCRB input capture/compare match	GTST[1] (TCFB)	Possible
	GPTn_CMPC	GPT32n.GTCCRC compare match	GTST[2] (TCFC)	Possible
	GPTn_CMPD	GPT32n.GTCCRD compare match	GTST[3] (TCFD)	Possible
	GPTn_CMPE	GPT32n.GTCCRE compare match	GTST[4] (TCFE)	Possible
	GPTn_CMPF	GPT32n.GTCCRF compare match	GTST[5] (TCFF)	Possible
	GPTn_OVF	GPT32n.GTCNT overflow (GPT32n.GTPR compare match)	GTST[6] (TCFPO)	Possible
	GPTn_UDF	GPT32n.GTCNT underflow	GTST[7] (TCFPU)	Possible
	GPTn_PC	Period count function finish (n = 0 to 3)	GTST[31] (PCF)	Possible
n = 8 to 13	GPTn_CCMPA	GPT16n.GTCCRA input capture/compare match	GTST[0] (TCFA)	Possible
	GPTn_CCMPB	GPT16n.GTCCRB input capture/compare match	GTST[1] (TCFB)	Possible
	GPTn_CMPC	GPT16n.GTCCRC compare match	GTST[2] (TCFC)	Possible
	GPTn_CMPD	GPT16n.GTCCRD compare match	GTST[3] (TCFD)	Possible
	GPTn_CMPE	GPT16n.GTCCRE compare match	GTST[4] (TCFE)	Possible
	GPTn_CMPF	GPT16n.GTCCRF compare match	GTST[5] (TCFF)	Possible
	GPTn_OVF	GPT16n.GTCNT overflow (GPT32n.GTPR compare match)	GTST[6] (TCFPO)	Possible
	GPTn_UDF	GPT16n.GTCNT underflow	GTST[7] (TCFPU)	Possible
	GPTn_PC	Period count function finish (n = 8 to 10)	GTST[31] (PCF)	Possible

(1) GPTn_CCMPA interrupt (n = 0 to 13)

An interrupt request is generated under the following conditions:

- When the GTCCRA register functions as a compare match register, the GTCNT counter value matches with the GTCCRA register
- When the GTCCRA register functions as an input capture register, the input-capture signal causes transfer of the GTCNT counter value to the GTCCRA register.

(2) GPTn_CCMPB interrupt (n = 0 to 13)

An interrupt request is generated under the following conditions:

- When the GTCCRB register functions as a compare match register, the GTCNT counter value matches with the GTCCRB register

- When the GTCCRB register functions as an input capture register, the input-capture signal causes transfer of the GTCNT counter value to the GTCCRB register.

(3) GPTn_CMPC interrupt (n = 0 to 13)

An interrupt request is generated under the following condition:

- When the GTCCRC register functions as a compare match register, the GTCNT counter value matches with the GTCCRC register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 01b, 10b, 11b (buffer operation with the GTCCRC register).

(4) GPTn_CMPD interrupt (n = 0 to 13)

An interrupt request is generated under the following condition:

- When the GTCCRD register functions as a compare match register, the GTCNT counter value matches with the GTCCRD register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 10b, 11b (buffer operation with the GTCCRD register).

(5) GPTn_CMPE interrupt (n = 0 to 13)

An interrupt request is generated under the following condition:

- When the GTCCRE register functions as a compare match register, the GTCNT counter value matches with the GTCCRE register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 01b, 10b, 11b (buffer operation with the GTCCRE register).

(6) GPTn_CMPF interrupt (n = 0 to 13)

An interrupt request is generated under the following condition:

- When the GTCCRF register functions as a compare match register, the GTCNT counter value matches with the GTCCRF register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 10b, 11b (buffer operation with the GTCCRF register).

(7) GPTn_OVF interrupt (n = 0 to 13)

An interrupt request is generated in the following conditions:

- In saw-wave mode, interrupt requests are enabled at overflows (when the GTCNT counter value changes from GTPR to 0 during up-counting)
- In triangle-wave mode, interrupt requests are enabled at crests (the GTCNT changes from GTPR to GTPR-1)

- In counting by hardware sources, an overflow (GTCNT changes from GTPR to 0 in up count) has occurred.

(8) GPTn_UDF interrupt (n = 0 to 13)

An interrupt request is generated in the following conditions.

- In saw-wave mode, interrupt requests are enabled at underflows (when the GTCNT counter value changes from 0 to GTPR during down-counting)
- In triangle-wave mode, interrupt requests are enabled at troughs (the GTCNT changes from 0 to 1)
- In counting by hardware sources, underflow (GTCNT changes from 0 to GTPR in down count) has occurred.

About Interrupt signals and interrupt status flags, see section 21.2.16. GTST : General PWM Timer Status Register.

(9) GPTn_PC Interrupt (n = 0 to 3, 8 to 10)

When the GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 1, an interrupt request is generated at the end of cycle.

21.4.2 DMAC and DTC Activation

The DMAC and DTC can be activated by the interrupt in each channel. For details, see section 13, Interrupt Controller Unit (ICU), section 16, DMA Controller (DMAC), and section 17, Data Transfer Controller (DTC).

21.5 A/D Conversion Start Request

The A/D conversion start request can be issued at a compare match between the GTCNT counter and the GTADTRA or GTADTRB register. Up-counting only, down-counting only, or both up-counting and down-counting can be specified by setting the GTINTAD register.

During event count operation, the A/D conversion start request can not be generated.

The A/D conversion start request is directly output to ADC12.

The GTADTRA and GTADTRB registers each has two buffer registers. Buffer operation with the GTADTRA register used together with the GTADTBRA and GTADTDBRA registers, and buffer operation with the GTADTRB register used together with the GTADTBRB and GTADTDBRB registers can be performed.

The timing of the generation of requests to start A/D conversion can be monitored by an external pin. When the A/D conversion start request signal to be monitored is selected in the GTADSMR.ADSMSk bit (k = 0, 1) and when the output is enabled in the ADSMENk bit, a signal is output synchronized with a cycle frame of the timer used to generate the A/D conversion start request signal, of which the output is driven high at the generation of the A/D conversion start request signal by the GTADSMk pin, or at the end of the cycle of which the output is driven low. When a signal to request the start of A/D conversion is generated at the end of the cycle, the generation of this signal has priority in terms of monitoring output and the output remains at the high level till the end of the next cycle. The registers (GTADTRA and GTADTRB) that are sources of generating the A/D conversion start request signals and their counting directions can be checked by the A/D conversion start request flags (ADTRAUF, ADTRADF, ADTRBUF, and ADTRBDF) in the GTST register. When the output of the same A/D conversion start request signal monitoring output is enabled for multiple channels, ORed signals will be output from the GPT32.

Figure 21.70 shows an example of A/D conversion start request operation, Table 21.48 shows example for setting A/D conversion start request operation.

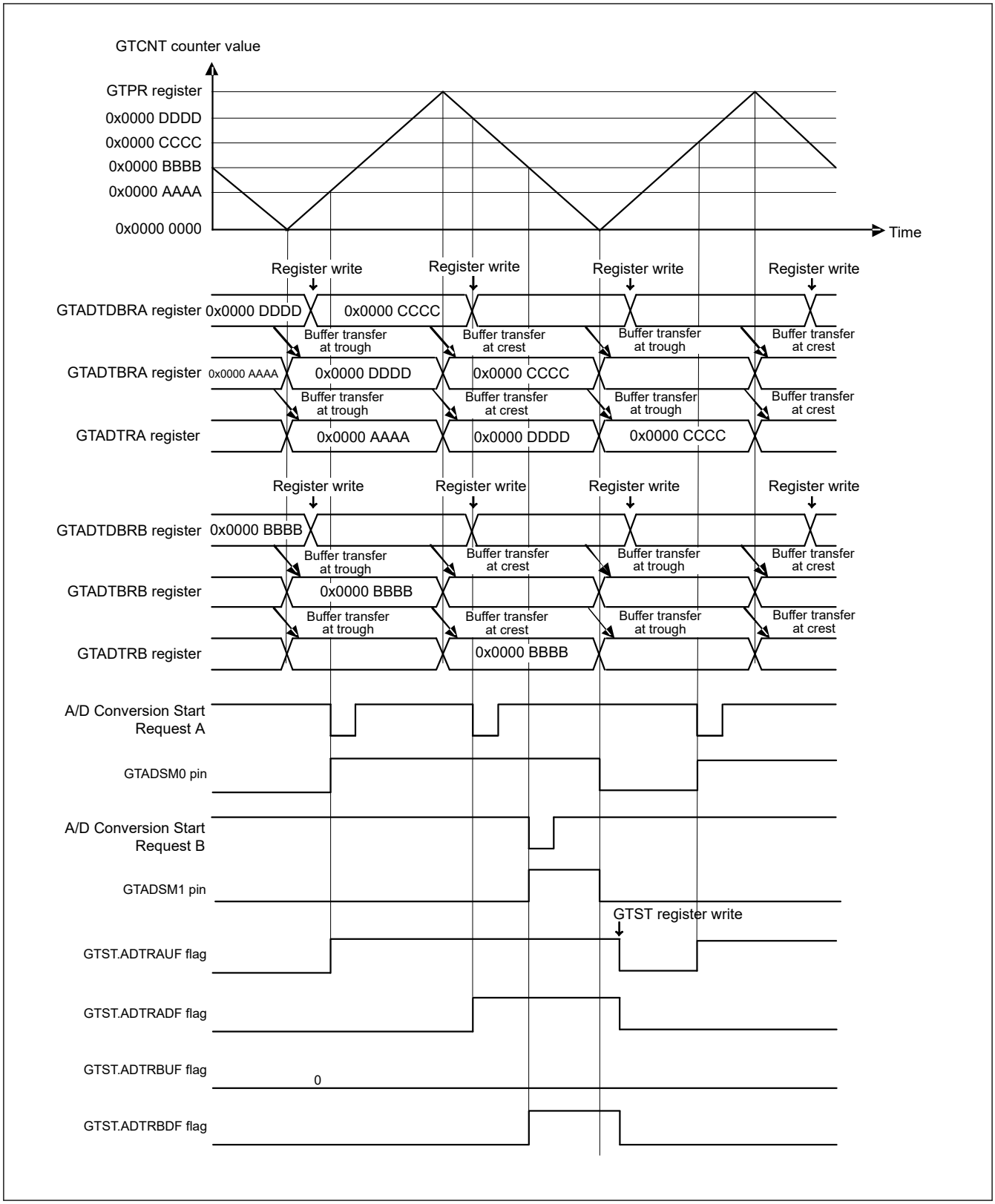


Figure 21.70 Example of A/D conversion start request timing operation (triangle waves, double buffer operation, buffer transfer at both troughs and crests, A/D conversion start request by GTADTRA register at both up-counting and down-counting, A/D conversion start request by GTADTRB register at down-counting)

Table 21.48 Example for setting A/D conversion start request timing operation

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.70, 100b, 101b, or 110b (triangle-wave PWM mode) is set.
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set buffer operation	Set buffer operation with the ADTTA[1:0], ADTTB[1:0], ADTDA, and ADTDB bits in the GTBER register. In Figure 21.70, ADTTA[1:0] = 11b, ADTTB[1:0] = 11b, ADTDA = 1, and ADTDB = 1.
6	Set compare match value	Set the A/D conversion start request point in the GTADTRA and GTADTRB registers.
7	Set buffer value	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.
8	Enable A/D conversion start request	Set to enable A/D conversion start request with the ADTRAUEN, ADTRADEN, ADTRBUEN, and ADTRBDEN bits in the GTINTAD register. In Figure 21.70, ADTRAUEN = 1, ADTRADEN = 1, ADTRBUEN = 0, and ADTRBDEN = 1.
9	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
10	Set buffer value for each cycle	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.

Figure 21.71 shows an example for A/D conversion start request timing operation.

This shows an example of the output of A/D conversion start request A for the A/D converter. The A/D conversion start request A signal is output in response to a match in comparison with the GTADTRA register.

If GPT is operating with PCLKD and ADC is operating with PCLKA, A/D conversion start request A is passed to synchronizer on the next rising edge of PCLKA.

A/D conversion start request A is synchronized and passed to ADC.

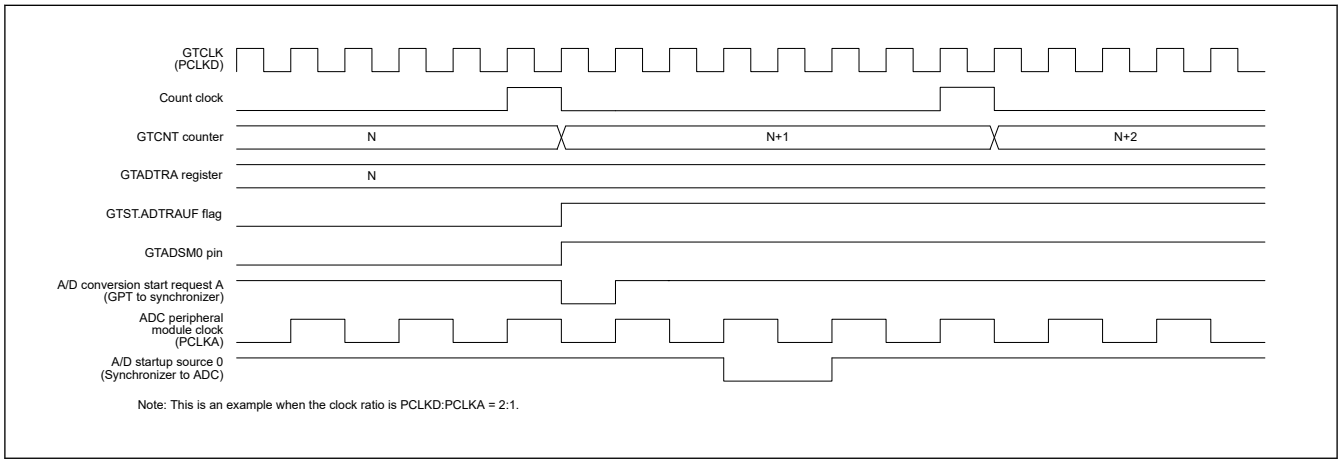


Figure 21.71 Example of A/D conversion start request timing operation

21.6 Operations Linked by ELC

21.6.1 Event Signal Output to ELC

The GPT can perform operation linked with another module set in advance when its interrupt request signal is used as an event signal by the Event Link Controller (ELC).

The A/D conversion start request during up-counting/down-counting can be enabled/disabled individually with the A/D conversion start request enable bit to output events output to ELC.

The GPT has the following ELC event signals:

- Generation of compare match and input capture A interrupt (GPTn_CCMPA)
- Generation of compare match and input capture B interrupt (GPTn_CCMPB)
- Generation of compare match C interrupt (GPTn_CMPC)
- Generation of compare match D interrupt (GPTn_CMPD)
- Generation of compare match E interrupt (GPTn_CMPE)
- Generation of compare match F interrupt (GPTn_CMPF)
- Generation of overflow interrupt (GPTn_OVF)
- Generation of underflow interrupt (GPTn_UDF)
- Finish of period count function (GPTm_PC)

Note: n = 0 to 13
m = 0 to 3, 8 to 10

21.6.2 Event Signal Inputs from ELC

The GPT can perform the following operations in response to a maximum of 8 events from the ELC:

- Start counting, stop counting, clear counting
- Up-counting, down-counting
- Input capture.

See section 18, Event Link Controller (ELC) for the connection between the ELC and the event signal input.

21.7 Noise Filter Function

Each pin for use in input capture and Hall sensor input to the GPT is equipped with a noise filter. The noise filter samples input signals at the sampling clock and removes the pulses whose length is less than 3 sampling cycles.

The noise filter functionality includes enabling and disabling the noise filter for each pin and setting of the sampling clock for each channel.

Figure 21.72 shows the timing of noise filtering.

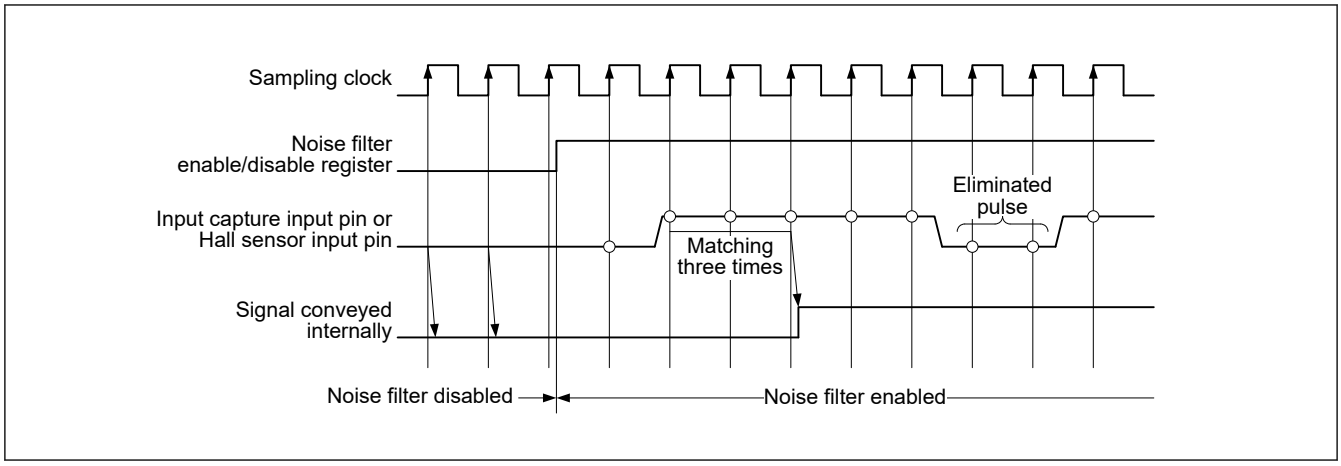


Figure 21.72 Timing of noise filtering

If noise filtering is enabled, the input capture operation or hall sensor input operation is performed on the edges of the noise filtered signal after a delay of (sampling interval × 2 + PCLKD) at the shortest. This is due to the noise filtering for the input capture input or hall sensor input.

21.8 Protection Function

21.8.1 Write-Protection for Registers

To prevent registers from being accidentally modified, registers can be write-protected in channel units by setting GTWP.WP. Write-protection can be set for the following registers:

GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTADTRA, GTADTBRA, GTADTDBRA, GTADTRB, GTADTBRB, GTADTDBRB, GTDTCR, GTDVU, GTADSMR, GTICLF, GTPC.

Every bit in registers GTSTR, GTSTP and GTCLR which can update the corresponding registers in other channels and can be updated by any of the corresponding registers in other channels conversely, can be protected by setting the GTWP.STRWP, STPWP, and CLRWP bits, respectively, per channel.

Likewise, writing to the GTSECSR and GTSECR registers, which can control all channels by writing to the GTSECSR and GTSECR registers of a given channel, can by enabled or disabled by the setting of the GTWP.CMNWP bit.

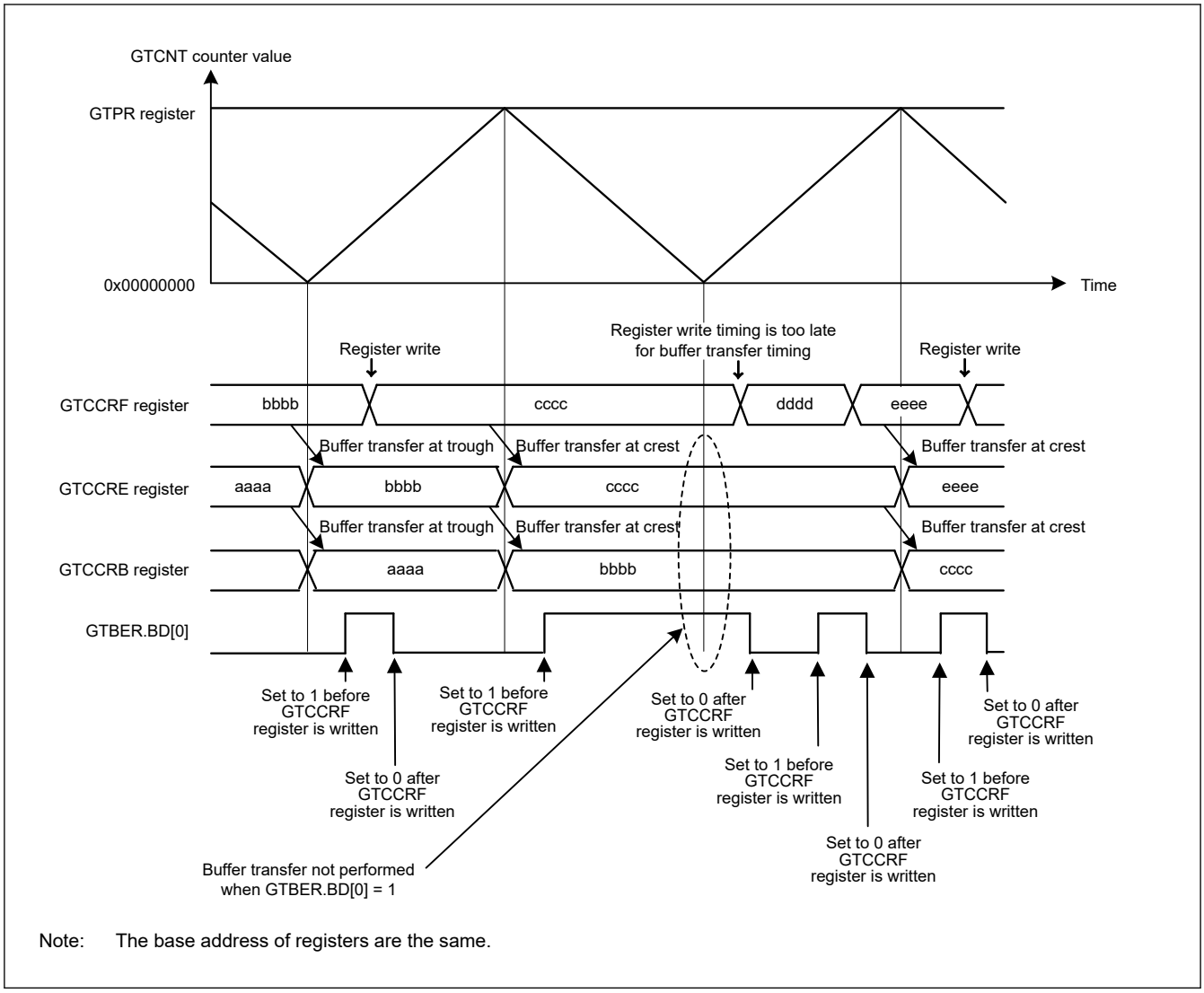
Protection using the GTWP register is only for write operations by the CPU. This protection does not cover updates to registers that occur in association with CPU writes.

21.8.2 Disabling of Buffer Operation

If the timing of the buffer register write is delayed relative to the timing for the buffer transfer, buffer operation can be suspended with the GTBER.BD[2], BD[1] and BD[0] bits settings. Specifically, buffer transfer can be temporarily disabled even though a buffer transfer condition is generated during buffer register write, by setting the BD[2], BD[1] and BD[0] bits to 1 (buffer operation disabled) before buffer register write, and setting the bits to 0 (buffer operation enabled) after completion of writing to all the buffer registers.

The BD[2], BD[1] and BD[0] bits can be set on channel basis by writing directly to the GTBER register or it can be set to 0 simultaneously by setting the GTSECR register for multiple channels which were set by the GTSECSR register.

Figure 21.73 shows an example of operation for disabling buffer operation by writing to the GTBER register.



Note: The base address of registers are the same.

Figure 21.73 Example of operation for disabling buffer operation with triangle waves, double buffer operation, and buffer transfer at both troughs and crests

21.8.2.1 Simultaneous Control of Buffer Operations of Multiple Channels

The GTBER.BD bit can be set by writing directly to the GTBER register per channel or by making settings in the GTSECR register for multiple channels that have already set in the GTSECSR register.

Follow the procedure below to simultaneously set the GTBER.BD bits of multiple channels.

1. Select the channels for simultaneously setting by the GTSECSR register
   Set the GTSECSR register so that the values at the bit positions for the corresponding channels for simultaneously setting of the GTBER.BD bits become 1. All GTSECSR registers can be updated by writing to the GTSECSR register of any channel.
2. Simultaneously set the GTBER.BD bits by updating the GTSECR register
   In the GTSECR register, set the operation of the GTBER.BD bits (enabling or disabling of buffer operation) which are to be simultaneously set. Writing to a GTSECR register from any channel updates the GTBER.BD bits in all channels corresponding to the bits set as 1 in the GTSECSR register, in accordance with the value of the GTSECR register.

Figure 21.74 show examples of simultaneously controlling the enabling or disabling of buffer operation for multiple channels.

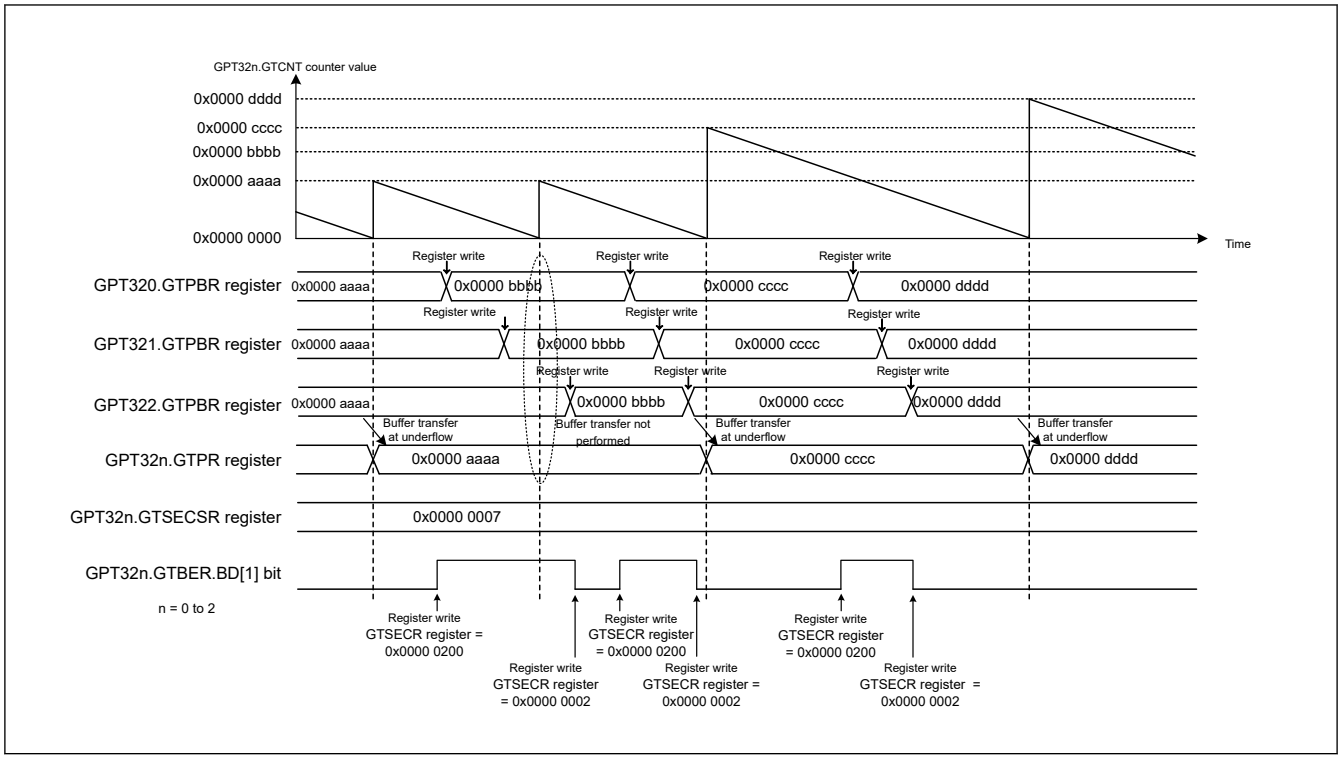


Figure 21.74 Example of Multiple Channel Operation for Disabling Buffer Operation (Saw Waves, Single Buffer Operation)

21.8.3 GTIOCnm Pin Output Negate Control (n = 0 to 13, m = A, B)

For protection from system failure, the output disable control that changes the GTIOCnm pin output value forcibly is provided for GTIOCnm pin output by the request of output disable from POEG. Output protection is required when the same output level being on the GTIOCnA and GTIOCnB pins is detected. GPT detects this condition and generates output disable requests to POEG according to the setting of the output disable request permission bits, such as GTINTAD.GRPABH, GTINTAD.GRPABL. After the POEG performs the logical OR of the output disable request from each channel and the output disable request from the external input, the POEG generates output disable requests to GPT.

One output disable signal (representing the shared output disable request signal of the GTIOCnA pin and the GTIOCnB pin) out of 4 output disable requests generated by the POEG is selected by setting GTINTAD.GRP[1:0]. The status of the selected disable output request is monitored by reading the GTST.ODF bit. The output level during output disable is set based on the GTIOR.OADF[1:0] bits for the GTIOCnA pin and the GTIOR.OBDF[1:0] setting for the GTIOCnB pin.

The change to the output disable state is performed asynchronously by generating the output disable request from the POEG. The release of the output disable state is performed at end of cycle by terminating the output disable request. It is after 3 PCLKD at shortest when the output disable condition is released after the output disable request becomes no longer satisfied. To reliably control output disabling, clear the flag of POEG for which the condition for the request to disable the output is no longer satisfied after 4 cycles of PCLKD.

When event count is performed or when the output disable state should be released immediately without waiting for end of cycle, GTIOR.OADF[1:0] should be set to 00b (for GTIOCnA pin) or GTIOR.OBDF[1:0] should be set to 00b (for the GTIOCnB pin).

Figure 21.75 shows an example of the GTIOCnm pin output disable control operation. (n = 0 to 13, m = A, B)

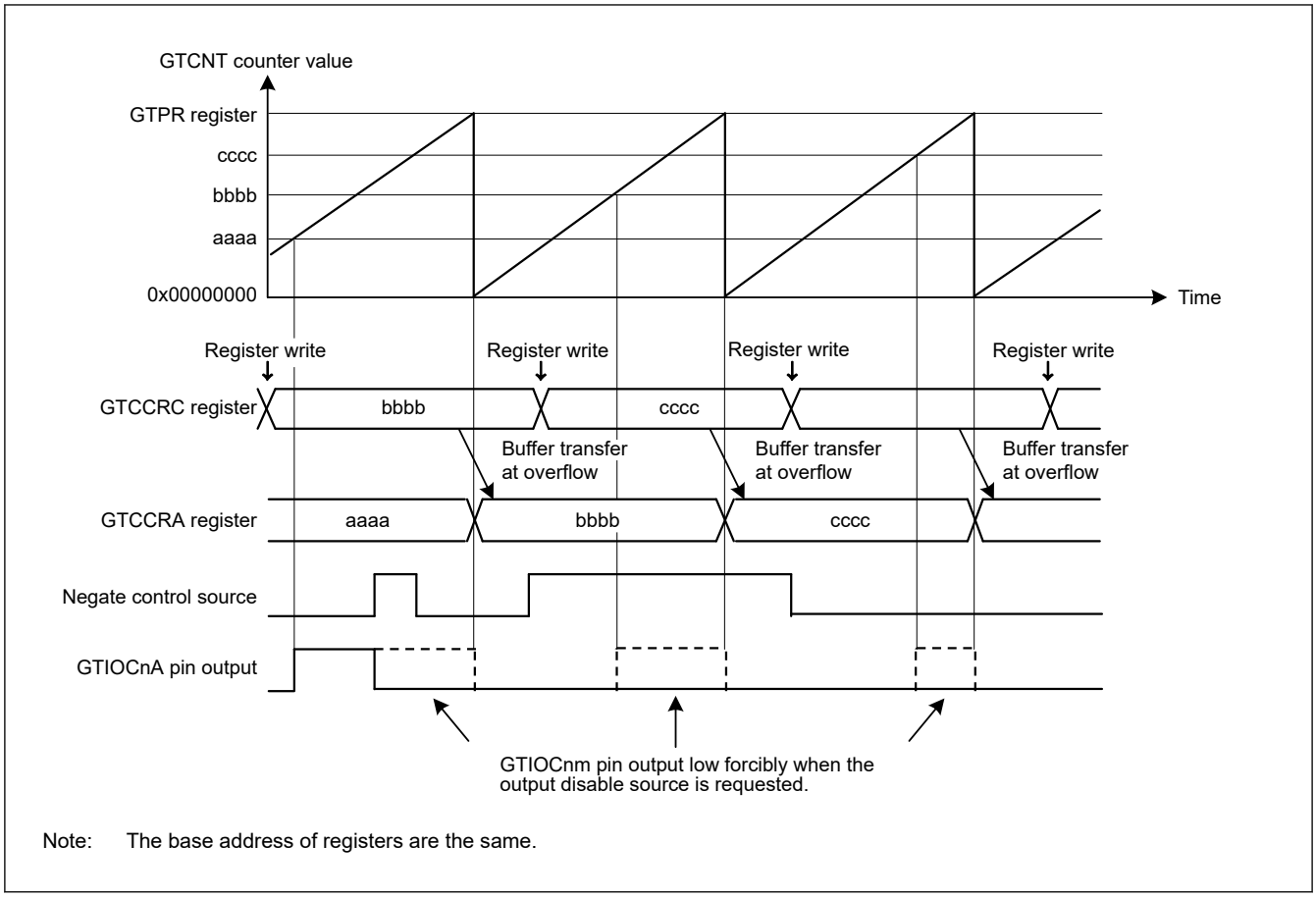


Figure 21.75 Example of GTIOCnm pin output disable control operation in saw-wave up-counting, buffer operation, active level 1, high output at GTCCRA compare match, low output at cycle end, and low output at output disable (n = 0 to 13, m = A, B)

21.9 Initialization Method of Output Pins

21.9.1 Pin Settings after Reset

The GPT registers are initialized at a reset. Start counting after selecting the port pin function with the PmnPFS register, setting GTIOR.OAE and GTIOR.OBE bits, and outputting the GPT function to external pins.

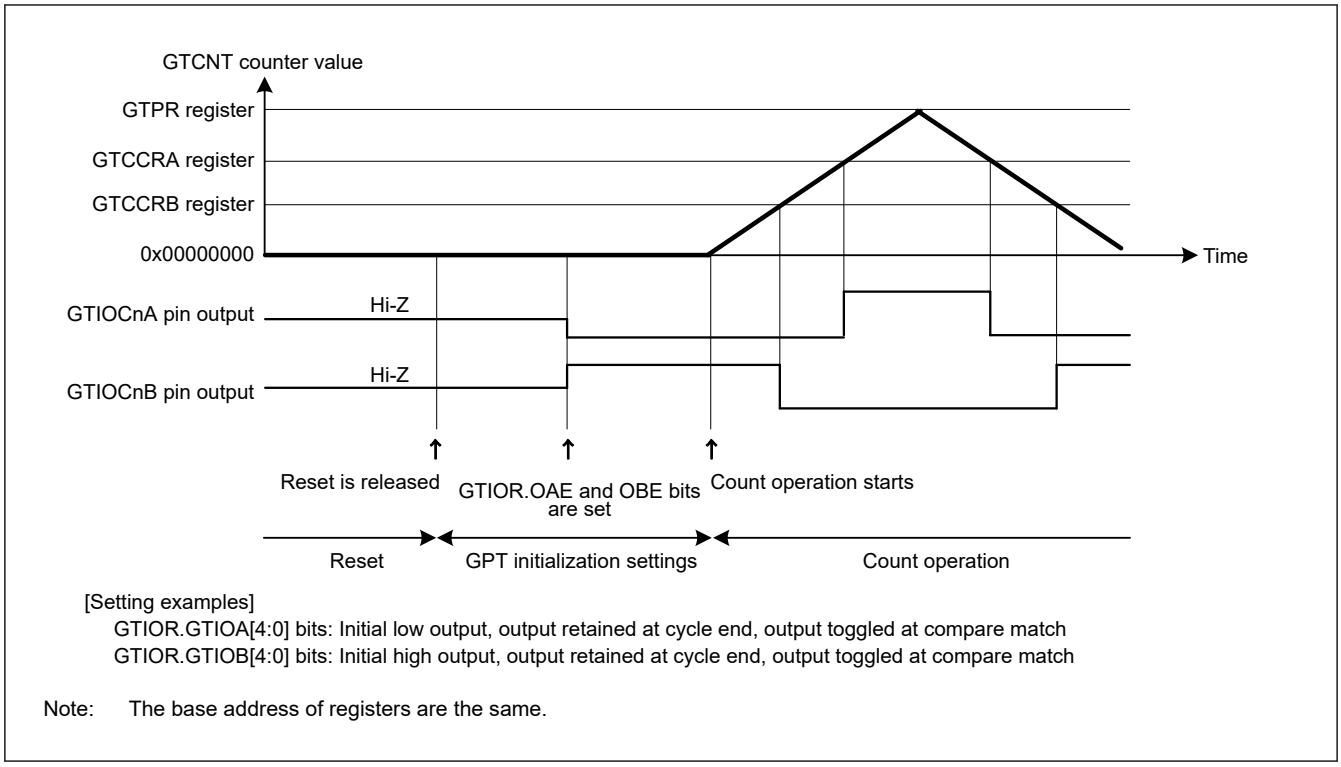


Figure 21.76 Example of pin settings after reset

21.9.2 Pin Initialization Due to Error During Operation

If an error occurs during GPT operation, the following four types of pin control can be performed before pin initialization:

- Set the OAHLD and OBHLD bits in GTIOR to 1 and retain the outputs at count stop
- Set the OAHLD and OBHLD bits in GTIOR to 0, specify arbitrary output values at OADFLT and OBDFLT in GTIOR, and output the arbitrary values at count stop
- Set the pin to output an arbitrary value as a general output port by setting the PDR, PODR registers and PmnPFS.PMR bit of the I/O port in advance. Set the OAE and OBE bits in GTIOR to 0, and the control bit associated with the pin in the PMR to 0 to allow arbitrary values to be output from the pin set as a general output port when an error occurs.
- Drive the output to a high impedance state using the POEG function.

If the automatic dead time setting is made, clear the GTDTCR.TDE bit to 0 after counting stops. When counting stops, only the values of registers that are changed by a GPT external source change. If counting is resumed, operation continues from where it stopped. If counting is stopped, the registers must be initialized before counting starts.

21.10 Usage Notes

21.10.1 Module-Stop Function Setting

The Module Stop Control Register can enable or disable GPT operation. The GPT is initially stopped after a reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

21.10.2 GTCCRn Settings during Compare Match Operation (n = A to F)

(1) When automatic dead time setting is made in triangle-wave PWM mode

The GTCCRA register must satisfy all of the following conditions:

- GTDVU < GTCCRA
- 0 < GTCCRA < GTPR

(2) When automatic dead time setting is not made in triangle-wave PWM mode

The GTCCRA register must be set within the range of 0 < GTCCRA < GTPR. If GTCCRA = 0 or GTCCRA = GTPR is set, a compare match occurs within the cycle only when GTCCRA = 0 or GTCCRA = GTPR is satisfied. When GTCCRA > GTPR, no compare match occurs.

Similarly, GTCCRB must be set within the range of 0 < GTCCRB < GTPR. If GTCCRB = 0 or GTCCRB = GTPR is set, a compare match occurs within the cycle only when GTCCRB is 0 or GTCCRB = GTPR is satisfied. When GTCCRB > GTPR, no compare match occurs.

(3) When automatic dead time setting is made in saw-wave one-shot pulse mode

The GTCCRC and GTCCRD registers must be set to satisfy the following restrictions. If the restrictions are not satisfied, the correct output waveforms with secured dead time may not be obtained.

- In up-counting: GTCCRC < GTCCRD, GTCCRC > GTDVU, GTCCRD < GTPR - GTDVU
- In down-counting: GTCCRC > GTCCRD, GTCCRC < GTPR - GTDVU, GTCCRD > GTDVU

(4) When automatic dead time setting is not made in saw-wave one-shot pulse mode

The GTCCRC and GTCCRD registers must be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRC < GTCCRD < GTPR
- In down-counting: GTPR > GTCCRC > GTCCRD > 0

Similarly, GTCCRE and GTCCRF must be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRE < GTCCRF < GTPR
- In down-counting: GTPR > GTCCRE > GTCCRF > 0.

(5) In saw-wave PWM mode

The GTCCRA register must be set with the range of 0 < GTCCRA < GTPR. If GTCCRA = 0 or GTCCRA = GTPR is set, a compare match occurs within the cycle only when GTCCRA = 0 or GTCCRA = GTPR is satisfied. If GTCCRA > GTPR is set, no compare match occurs.

Similarly, GTCCRB must be set with the range of 0 < GTCCRB < GTPR. If GTCCRB = 0 or GTCCRB = GTPR is set, a compare match occurs within the cycle only when GTCCRB = 0 or GTCCRB = GTPR is satisfied. If GTCCRB > GTPR is set, no compare match occurs.

21.10.3 Setting Range for GTCNT Counter

The GTCNT counter register must be set with the range of 0 ≤ GTCNT ≤ GTPR.

21.10.4 Starting and Stopping the GTCNT Counter

The control timing of starting and stopping the GTCNT counter by the GTCR.CST bit synchronizes the count clock that is selected in GTCR.TPCS[3:0]. When GTCR.CST is updated, the GTCNT counter starts/stops after a count clock that is selected in GTCR.TPCS[3:0]. Therefore, an event generated before the GTCNT counter actually starts is ignored, resulting in situations in which an event is accepted or an interrupt occurs after GTCR.CST is set to 0.

21.10.5 Priority Order of Each Event

(1) GTCNT register

Table 21.49 shows a priority order of events updating the GTCNT register.

Table 21.49 Priority order of sources updating GTCNT

Source updating GTCNT	Priority order
Writing by CPU (writing to GTCNT/GTCLR)	High
Clear by hardware sources set in GTCSR	↑
Count up or down by hardware sources set in GTUPSR/GTDNSR	↑
Count operation	Low

If up-counting and down-counting by hardware sources occur at the same time, the GTCNT counter value does not change. When there is a conflict between updating the GTCNT register and reading by the CPU, pre-update data is read.

(2) GTCR.CST bit

When there is a conflict between starting/stopping by hardware sources set in the GTSSR/GTPSR registers and writing by the CPU (writing to GTCR/GTSTR/GTSTP registers), the writing by CPU has priority over the starting/stopping by hardware sources.

In case that stop by the period count function conflicts with start by the CPU writing (GTCR register writing/GTSTR register writing), the period count function is finished with setting the GTST.PCF flag. The CST bit is not changed and the GTCNT continues to count.

When there is a conflict between starting by hardware sources set in the GTSSR register and stopping by hardware sources set in GTPSR register, the GTCR.CST bit value does not change. When there is a conflict between updating the GTCR.CST bit and reading by the CPU (reading from GTCR/GTSTR/GTSTP registers), pre-update data is read.

(3) GTCCRm registers (m = A to F)

When there is a conflict between input capture/buffer transfer operation and writing to the GTCCRm registers, the writing to GTCCRm registers has priority over input capture/buffer transfer operation. When there is a conflict between input capture and writing to the counter register by the CPU or updating the counter register by hardware sources, the pre-update counter value is captured. When there is a conflict between updating the GTCCRm registers and reading by the CPU, pre-update data is read.

(4) GTPR register

When there is a conflict between buffer transfer operation and writing to the GTPR register, writing to GTPR register has priority over buffer transfer operation. When there is a conflict between updating GTPR register and reading by the CPU, pre-update data is read.

(5) GTADTRm registers (m = A, B)

When there is a conflict between buffer transfer operation and writing to GTADTRm register, writing to GTADTRm register has priority over buffer transfer operation.

When there is a conflict between updating the GTADTRm register and reading by the CPU, pre-update data is read.

(6) GTDVU registers

When there is a conflict between buffer transfer operation and writing to GTDVU register, writing to GTDVU register has priority over buffer transfer operation.

When there is a conflict between updating the GTDVU register and reading by the CPU, pre-update data is read.

(7) GTIOR.GTIOm registers (m = A, B)

When there is a conflict between buffer transfer operation and writing to GTIOR.GTIOm register, writing to GTIOR.GTIOm register has priority over buffer transfer operation.

When there is a conflict between updating the GTIOR.GTIOm and reading by the CPU, pre-update data is read.

21.10.6 Prohibit Invalid Register Setting

The register settings that are instructed to be invalid, such as "The setting is invalid during event counting operation", are not guaranteed. These setting is prohibited.

22. Low Power Asynchronous General Purpose Timer (AGT)

22.1 Overview

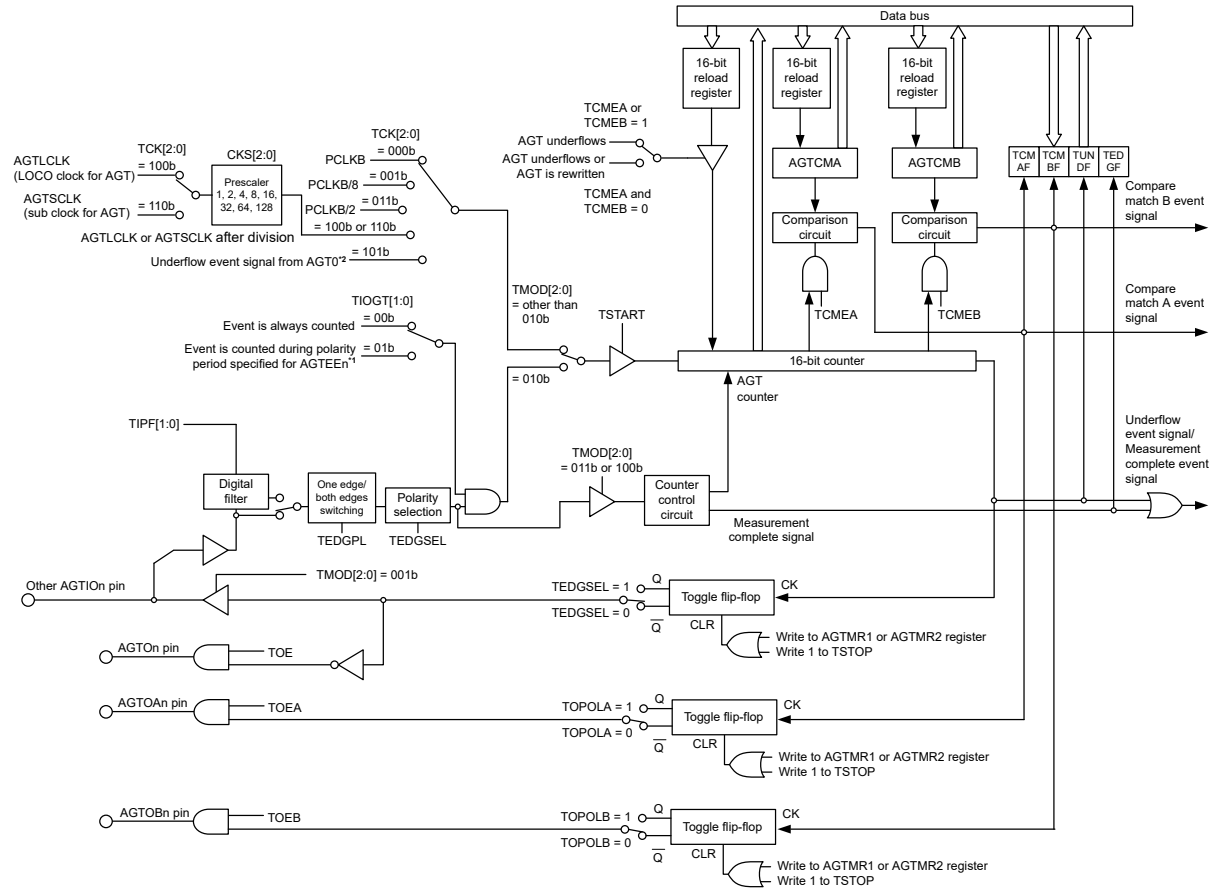
The Low Power Asynchronous General Purpose Timer (AGT) is a 16-bit timer that can be used for pulse output, external pulse width or period measurement, and counting external events. This timer consists of a reload register and a down counter. The reload register and the down counter are allocated to the same address, and can be accessed with the AGT register.

Table 22.1 lists the AGT specifications, Figure 22.1 shows a block diagram, and Table 22.2 lists the I/O pins.

Table 22.1 AGT specifications

Parameter		Description
Operating modes	Timer mode	The count source is counted
	Pulse output mode	The count source is counted and the output is inverted at each timer underflow
	Event counter mode	An external event is counted
	Pulse width measurement mode	An external pulse width is measured
	Pulse period measurement mode	An external pulse period is measured
Number of Channels		16 bits × 2 channels (AGTn (n = 0, 1))
Count source (operating clock)*2	Timer mode	PCLKB, PCLKB/2, PCLKB/8, AGTLCLK/d, AGTSCLK/d (d = 1, 2, 4, 8, 16, 32, 64, or 128), or underflow signal of AGT0 selectable.*1
	Pulse output mode	
	Pulse width measurement mode	
	Pulse period measurement mode	
	Event counting mode	External event input
Interrupt and Event Link function		• Underflow event signal or measurement complete event signal – When the counter underflows – When the measurement of the active width of the external input pin (AGTIOn) completes in pulse width measurement mode – When the set edge of the external input pin (AGTIOn) is input in pulse period measurement mode. • Compare match A event signal – When the values of AGT register and AGTCMA register matched (compare match A function enabled). • Compare match B event signal – When the values of AGT and AGTCMB registers matched (compare match B function enabled). • Return from Software Standby mode can be performed with AGT1_AGTI, AGT1_AGTCMAI, or AGT1_AGTCMBI*3
Selectable functions		• Compare match function One or two of the AGT Compare Match A register and AGT Compare Match B register is selectable.
Module-stop function		Module-stop state can be set for each channel to reduce power consumption.
TrustZone Filter		Security and Privilege attribution can be set for each channels.

Note 1. AGT0 cannot use underflow signal. AGT1 connects directly with the underflow event signal from the AGT0 timer.
Note 2. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ the frequency of the count source clock. But when selecting AGTLCLK/d or AGTSCLK/d as the count source clock, the frequency of PCLKB should be at least twice that of the count source clock.
Note 3. For details, see section 10, Low Power Modes.



Note 1. The polarity can be selected by the EEPS bit in the AGTISR register.
Note 2. AGT0 cannot use AGT0 underflow event. AGT1 uses the underflow of AGT0.

Figure 22.1 AGT block diagram

Table 22.2 AGT I/O pins

Pin name	I/O	Function
AGTEEn	Input	External event input enable for AGT
AGTIOn	Input/output	External event input and pulse output for AGT
AGTOn	Output	Pulse output for AGT
AGTOAn	Output	Compare match A output for AGT
AGTOBn	Output	Compare match B output for AGT

Note: Channel number: n = 0, 1

22.2 Register Descriptions

22.2.1 AGT : AGT Counter Register

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x00

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	16-bit counter and reload register Setting range : 0x0000 to 0xFFFF	R/W

Note: S-TYPE-3, P-TYPE-3

AGTn.AGT is a 16-bit register. The write value is written to the reload register and the read value is read from the counter.

The states of the reload register and the counter change according to the TSTART bit in the AGTCR register and TCMEA/TCMEB bit in the AGTCMSR register. For details, see section 22.3.1. Reload Register and Counter Rewrite Operation.

When 1 is written to the TSTOP bit in the AGTCR register, AGT counter is forcibly stopped and set to 0xFFFF.

When the TCK[2:0] bits setting in the AGTMR1 register are a value other than 001b (PCLKB/8) or 011b (PCLKB/2), if the AGT register is set to 0x0000, a request signal to the ICU, the DTC, the DMAC, and the ELC is generated once immediately after the count starts. The AGTOn, AGTIOn pin output are toggled.

When the AGT register is set to 0x0000 in event counter mode, regardless of the value of TCK[2:0] bits, a request signal to the ICU, the DTC, the DMAC, and the ELC is generated once immediately after the count starts.

In addition, the AGTOn pin output is toggled even during a period other than the specified count period. When the AGT register is set to 0x0001 or more, a request signal is generated each time AGT underflows.

22.2.2 AGTCMA : AGT Compare Match A Register

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x02

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	16-bit compare match A data is stored.*1 Setting range : 0x0000 to 0xFFFF	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Set the AGTCMA register to 0xFFFF when compare match A is not used.

The AGTCMA register is a read/write register to set a value for compare match with the AGT counter. The states of the reload register and compare register A change according to the TSTART bit in the AGTCR register. For details, see section 22.3.2. Reload Register and AGT Compare Match A/B Register Rewrite Operation.

22.2.3 AGTCMB : AGT Compare Match B Register

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x04

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	16-bit compare match B data is stored.*1 Setting range : 0x0000 to 0xFFFF	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Set the AGTCMB register to 0xFFFF when compare match B is not used.

The AGTCMB register is a read/write register to set a value for compare match with the AGT counter. The states of the reload register and compare register B change according to the TSTART bit in the AGTCR register. For details, see section 22.3.2. Reload Register and AGT Compare Match A/B Register Rewrite Operation.

22.2.4 AGTCR : AGT Control Register

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TCMBF	TCMAF	TUNDF	TEDGF	—	TSTOP	TCSTF	TSTART
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSTART	AGT Count Start*2 0: Count stops 1: Count starts	R/W
1	TCSTF	AGT Count Status Flag*2 0: Count stopped 1: Count in progress	R
2	TSTOP	AGT Count Forced Stop*1 0: Writing is invalid 1: The count is forcibly stopped	W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	TEDGF	Active Edge Judgment Flag 0: No active edge received 1: Active edge received	R/(W)*3
5	TUNDF	Underflow Flag 0: No underflow 1: Underflow	R/(W)*3
6	TCMAF	Compare Match A Flag 0: No match 1: Match	R/(W)*3
7	TCMBF	Compare Match B Flag 0: No match 1: Match	R/(W)*3

Note: S-TYPE-3, P-TYPE-3

Note 1. When 1 (count is forcibly stopped) is written to the TSTOP bit, the TSTART bit and TCSTF flag are initialized at the same time. The pulse output level is also initialized. The read value is 0.
Note 2. For information on using the TSTART bit and TCSTF flag, see section 22.4.1. Count Operation Start and Stop Control.
Note 3. Only 0 can be written to clear the flag.

TSTART bit (AGT Count Start)

The count operation is started by writing 1 to the TSTART bit and stopped by writing 0. When the TSTART bit is set to 1 (count starts), the TCSTF flag is set to 1 (count in progress) in synchronization with the count source. Also, after 0 is written to the TSTART bit, the TCSTF flag is set to 0 (count stops) in synchronization with the count source. For details, see section 22.4.1. Count Operation Start and Stop Control.

TCSTF flag (AGT Count Status Flag)

The TCSTF flag indicates the AGT count status.

[Setting condition]

- When 1 is written to the TSTART bit (the TCSTF flag is set to 1 in synchronization with the count source).

[Clearing conditions]

- When 0 is written to the TSTART bit (the TCSTF flag is set to 0 in synchronization with the count source)
- When 1 is written to the TSTOP bit.

TSTOP bit (AGT Count Forced Stop)

When 1 is written to the TSTOP bit, the count is forcibly stopped. The read value is 0.

TEDGF flag (Active Edge Judgment Flag)

The TEDGF flag indicates that an active edge was detected.

[Setting condition]

- When the measurement of the active width of the external input pin (AGTIOn) is complete in pulse width measurement mode
- When the set edge of the external input pin (AGTIOn) is input in pulse period measurement mode.

[Clearing condition]

- When 0 is written to this flag by software.

TUNDF flag (Underflow Flag)

The TUNDF flag indicates that the counter underflowed.

[Setting condition]

- When the counter underflows.

[Clearing condition]

- When 0 is written to this flag by software.

TCMAF flag (Compare Match A Flag)

The TCMAF flag indicates that compare match A was detected.

[Setting condition]

- When the value in the AGT register matches the value in the AGTCMA register.

[Clearing condition]

- When 0 is written to this flag by software.

TCMBF flag (Compare Match B Flag)

The TCMBF flag indicates that compare match B was detected.

[Setting condition]

- When the value in the AGT register matches the value in the AGTCMB register.

[Clearing condition]

- When 0 is written to this flag by software.

22.2.5 AGTMR1 : AGT Mode Register 1

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x09

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	TCK[2:0]			TEDGPL	TMOD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TMOD[2:0]	Operating Mode*3 0 0 0: Timer mode 0 0 1: Pulse output mode 0 1 0: Event counter mode 0 1 1: Pulse width measurement mode 1 0 0: Pulse period measurement mode Others: Setting prohibited	R/W
3	TEDGPL	Edge Polarity*4 0: Single-edge 1: Both-edge	R/W
6:4	TCK[2:0]	Count Source*1 *2 *5 *7 0 0 0: PCLKB 0 0 1: PCLKB/8 0 1 1: PCLKB/2 1 0 0: Divided clock AGTLCLK specified by CKS[2:0] bits in the AGTMR2 register 1 0 1: Underflow event signal from AGT0*6 1 1 0: Divided clock AGTSCLK specified by CKS[2:0] bits in the AGTMR2 register Others: Setting prohibited	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Write access to the AGTMR1 register initializes the output from the AGTOn, AGTIOn, AGTOAn, and AGTOBn pins. For details on the output level at initialization, see section 22.2.7. AGTIOC : AGT I/O Control Register.
Note 1. When event counter mode is selected, the external input pin (AGTIOn) is selected as the count source regardless of the setting of TCK[2:0] bits.
Note 2. Do not switch count sources during count operation. Only switch count sources when both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count stops).
Note 3. The operating mode can only be changed when the count is stopped while both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count is stopped). Do not change the operating mode during count operation.
Note 4. The TEDGPL bit is enabled only in event counter mode.
Note 5. To run AGT in Software Standby mode, select AGTLCLK or AGTSCLK (TCK[2:0] = 100b, 110b).
Note 6. AGT0 cannot use AGT0 underflow (setting prohibited). AGT1 uses the AGT0 underflow.
Note 7. Do not change the TCK[2:0] bits when the CKS[2:0] bits in the AGTMR2 register is not 000b. First, change the CKS[2:0] bits in the AGTMR2 register to 000b. Then change the TCK[2:0] bits and wait for one cycle of the count source.

22.2.6 AGTMR2 : AGT Mode Register 2

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LPM	—	—	—	—	CKS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CKS[2:0]	AGTLCLK or AGTSCLK Count Source Clock Frequency Division Ratio*1 *2 *3 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	LPM	Low Power Mode 0: Normal mode 1: Low power mode	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Do not rewrite the CKS[2:0] bits during count operation. Only rewrite the CKS[2:0] bits when both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count stops).
Note 2. When count source is AGTLCLK or AGTSCLK, the switch of CKS[2:0] bits is valid.
Note 3. Do not switch the TCK[2:0] bits in the AGTMR1 register when CKS[2:0] bits are not 000b. Switch the TCK[2:0] bits in the AGTMR1 register after CKS[2:0] bits are set to 000b, and wait for 1 cycle of the count source.

CKS[2:0] bit (AGTLCLK or AGTSCLK Count Source Clock Frequency Division Ratio)

CKS[2:0] bits select the Count Source Clock Frequency Division Ratio for AGTLCLK or AGTSCLK.

LPM bit (Low Power Mode)

The LPM bit sets the low power operation, which impacts access to certain AGT registers. Set this bit to 1 to operate in low power.

When this bit is 1, access to the following registers is prohibited:

- AGT/AGTCMA/AGTCMB/AGTCR.

After this bit is switched from 1 to 0, the first access to the register is constrained as follows:

- When reading from the AGT register, read AGT register twice. Only the second reading of data is valid.
- When writing to the AGT, AGTCMA, AGTCMB, and AGTCR register, allow at least 2 cycles of the count source clock when writing to the register.
- When confirm the value written to the AGT, AGTCMA, AGTCMB, and AGTCR registers.
  - When the count operation is stopped; after writing data, it can be read in the next cycle.
  - When the count operation is operating; after writing data, it can be read 4 cycles after the count source clock.

Figure 22.2 shows the flow of how to write LPM bit

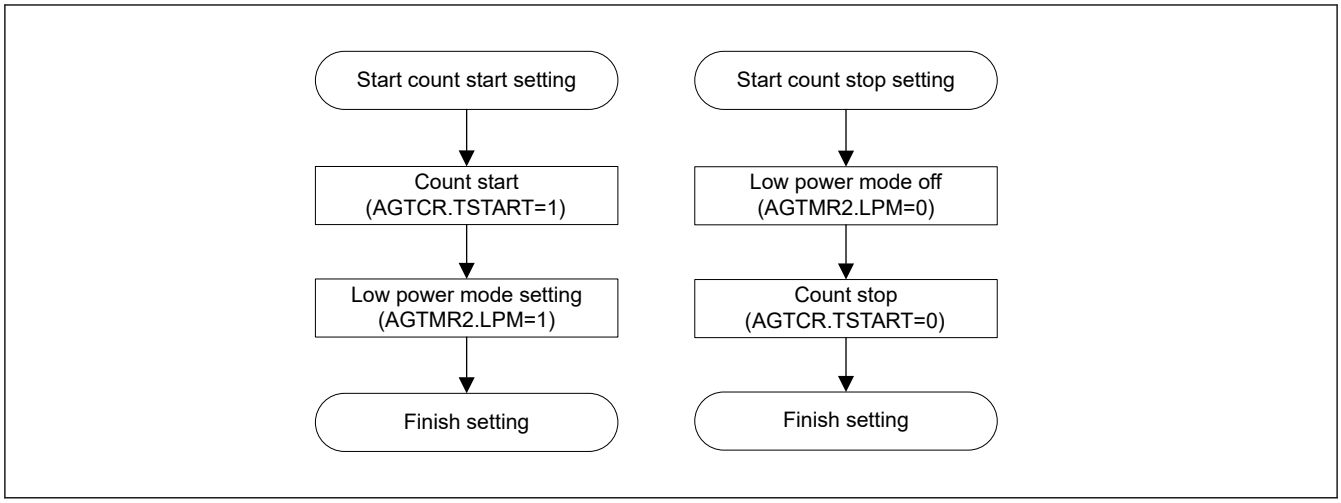


Figure 22.2 LPM how to write flow chart

22.2.7 AGTIOC : AGT I/O Control Register

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TIOGT[1:0]		TIPF[1:0]		—	TOE	—	TEDG SEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TEDGSEL	I/O Polarity Switch Function varies depending on the operating mode (see Table 22.3 and Table 22.4).	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	TOE	AGTOn pin Output Enable 0: AGTOn pin output disabled 1: AGTOn pin output enabled	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	TIPF[1:0]	Input Filter*3 These bits specifies the sampling frequency of the filter for the AGTIOn input. If the input to the AGTIOn pin is sampled and the value matches three successive times, that value is taken as the input value. 0 0: No filter 0 1: Filter sampled at PCLKB 1 0: Filter sampled at PCLKB/8 1 1: Filter sampled at PCLKB/32	R/W
7:6	TIOGT[1:0]	Count Control*1 *2 0 0: Event is always counted 0 1: Event is counted during polarity period specified for AGTEEn pin Others: Setting prohibited	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. When AGTEEn pin is used, the polarity to count an event can be selected with the EEPS bit in the AGTISR register.
Note 2. TIOGT[1:0] bits are enabled only in event counter mode.
Note 3. When event counter mode operation is performed during Software Standby mode, the digital filter function cannot be used.

TEDGSEL bit (I/O Polarity Switch)

The TEDGSEL bit switches the AGTOn pin output polarity and the AGTIOn pin input/output edge and polarity.

In pulse output mode, it only controls polarity of the AGTOn pin output and AGTIOn pin output. AGTOn pin output and AGTIOn pin output are initialized when the AGTMR1 register is written or the TSTOP bit in the AGTCR register is written with 1.

TOE bit (AGTOn pin Output Enable)

The TOE bit selects whether the AGTOn pin output is disabled or enabled.

TIPF[1:0] bits (Input Filter)

The TIPF[1:0] bits specify the sampling frequency of the AGTIOn pin input filter. When the input to the AGTIOn pin is sampled and the values match three times in succession, the value is regarded as the input value.

TIOGT[1:0] bits (Count Control)

The TIOGT[1:0] bits control the event count.

Table 22.3 AGTIOn pin I/O edge and polarity switching

Operating mode	Function
Timer mode	Not used
Pulse output mode	0: Output is started at high (initialization level: high) i.e. inverted output 1: Output is started at low (initialization level: low). i.e. normal output
Event counter mode	0: Count on rising edge 1: Count on falling edge.
Pulse width measurement mode	0: Low-level width is measured 1: High-level width is measured.
Pulse period measurement mode	0: Measure from one rising edge to the next rising edge 1: Measure from one falling edge to the next falling edge.

Table 22.4 AGTOn pin output polarity switching

Operating mode	Function
All modes	0: Output is started at low (initial level: low): Normal output 1: Output is started at high (initial level: high): Inverted output

22.2.8 AGTISR : AGT Event Pin Select Register

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	EEPS	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	EEPS	AGTEEn Polarity Selection 0: An event is counted during the low-level period 1: An event is counted during the high-level period	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

EEPS bit (AGTEEn Polarity Selection)

The EEPS bit selects the polarity of events to be counted.

22.2.9 AGTCMSR : AGT Compare Match Function Select Register

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	TOPOLB	TOEB	TCMEB	—	TOPOLA	TOEA	TCMEA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCMEA	AGT Compare Match A Register Enable*1 *2 0: AGT Compare match A register disabled 1: AGT Compare match A register enabled	R/W
1	TOEA	AGTOAn Pin Output Enable*1 *2 0: AGTOAn pin output disabled 1: AGTOAn pin output enabled	R/W
2	TOPOLA	AGTOAn Pin Polarity Select*1 *2 0: AGTOAn pin output is started on low. i.e. normal output 1: AGTOAn pin output is started on high. i.e. inverted output	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	TCMEB	AGT Compare Match B Register Enable*1 *2 0: Compare match B register disabled 1: Compare match B register enabled	R/W
5	TOEB	AGTOBn Pin Output Enable*1 *2 0: AGTOBn pin output disabled 1: AGTOBn pin output enabled	R/W
6	TOPOLB	AGTOBn Pin Polarity Select*1 *2 0: AGTOBn pin output is started on low. i.e. normal output 1: AGTOBn pin output is started on high. i.e. inverted output	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Do not rewrite the AGTCMSR register during a count operation. Only rewrite the AGTCMSR register when both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count stops).
Note 2. Do not set 1 when in pulse width measurement mode or pulse period measurement mode.

22.2.10 AGTIOSEL : AGT Pin Select Register

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TIES	—	—	—	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	TIES	AGTIOn Pin Input Enable 0: External event input is disabled during Software Standby mode 1: External event input is enabled during Software Standby mode	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The AGTIOSEL register sets the AGTIOn pin when using the AGTIOn pin in Software Standby mode.

TIES bit (AGTIOn Pin Input Enable)

The TIES bit enables or disables an external event input.

22.3 Operation

22.3.1 Reload Register and Counter Rewrite Operation

Regardless of the operating mode, the timing of the rewrite operation to the reload register and the counter differs depending on the value of the TSTART bit in the AGTCR register and of the TCMEA or TCMEB bit in the AGTCMSR register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and the counter. When the TSTART bit is 1 (count starts) and the TCMEA bit and TCMEB bit are 0 (AGT compare match A/B register are invalid), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the next count source. When the TSTART bit is 1 (count starts) and the TCMEA bit or the TCMEB bit is 1 (AGT compare match A register or compare match B register is valid), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the underflow of the counter.

Figure 22.3 and Figure 22.4 show the timing of rewrite operation with TSTART bit value and TCMEA/TCMEB bit value.

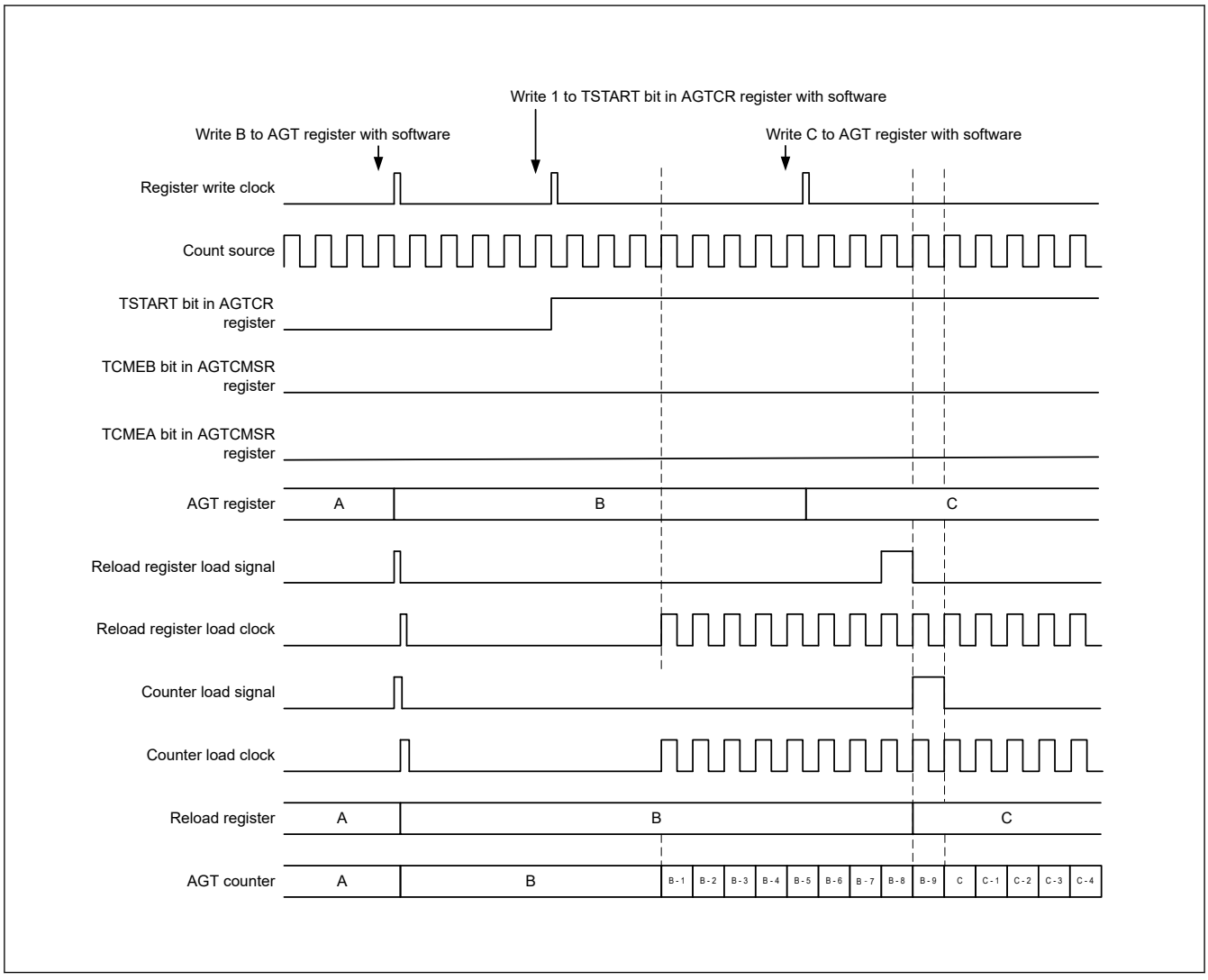


Figure 22.3 Timing of rewrite operation with TSTART, TCMEA, and TCMEB bit value when AGT compare match A register and AGT compare match B register is invalid

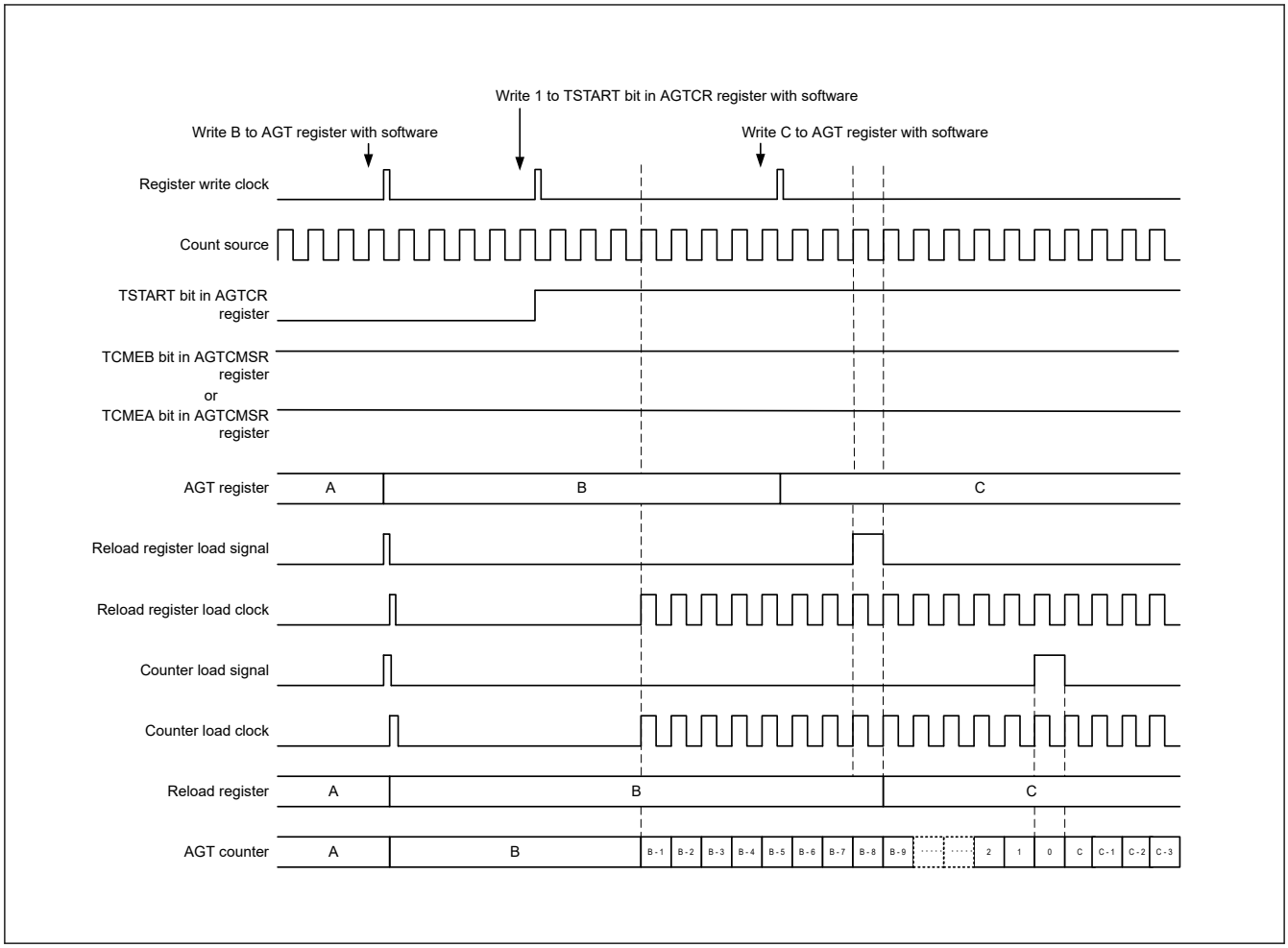


Figure 22.4 Timing of rewrite operation with TSTART bit value and TCMEA or TCMEB bit value when AGT compare match A register or AGT compare match B register is valid

22.3.2 Reload Register and AGT Compare Match A/B Register Rewrite Operation

Regardless of the operating mode, the timing of the rewrite operation to the reload register and AGT compare register A/B depends on the value of the TSTART bit in the AGTCR register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and AGT compare register A/B. When the TSTART bit is 1 (count starts), the value is written to the reload register in synchronization with the count source, and then to the compare register in synchronization with the underflow of the counter.

Figure 22.5 shows the timing of rewrite operation with TSTART bit value for compare register A. AGT Compare register B is of the same timing as AGT compare register A.

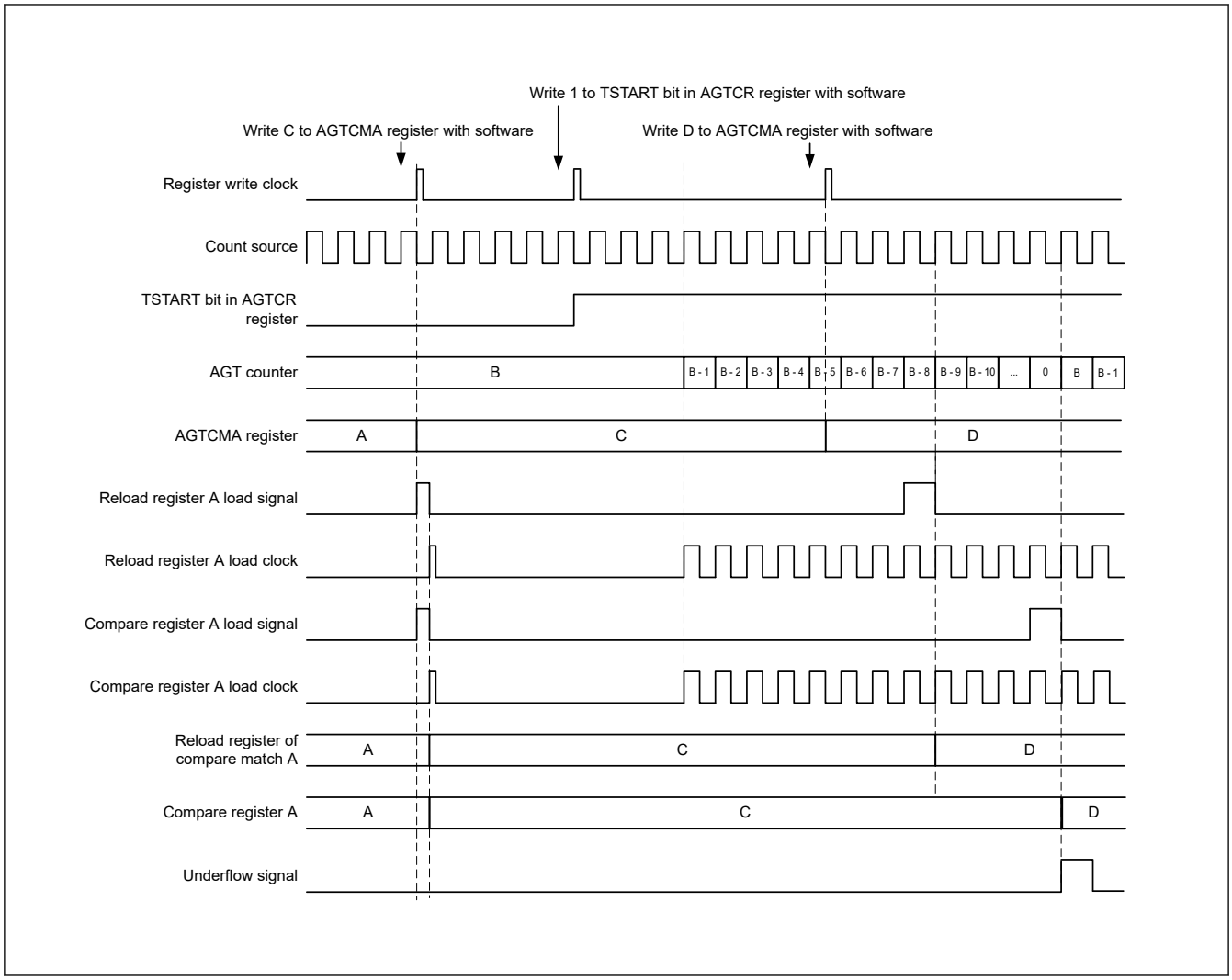


Figure 22.5 Timing of rewrite operation with the TSTART bit value for AGT compare register A

22.3.3 Timer Mode

In this mode, the AGT counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register. In timer mode, the count value is decremented by 1 on each rising edge of the count source. When the count value reaches 0x0000 and the next count source is input, an underflow occurs and an interrupt request is generated.

Figure 22.6 shows the operation example in timer mode.

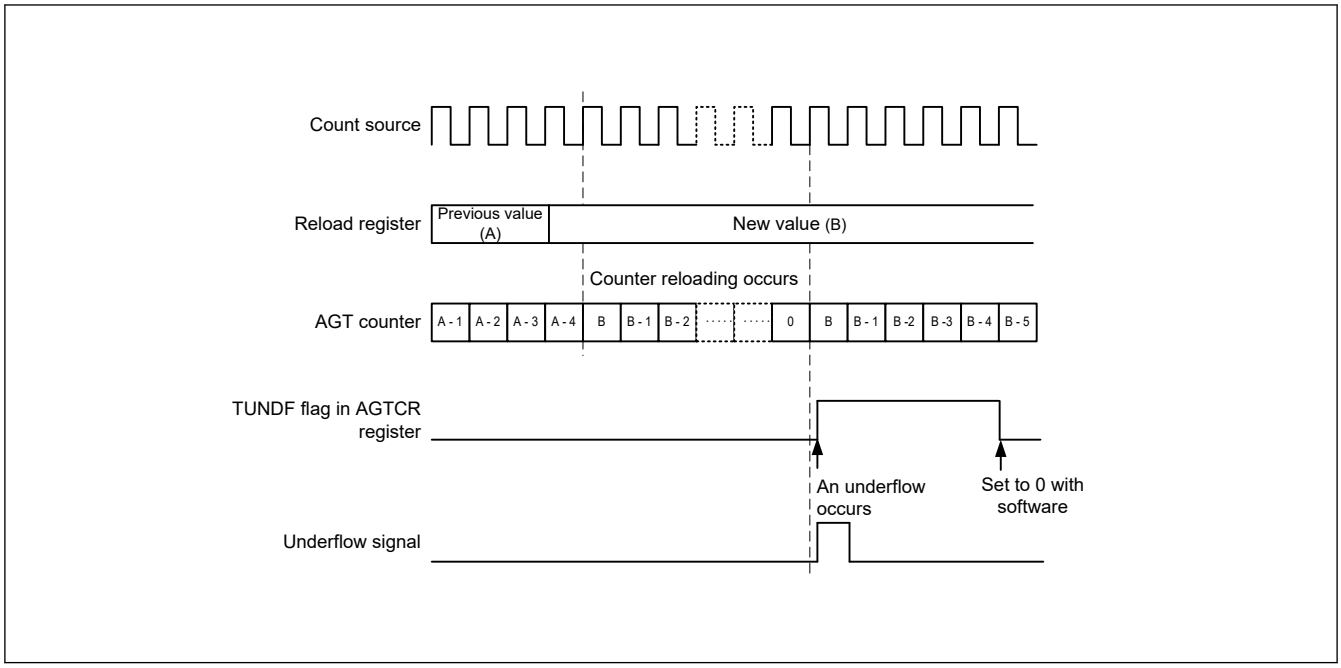


Figure 22.6 Operation example in timer mode

22.3.4 Pulse Output Mode

In pulse output mode, the counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register, and the output level of the AGTIOn and AGTOn pins inverted each time an underflow occurs.

In pulse output mode, the count value is decremented by 1 on each rising edge of the count source. When the count value reaches 0x0000 and the next count source is input, an underflow occurs and an interrupt request is generated. In addition, a pulse can be output from the AGTIOn and AGTOn pins. The output level is inverted each time an underflow occurs. The pulse output from the AGTOn pin can be stopped with the TOE bit in the AGTIOC register. The output level can be selected with the TEDGSEL bit in the AGTIOC register.

Figure 22.7 shows the operation example in pulse output mode.

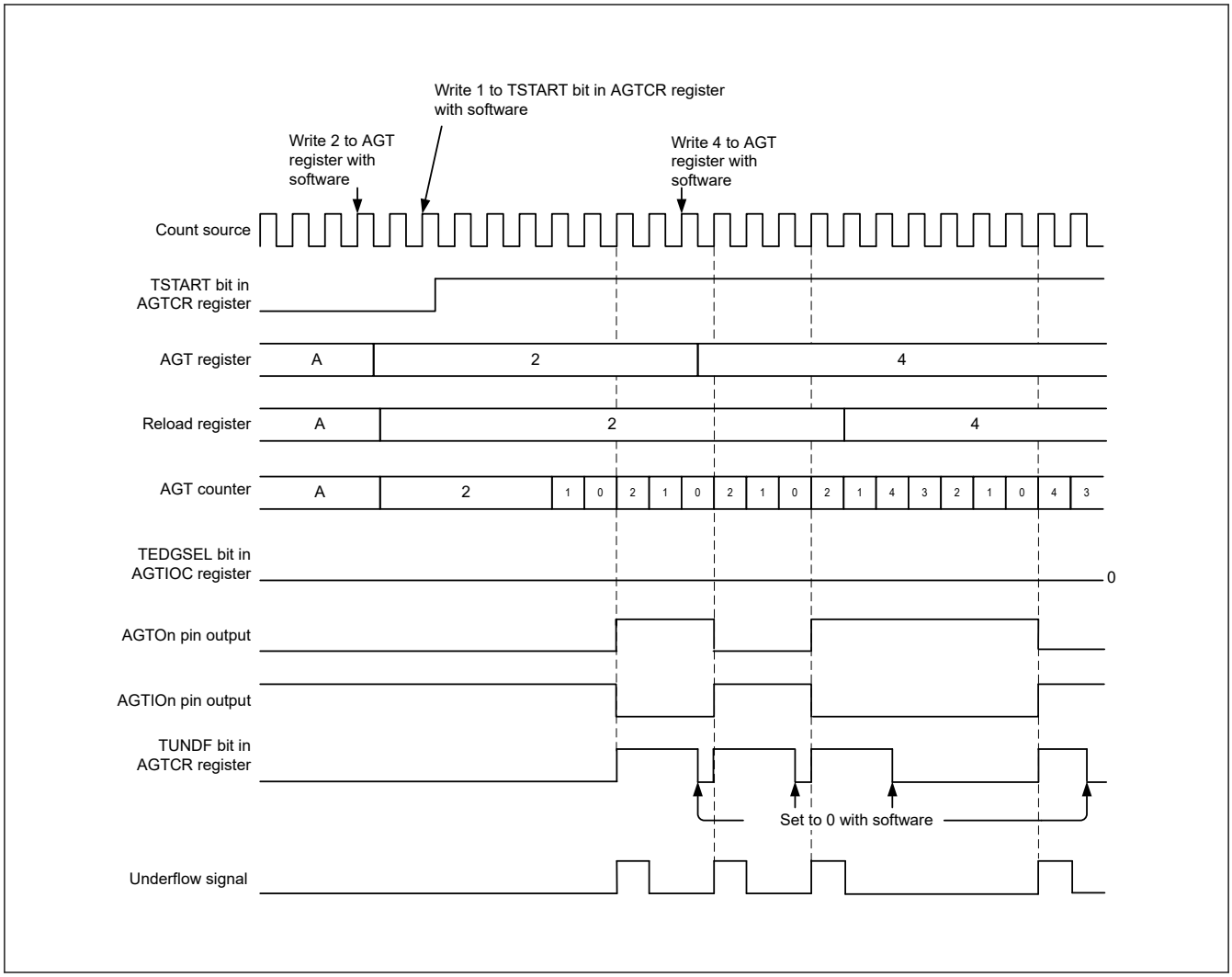


Figure 22.7 Operation example in pulse output mode

22.3.5 Event Counter Mode

In event counter mode, the counter is decremented by an external event signal (count source) input to the AGTIOn pin. Various periods for counting events can be set with the TIOGT[1:0] bits in the AGTIOC register and AGTISR registers. In addition, the filter function for the AGTIOn pin input can be specified with bits TIPF[1:0] in the AGTIOC register. The output from the AGTOn pin can be toggled even in event counter mode.

Figure 22.8 shows the operation example in event counter mode.

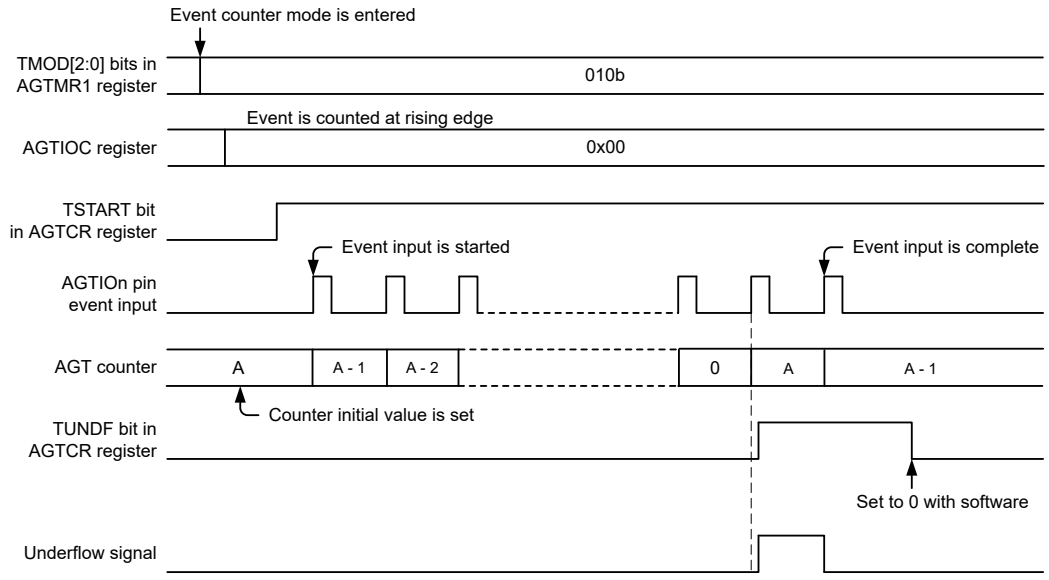
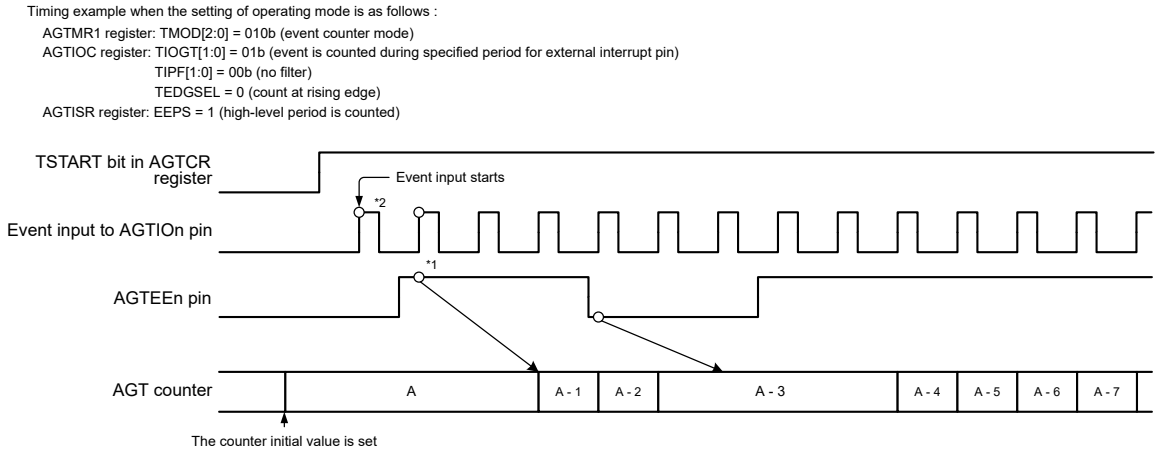


Figure 22.8 Operation example 1 in event counter mode

Figure 22.9 shows an operation example for counting during the specified period in event counter mode (TIOGT[1:0] bits in the AGTIOC register are set to 01b).



Note 1. To control synchronization, there is a delay of 2 cycles of the count source until the count operation is affected. It is also possible that the count start timing is shifted by 1 cycle because of the phase difference between the AGTEEn pin and the sampling clock.

Note 2. Count operation can be performed for 2 cycles of the count source immediately after the count starts, depending on the previous state before the count stops.
To disable the count for 2 cycles immediately after the count starts, write 1 to the TSTOP bit in the AGTCR register to initialize the internal circuit, and then complete the operation settings before starting the count operation.

Figure 22.9 Operation example 2 in event counter mode

22.3.6 Pulse Width Measurement Mode

In pulse width measurement mode, the pulse width of an external signal input to the AGTIOn pin is measured. When the level specified by the TEDGSEL bit in the AGTIOC register is input to the AGTIOn pin, the counter is decremented by the

count source selected with the TCK[2:0] bits in the AGTMR1 register. When the specified level on the AGTIOn pin ends, the counter is stopped, the TEDGF flag in the AGTCR register is set to 1 (active edge received), and an interrupt request is generated. The measurement of pulse width data is performed by reading the count value while the counter is stopped. Also, when the counter underflows during measurement, the TUNDF flag in the AGTCR register is set to 1 and an interrupt request is generated.

Figure 22.10 shows the operation example in pulse width measurement mode.

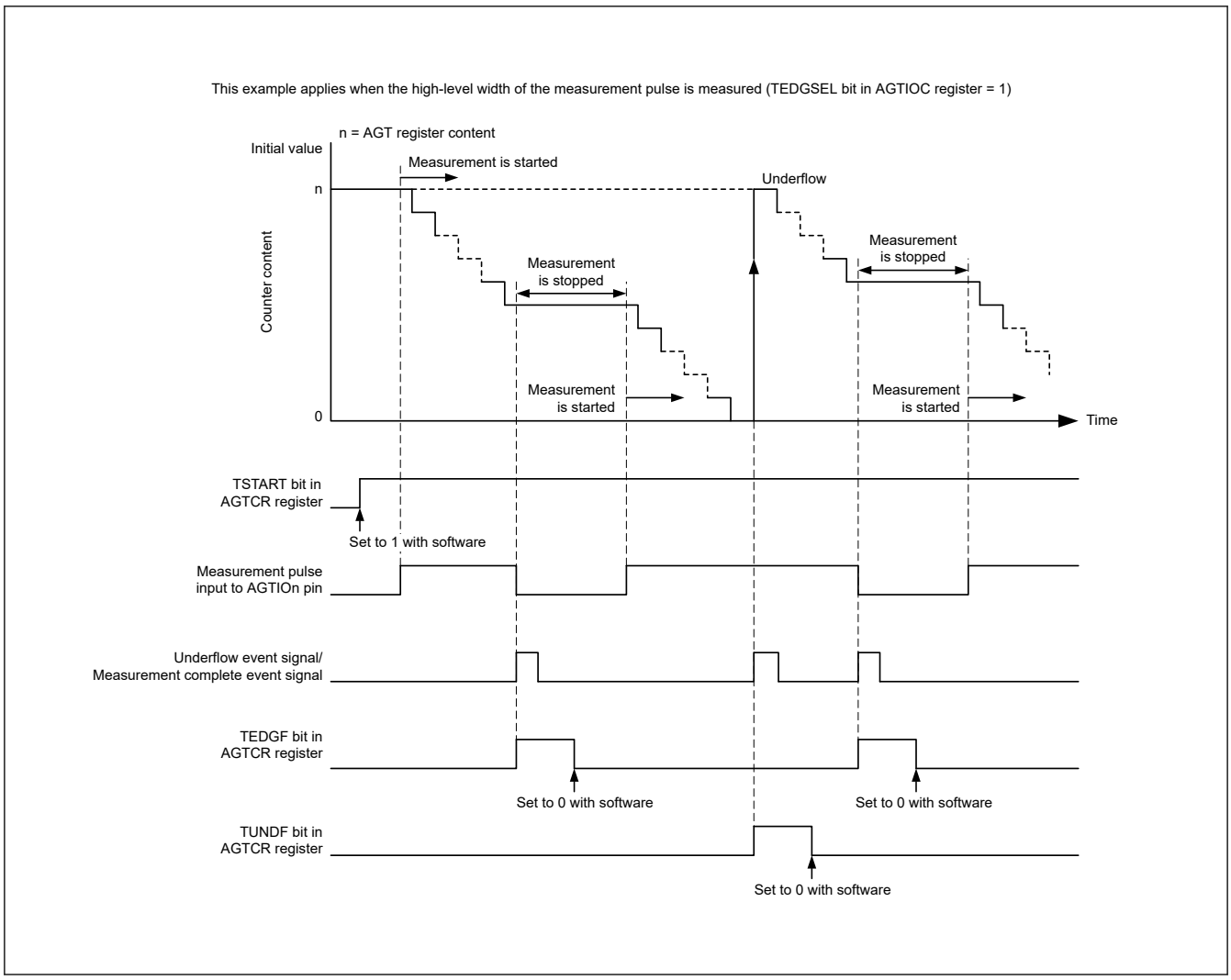


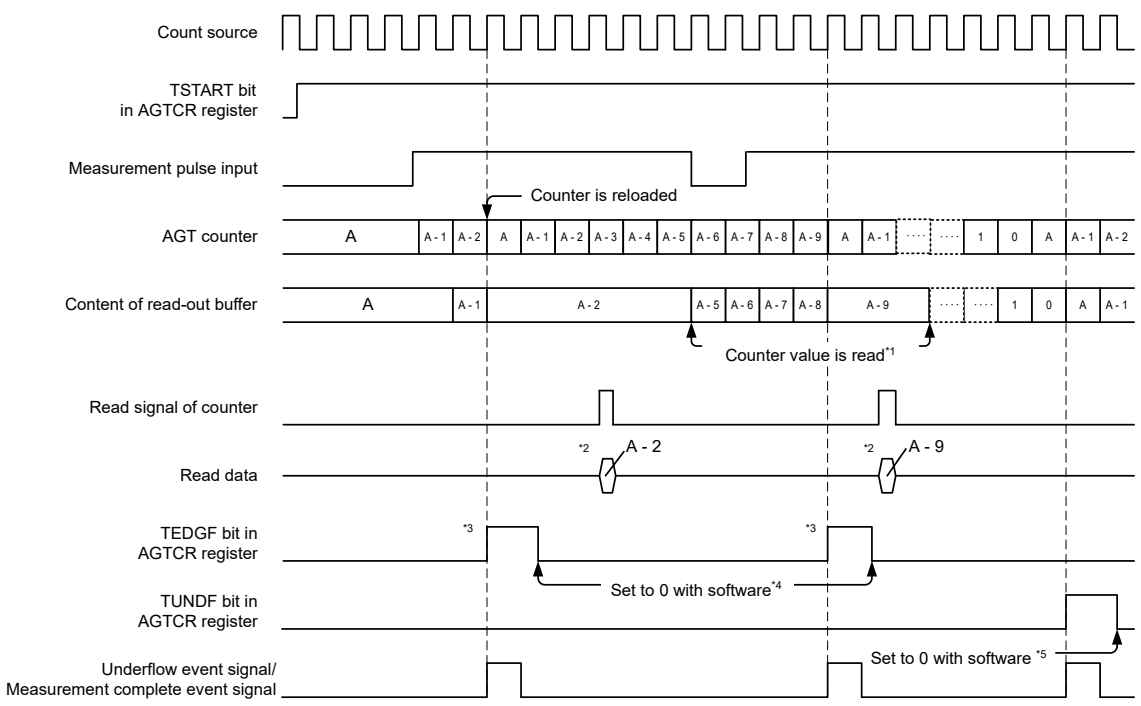
Figure 22.10 Operation example in pulse width measurement mode

22.3.7 Pulse Period Measurement Mode

In pulse period measurement mode, the pulse period of an external signal input to the AGTIOn pin is measured. The counter is decremented by the count source selected with TCK[2:0] bits in the AGTMR1 register. When a pulse with the period specified by the TEDGSEL bit in the AGTIOC register is input to the AGTIOn pin, the count value is transferred to the read-out buffer on the rising edge of the count source. The value in the reload register is loaded to the counter at the next rising edge. Simultaneously, the TEDGF flag in the AGTCR register is set to 1 (active edge received) and an interrupt request is generated. The read-out buffer (AGT register) is read at this time and the difference from the reload value (see section 22.4.6. How to Calculate Event Number, Pulse Width, and Pulse Period) is the period data of the input pulse. The period data is retained until the read-out buffer is read. When the counter underflows, the TUNDF flag in the AGTCR register is set to 1 (underflow) and an interrupt request is generated.

Figure 22.11 shows the operation example in pulse period measurement mode.

Only input pulses with a period longer than twice the period of the count source are measured. Also, the low-level and high-level widths must both be longer than the period of the count source. If a pulse period shorter than these conditions is input, the input might be ignored.



This example applies when the initial value of the AGT register is set to A, the TEDGSEL bit in the AGTIOC register is set to 0, and the period from one rising edge to the next edge of the measurement pulse is measured.

Note 1. Reading from the AGT register must be performed during the period from when the TEDGF flag is set to 1 (active edge received) until the next active edge is input. The content of the read-out buffer is retained until the AGT register is read. If it is not read before the active edge is input, the measurement result of the previous period is retained.
Note 2. When the AGT register is read in pulse period measurement mode, the content of the read-out buffer is read.
Note 3. When the active edge of the measurement pulse is input and then the set edge of an external pulse is input, the TEDGF flag in the AGTCR register is set to 1 (active edge received).
Note 4. To set to 0 with software, write 0 to the TEDGF flag in the AGTCR register with an 8-bit memory manipulation instruction.
Note 5. To set to 0 with software, write 0 to the TUNDF flag in the AGTCR register with an 8-bit memory manipulation instruction.

Figure 22.11 Operation example in pulse period measurement mode

22.3.8 Compare Match function

The compare match function detects matches (compare match) between the content of the AGTCMA or AGTCMB register and the content of the AGT register. This function is enabled when the TCMEA or TCMEB bit in the AGTCMSR register is 1 (compare match A register or compare match B register is valid). The counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register, and when the values of AGT and AGTCMA or AGTCMB match, the TCMAF/TCMBF flag in the AGTCR register is set to 1 (match), and an interrupt request is generated.

When the compare match function is enabled, the timing of the rewrite operation to the reload register and the counter differs. See section 22.3.1. Reload Register and Counter Rewrite Operation for details. In addition, the output level of the AGTOAn, AGTOBn pins is inverted by the match and by the underflow. The output level can be selected with the TOPOLA or TOPOLB bit in the AGTCMSR register.

Figure 22.12 shows the operation example in compare match function.

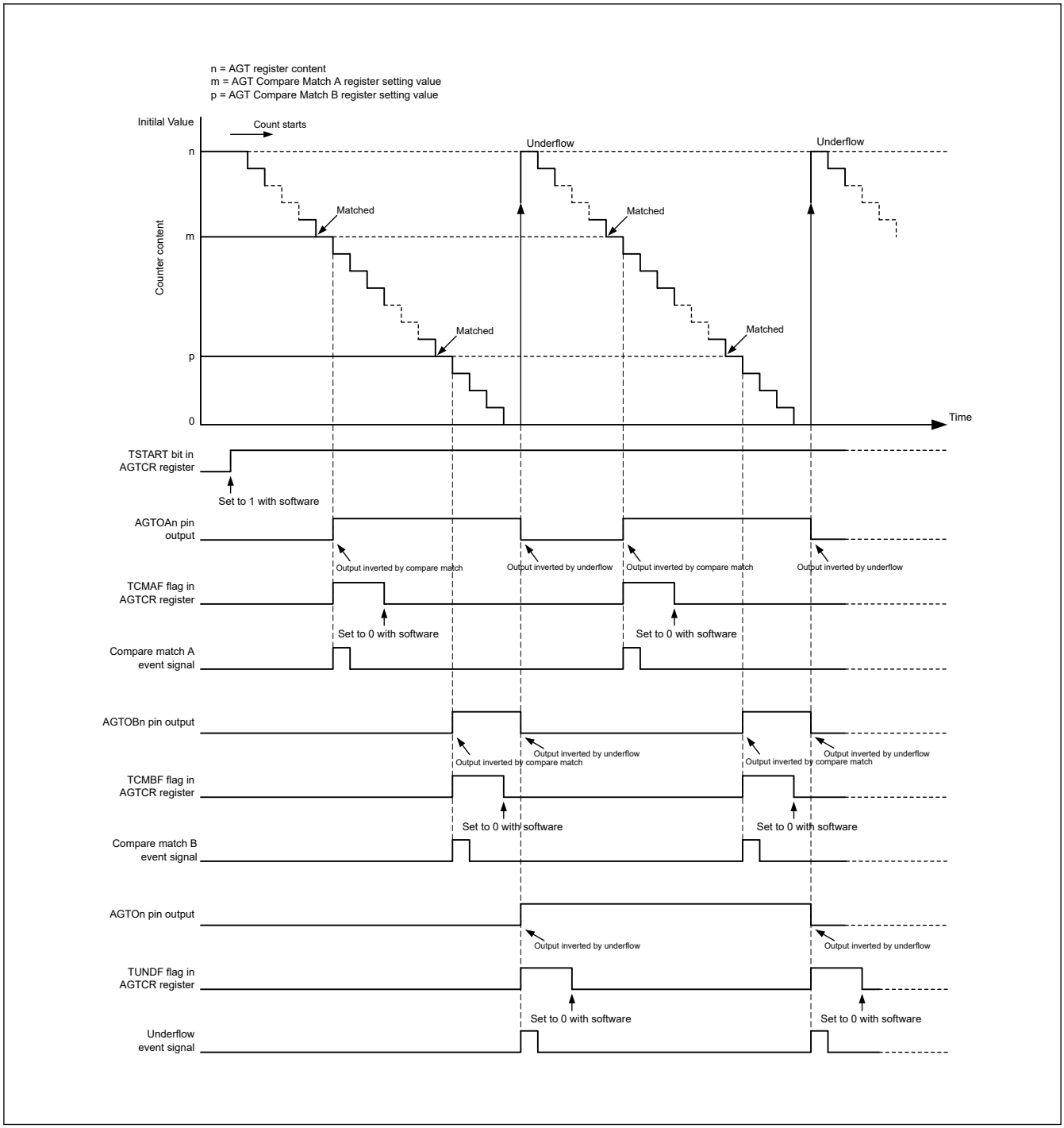


Figure 22.12 Operation example in compare match function (TOPOLA = 0, TOPOLB = 0)

22.3.9 Output Settings for Each Mode

Table 22.5 to Table 22.8 list the states of pins AGTOn, AGTIOn, AGTOAn, and AGTOBn pins in each mode.

Table 22.5 AGTOn pin setting

Operating mode	AGTIOC register		AGTOn pin output
	TOE bit	TEDGSEL bit	
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

Table 22.6 AGTIOn pin setting

Operating mode	AGTIOC register TEDGSEL bit	AGTIOn pin I/O
Timer mode	0 or 1	Input (not used)
Pulse output mode	1	Normal output
	0	Inverted output
Event counter mode	0 or 1	Input
Pulse width measurement mode		
Pulse period measurement mode		

Table 22.7 AGTOAn pin setting

Operating mode	AGTCMSR register		AGTOAn pin output
	TOEA bit	TOPOLA bit	
Timer mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse output mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Event counter mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse width measurement mode	0	0	Prohibited
Pulse period measurement mode			

Table 22.8 AGTOBn pin setting

Operating mode	AGTCMSR register		AGTOBn pin output
	TOEB bit	TOPOLB bit	
Timer mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse output mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Event counter mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse width measurement mode	0	0	Prohibited
Pulse period measurement mode			

22.3.10 Standby Mode

The AGT can operate in Software Standby mode. Set it to Software Standby mode with count operation start (TSTART = 1, TCSTF = 1).

Table 22.9 and Table 22.10 show the setting that can be used in Software Standby mode.

Table 22.9 Usable settings in Software Standby mode (AGT0)

Operating mode	AGTMR1.TCK[2:0]	Operating clock	Resurgence factor of CPU
Timer mode	100b or 110b	AGTLCLK or AGTSCLK	—
Pulse output mode	100b or 110b	AGTLCLK or AGTSCLK	—
Event counter mode*2*3*4	—	AGTIO0*1	—
Pulse width measurement mode	100b or 110b	AGTLCLK or AGTSCLK	—
Pulse period measurement mode	100b or 110b	AGTLCLK or AGTSCLK	—

Note: —: invalid
Note 1. When using the AGTIO0 pin for external event input in Software Standby mode, set AGTIOSEL.TIES = 1.
Note 2. AGTEE pin is not available during Software Standby mode. External events are always enabled.
Note 3. When AGTIOSEL.TIES = 0 and the external event input signal is disabled during Software Standby mode, stop the count operation before entering Software Standby mode. After returning from Software Standby mode, restart the count operation if necessary.
Note 4. When external event input is disabled during Software Standby mode (AGTIOSEL.TIES = 0), stop the count operation before the MCU enters Software Standby mode. Restart the count operation after the MCU returns from Software Standby mode.

Table 22.10 Usable settings in Software Standby mode (AGT1)

Operating mode	AGTMR1.TCK[2:0]	Operating clock	Resurgence factor of CPU
Timer mode	100b or 110b or 101b*1	AGTLCLK or AGTSCLK or AGT0 underflow	• Underflow • Compare match A/B
Pulse output mode	100b or 110b or 101b*1	AGTLCLK or AGTSCLK or AGT0 underflow	• Underflow • Compare match A/B
Event counter mode*3*4*5	—	AGTIO1*2	• Underflow • Compare match A/B
Pulse width measurement mode	100b or 110b or 101b*1	AGTLCLK or AGTSCLK or AGT0 underflow	• Underflow • Active edge
Pulse period measurement mode	100b or 110b or 101b*1	AGTLCLK or AGTSCLK or AGT0 underflow	• Underflow • Active edge

Note: —: invalid
Note: Release of Software Standby mode is only AGT1.
Note: Compare match A/B is resurgence factor of CPU from Software Standby mode.
Note 1. Only when AGT0 operates in Table 22.9
Note 2. When using the AGTIO1 pin for external event input in Software Standby mode, set AGTIOSEL.TIES = 1.
Note 3. AGTEE pin is not available during Software Standby mode. External events are always enabled
Note 4. When AGTIOSEL.TIES = 0 and the external event input signal is disabled during Software Standby mode, stop the count operation before entering Software Standby mode. After returning from Software Standby mode, restart the count operation if necessary.
Note 5. When external event input is disabled during Software Standby mode(AGTIOSEL.TIES = 0), stop the count operation before the MCU enters Software Standby mode. Restart the count operation after the MCU returns from Software Standby mode.

22.3.11 Interrupt Sources

The AGTn has three interrupt sources as listed in Table 22.11.

Table 22.11 AGT interrupt sources

Name	Interrupt source	DMAC/DTC activation
AGTn_AGTI	• When the counter underflows • When measurement of the active width of the external input pin (AGTIOn) is complete in pulse width measurement mode • When the set edge of the external input pin (AGTIOn) is input in pulse period measurement mode.	Possible
AGTn_AGTCMAI	• When the values of AGT register and AGTCMA register match	Possible
AGTn_AGTCMBI	• When the values of AGT register and AGTCMB register match	Possible

Note: Channel number (n = 0, 1)

22.3.12 Event Signal Output to ELC

The AGT uses the Event Link Controller (ELC) to perform a link operation to a specified module using the interrupt request signal as the event signal. The AGT outputs compare match A, compare match B, and underflow/measurement complete signals as event signals. For details, see section 18, Event Link Controller (ELC).

22.4 Usage Notes

22.4.1 Count Operation Start and Stop Control

- When the operating mode (see Table 22.1) is set to other than the event counter mode, or the count source is set to other than AGTn underflow event signal (TCK[2:0] = 101b):
  - After 1 (count starts) is written to the TSTART bit in the AGTCR register while the count is stopped, the TCSTF flag in the AGTCR register remains 0 (count stops) for 3 cycles of the count source. Do not access the registers associated with AGT other than the TCSTF flag until this flag is set to 1 (count in progress).
  - After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF flag remains 1 for 3 cycles of the count source. When the TCSTF flag is set to 0, the count is stopped. Do not access the registers associated with AGT other than the TCSTF flag until this flag is set to 0.
- When the operating mode (see Table 22.1) is set to event counter mode, or the count source is set to AGT1 underflow event signal (TCK[2:0] = 101b):
  - After 1 (count starts) is written to the TSTART bit in the AGTCR register while the count is stopped, the TCSTF flag in the AGTCR register remains 0 (count stops) for 2 PCLKB cycles. Do not access the registers associated with AGT other than the TCSTF flag until this flag is set to 1 (count in progress).
  - After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF flag remains 1 for 2 PCLKB cycles. When the TCSTF flag is set to 0, the count is stopped. Do not access the registers associated with AGT other than the TCSTF flag until this flag is set to 0.

22.4.2 Access to Counter Register

When the TSTART bit and TCSTF flag in the AGTCR register are both 1 (count starts), allow at least 3 cycles of the count source clock between writes when writing to the AGT register successively.

22.4.3 When Changing Mode

The registers associated with AGT operating mode (AGTMR1, AGTMR2, AGTIOC, AGTISR, and AGTCMSR) can be changed only when the count is stopped with both the TSTART bit and TCSTF flag set to 0 (count stops). Do not change these registers during count operation.

When the registers associated with AGT operating mode are changed, the values of TEDGF, TUNDF, TCMAF, and TCMBF flags are undefined. Before starting the count, write 0 to the following flags:

- TEDGF (no active edge received)
- TUNDF (no underflow)
- TCMAF (no match)
- TCMBF (no match).

22.4.4 Output pin setting

When using the AGTOn, AGTIOn, AGTOAn, or AGTOBn as an output pin, set up the Operation and determine the initial output values. Then set PmnPFS.PMR bit to 1.

When using the AGTIOn as an input pin in pulse width measurement mode or pulse period measurement mode, set up the Operation and start count operation. Then start to enter external events from the AGTIOn pin. Invalidate the first measurement and validate the second and later completed measurements.

22.4.5 Digital Filter

When using the digital filter, do not start the timer operation for 5 cycles of the digital filter clock after setting TIPF[1:0] bits and when the TEDGSEL bit in the AGTIOC register changes.

22.4.6 How to Calculate Event Number, Pulse Width, and Pulse Period

- In event counter mode, event number is expressed mathematically as follows:
  Event number = initial value of counter [AGT register] - counter value of active event end

- In pulse width measurement mode, pulse width is expressed mathematically as follows:
  Pulse width = counter value of stopping measurement - counter value of next stopping measurement
- In pulse period measurement mode, input pulse period is expressed mathematically as follows:
  Period of input pulse = (initial value of counter [AGT register] - reading value of the read-out buffer) + 1.

22.4.7 When Count is Forcibly Stopped by TSTOP Bit

After the counter is forcibly stopped by the TSTOP bit in the AGTCR register, do not access the following I/O registers for 1 cycle of the count source:

- AGT
- AGTCMA
- AGTCMB
- AGTCR
- AGTMR1
- AGTMR2.

22.4.8 When Selecting AGT0 Underflow as the Count Source

Operate according to the following procedures described in this section when selecting the underflow event signal as the count source.

(1) Procedure for starting operation

1. Set AGT.
2. Start the count operation of AGT1.
3. Start the count operation of AGT0.

(2) Procedure for stopping operation

1. Stop the count operation of AGT0.
2. Stop the count operation of AGT1.
3. Stop the count source clock of AGT1 (write 000b in the AGTMR1.TCK[2:0] bits).

22.4.9 Module-stop function

AGT operation can be disabled or enabled using Module Stop Control Register D (MSTPCRD). The AGT module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes

22.4.10 When Switching Source Clock

When switching a clock source by changing SCKSCR.CKSEL[2:0], the clock output from the selector stops for 4 cycles of the switched clock. Therefore, when using the AGTIOn, AGTEEn, or both input as external event input, the clock source should not be switched. If switching the clock source while using the external event input, extend the input pulse width by 4 clock cycles of the switched source clock cycles.

23. Ultra-Low-Power Timer (ULPT)

23.1 Overview

This ultra-low-power timer (ULPT) is a 32-bit timer that can output pulses or count external events.

This 32-bit timer consists of reload registers and a down-counter. The reload registers and the down-counter are allocated to the same address and can be accessed through the ULPTCNT register.

Table 23.1 lists the ULPT specifications.

Table 23.1 ULPT specifications

Item		Description
Operating mode	Timer mode	Count the internal clock.
	Event counter mode	Count external events.
Number of Channels		32 bits × 2 channels (ULPTn (n = 0, 1))
Count source (operating clock)*1		The following signals can be selected: ● Internal clock: ULPTLCLK/d or ULPTSCLK/d (d = 1, 2, 4, 8, 16, 32, 64, 128) ● External event: ULPTEVIn pin
Count operation		The following functions can be selected: ● Selection between continuous mode and one-shot mode ● Selection among count enable mode, count start mode, and count restart mode
Interrupt/event link function (output)		● Underflow signal – When the counter underflows ● Compare match A signal – When the ULPTCNT value matches the ULPTCMA value (The compare match A function is enabled.) ● Compare match B signal – When the ULPTCNT value matches the ULPTCMB value (The compare match B function is enabled.)
Selectable functions		● Pulse output function – Inverts the output each time the counter underflows. ● Compare match function – Either or both compare match A register and compare match B register are selectable. ● Return from Software Standby mode can be performed with ULPTn_ULPTI, ULPTn_ULPTCMAI, or ULPTn_ULPTCMBI (n=0, 1).*2 ● Return from Deep Software Standby mode 1 can be performed with ULPTn_ULPTI (n=0, 1).*2
Module-stop function		Module-stop state can be set for each channels to reduce power consumption
TrustZone Filter		Security and Privilege attribution can be set for each channels.

Note 1. Make sure that the frequency of the peripheral module clock (PCLKB) is at least twice the frequency of the count source clock (AGTLCLK/d or AGTSCLK/d).
Note 2. For details, see section 10, Low Power Modes.

23.2 Configuration

Figure 23.1 shows the ULPT block diagram. Table 23.2 lists the I/O pins of the ULPT.

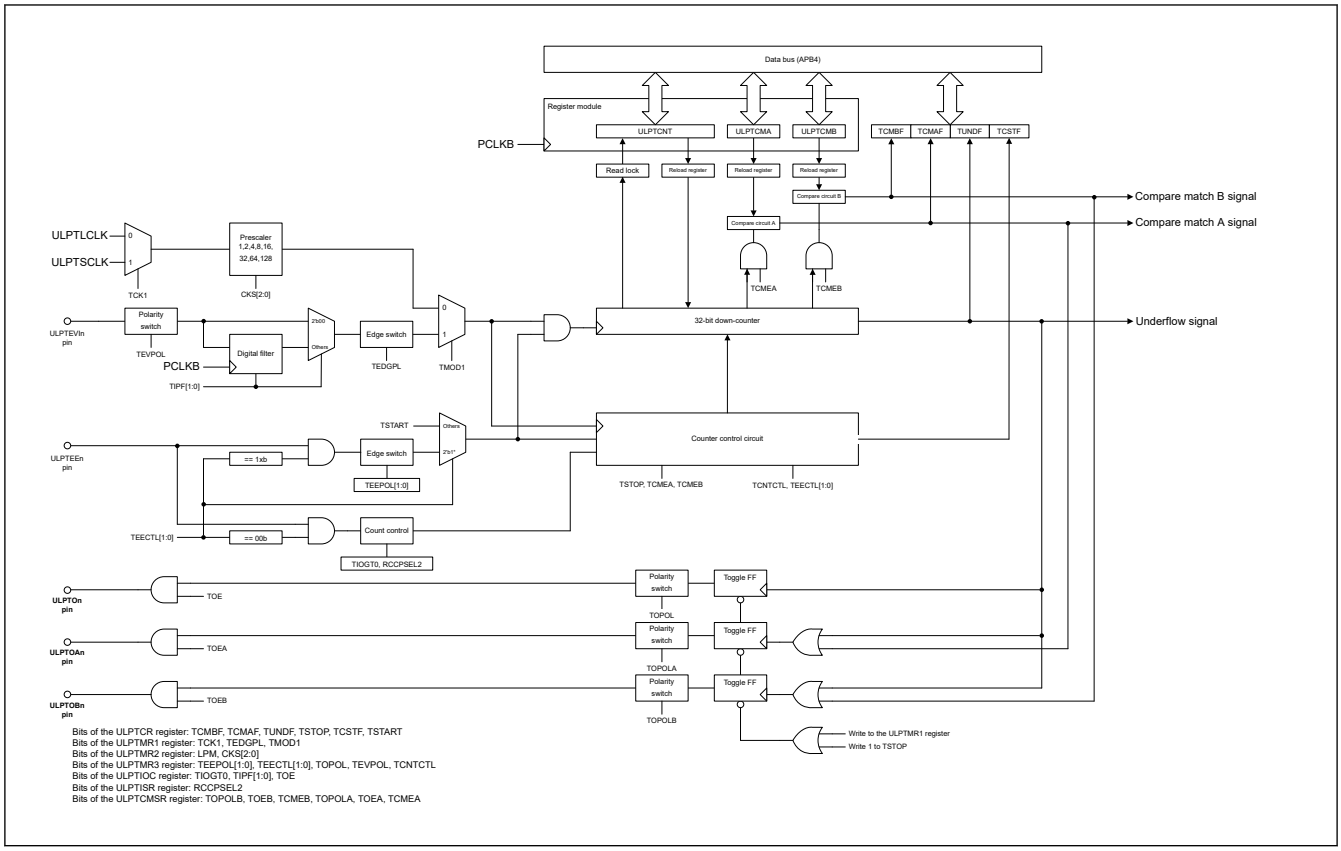


Figure 23.1 ULPT block diagram

Table 23.2 ULPT I/O pins

Pin name	I/O	Function
ULPTEEn*2	Input	External count control input*1
ULPTEVIn*2	Input	External event input
ULPTOn*2	Output	Pulse output
ULPTOAn*2	Output	Output compare match A output
ULPTOBn*2	Output	Output compare match B output

Note: Channel number: n = 0, 1

Note 1. The function of the ULPTEEn pin can be selected by the ULPTMR3.TEECTL[1:0] bits. For more details, see section 23.3.7. ULPTMR3 : ULPT Mode Register 3.

Note 2. Only pins marked "-DS" can be used in Deep Software Standby mode1.

23.3 Register Descriptions

23.3.1 ULPTCNT : ULPT Counter Register

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x00

Bit position:	31																												0
Bit field:	ULPTCNT																												
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	ULPTCNT	32-bit counter and reload register*1*2 Setting range : 0x00000000 to 0xFFFFFFFF	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. When 1 is written to the ULPTCR.TSTOP bit, the counter is forcibly stopped and set to 0xFFFFFFFF. The reload register is set to 0xFFFFFFFF
Note 2. When 0x00000000 is set in the ULPTCNT register, a request signal to the ICU, DTC, and ELC occurs only once immediately after the start of count operation (when the ULPTCR.TSTART bit is set to 1). However, the pulse output at the ULPTOn pin continues to be toggled.
The pulse output from the ULPTOn pin can be enabled or disabled by the ULPTIOC.TOE bit.
When the ULPTCNT register is set to a value equal to or greater than 0x00000001, a request signal is generated each time the counter underflows.

The ULPTCNT is a 32-bit register. The register value is written to the reload register and is read from the counter.

The states of the counter and reload register change according to the TSTART/TCSTF bit in the ULPTCR register and TCMEA/TCMEB bit in the ULPTCMSR register. For details, see section 23.4.2. Rewriting the counter and reload register.

The ULPTCNT register is only available for 32-bit read/write access.

23.3.2 ULPTCMA : ULPT Compare Match A Register

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x04

Bit position:	31																													0
Bit field:	ULPTCMA																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	ULPTCMA	32-bit Compare Match A Data*1*2 Setting range : 0x00000000 to 0xFFFFFFFF	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. When 1 is written to the ULPTCR.TSTOP bit, 0xFFFFFFFF is set in the ULPTCMA register.
Note 2. Set the ULPTCMA register to 0xFFFFFFFF when compare match A is not used (ULPTCMSR.TCMEA bit = 0).

The ULPTCMA register is a readable/writable register to set a value for compare match with the counter.

The states of compare circuit A and the reload register change according to the TSTART and TCSTF bits in the ULPTCR register. For details, see section 23.4.3. Rewriting the compare circuits and reload registers for compare match A/B.

The ULPTCMA register is only available for 32-bit read/write access.

23.3.3 ULPTCMB : ULPT Compare Match B Register

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x08

Bit position:	31																													0
Bit field:	ULPTCMB																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	ULPTCMB	32-bit Compare Match B Data*1*2 Setting range : 0x00000000 to 0xFFFFFFFF	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. When 1 is written to the ULPTCR.TSTOP bit, 0xFFFFFFFF is set in the ULPTCMB register.
Note 2. Set the ULPTCMB register to 0xFFFFFFFF when compare match B is not used (ULPTCMSR.TCMEB bit = 0).

The ULPTCMB register is a readable/writable register to set a value for compare match with the counter.

The states of compare circuit B and the reload register change according to the TSTART and TCSTF bits in the ULPTCR register. For details, see section 23.4.3. Rewriting the compare circuits and reload registers for compare match A/B.

The ULPTCMB register is only available for 32-bit read/write access.

23.3.4 ULPTCR : ULPT Control Register

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TCMBF	TCMAF	TUNDF	—	—	TSTOP	TCSTF	TSTART
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSTART	Counter Start 0: Stop the counter. 1: Start the counter.	R/W
1	TCSTF	Counter Status Flag*2 0: Counter stopped 1: Counter running	R
2	TSTOP	Counter Forcible Stop*1 0: Writing is invalid. 1: Stop the counter forcibly.	W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	TUNDF	Underflow Flag 0: No underflow occurred (counter ≠ 0x00000000). 1: Underflow occurred (counter = 0x00000000).	R/W*3
6	TCMAF	Compare Match A Flag 0: Not matched (counter ≠ ULPTCMA[31:0]) 1: Matched (counter = ULPTCMA[31:0])	R/W*3
7	TCMBF	Compare Match B Flag 0: Not matched (counter ≠ ULPTCMB[31:0]) 1: Matched (counter = ULPTCMB[31:0])	R/W*3

Note: S-TYPE-3, P-TYPE-3
Note 1. When 1 (stop the counter forcibly) is written to the TSTOP bit, the TSTOP, TSTART, and TCSTF bits are initialized. The pulse output, compare match A output and compare match B output are also initialized. This bit is read as 0.
Note 2. For precautions on using the TCSTF bit, see section 23.5.1. Start and stop control of the counter.
Note 3. Only 0 can be written to clear the flag.

The ULPTCR register controls the start and stop of the counter and indicates the status of the ULPT.

The ULPTCR register is only available for 8-bit read/write access.

It is recommended to set 1 to the TUNDF, TCMAF and TCMBF bits when changing the TSTART or TSTOP bits to prevent clearing TUNDF, TCMAF, or TCMBF by mistake. At this time, setting 1 to TUNDF, TCMAF and TCMBF bits is ignored.

TSTART bit (Counter Start)

When and how the counter is started depends on the setting of the ULPTMR3.TEECTL[1:0] bits.

The counter is stopped in the same way in any mode.

For details, see section 23.4.1. Count operation.

[Starting the counter]

- In count enable mode (ULPTMR3.TEECTL[1:0] bits = 00)
  When 1 is written to the TSTART bit.
- In count start mode or count restart mode (ULPTMR3.TEECTL[1:0] bits = 10 or 11)

When 1 is written to the TSTART bit, and then an edge at the ULPTEEn pin is input.

[Stopping the counter]

- When 0 is written to the TSTART bit.
- When 1 is written to the TSTOP bit.

TCSTF bit (Counter Status Flag)

The operation of this bit depends on the settings of the TCNTCTL and TEECTL[1:0] bits of the ULPTMR3 register.

For details, see section 23.4.1. Count operation.

[Setting conditions]

- In timer mode (ULPTMR1.TMOD1 = 0) and
  count enable mode (ULPTMR3.TEECTL[1:0] bits = 00)
  When 1 is written to the TSTART bit (with the TCSTF bit set to 1 in synchronization with the count source).
- In event counter mode (ULPTMR1.TMOD1 = 1) and
  count enable mode (ULPTMR3.TEECTL[1:0] bits = 00)
  When 1 is written to the TSTART bit (with the TCSTF bit set to 1 in synchronization with the count source).
- In count start mode or count restart mode (ULPTMR3.TEECTL[1:0] bits = 10 or 11)
  When 1 is written to the TSTART bit, and then an edge at the ULPTEEn pin is input
  (with the TCSTF bit set to 1 in synchronization with the count source).

[Clearing conditions]

- When 0 is written to the TSTART bit (with the TCSTF bit set to 0 in synchronization with the count source).
- When 1 is written to the TSTOP bit.
- When the counter underflows in one-shot mode (ULPTMR3.TCNTCTL = 1).
  Only under this condition, the TSTART bit does not change from 1.

TSTOP bit (Counter Forcible Stop)

When 1 is written to this bit, the counter is forcibly stopped. This bit is read as 0.

TUNDF bit (Underflow Flag)

[Setting condition]

- When the counter underflows (counter = 0x00000000).

[Clearing condition]

- When 0 is written to this bit.

TCMAF bit (Compare Match A Flag)

[Setting condition]

- When the ULPTCNT register value matches the ULPTCMA register value (counter = ULPTCMA[31:0]).

[Clearing condition]

- When 0 is written to this bit.

TCMBF bit (Compare Match B Flag)

[Setting condition]

- When the ULPTCNT register value matches the ULPTCMB register value (counter = ULPTCMB[31:0]).

[Clearing condition]

- When 0 is written to this bit.

23.3.5 ULPTMR1 : ULPT Mode Register 1

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	TCK1	—	TEDGPL	—	TMOD1	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	TMOD1	Operating Mode*1 0: Timer mode 1: Event counter mode	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	TEDGPL	ULPTEVIn Edge Polarity 0: Either edge (rising) 1: Both edges This bit is valid only in event counter mode (TMOD1 = 1).	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	TCK1	Count Source*2 0: Divided clock specified by the ULPTMR2.CKS[2:0] bits (ULPTLCLK) 1: Divided clock specified by the ULPTMR2.CKS[2:0] bits (ULPTSCLK) This bit is valid only in timer mode (TMOD1 = 0).	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Do not rewrite this register while the counter is running. Rewrite this register only when the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0).
Note: A write access to this register initializes the output from pins ULPTOn, ULPTOAn, and ULPTOBn, For details on the output level at initialization, see section 23.3.7. ULPTMR3 : ULPT Mode Register 3 and section 23.3.8. ULPTIOC : ULPT I/O Control Register.
Note 1. Timer mode: The internal clock (ULPTLCLK or ULPTSCLK) is used as the count source. The counter is decremented by 1 on each rising edge of the count source.
Event counter mode: The external event input (ULPTEVIn pin) is used as the count source. The counter is decremented on each edge of the count source. The edge polarity can be selected by the TEDGPL bit and the ULPTMR3.TEVPOL bit.
For details, see section 23.4.1. Count operation.
Note 2. The TCK1 bit can be changed only when the ULPTMR2.CKS[2:0] bits are 000. Otherwise, do not change the bit. Set the ULPTMR2.CKS[2:0] bits to 000, and then change the TCK1 bit after one cycle of the count source.

The ULPTMR1 register sets the operating mode of the ULPT.

The ULPTMR1 register is only available for 8-bit read/write access.

23.3.6 ULPTMR2 : ULPT Mode Register 2

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LPM	—	—	—	—	CKS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CKS[2:0]	ULPTLCLK/ULPTSCLK Count Source Clock Division Ratio*1*2 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128 This bit is valid only in timer mode (ULPTMR1.TMOD1 = 0)	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	LPM	Low Power Mode 0: Normal mode 1: Low power mode	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Do not rewrite this register while the counter is running. Rewrite this register only when the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0).
Note 1. The CKS[2:0] bits can be changed only in timer mode (ULPTMR1.TMOD1 = 0).
Set the CKS[2:0] bits to 000 in event counter mode (ULPTMR1.TMOD1 = 1).
Note 2. The ULPTMR1.TCK1 bit can be changed only when the CKS[2:0] bits are 000. Set the CKS[2:0] bits to 000, and then change the ULPTMR1.TCK1 bit after one cycle of the count source.

The ULPTMR2 register sets the operating mode of the ULPT.

The ULPTMR2 register is only available for 8-bit read/write access.

LPM bit (Low Power Mode)

This bit implements low power consumption by restricting the register access. Set this bit to 1 for low-power mode operation.

When this bit is 1, access to the following registers is prohibited:

- ULPTCNT, ULPTCMA, ULPTCMB, and ULPTCR

After this bit is switched from 1 to 0, the first access to an access-prohibited register is restricted as follows:

- ULPTCNT: The register must be read twice. Only the second reading of data is valid.
- ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR: In write access, leave two cycles of the count source.

23.3.7 ULPTMR3 : ULPT Mode Register 3

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TEEPOL[1:0]		TEECTL[1:0]		—	TOPOL	TEVPOL	TCNTCTL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCNTCTL	Count Function Select*1*2 0: Continuous mode 1: One-shot mode	R/W
1	TEVPOL	ULPTEVIn Polarity Switch 0: External event input (ULPTEVIn pin) 1: External event input (ULPTEVIn pin) in reverse This bit is valid only in event counter mode (ULPTMR1.TMOD1 = 1).	R/W

Bit	Symbol	Function	R/W
2	TOPOL	ULPTOn Polarity Select*3 0: Start the ULPTOn output with low level. 1: Start the ULPTOn output with high level.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	TEECTL[1:0]	ULPTEEn Function Select*1 0 0: Count enable mode 1 0: Count start mode 1 1: Count restart mode Other settings are prohibited. The count enable mode is valid only in event counter mode (ULPTMR1.TMOD1 = 1).	R/W
7:6	TEEPOL[1:0]	ULPTEEn Edge Polarity Select 0 0: Rising edge 0 1: Falling edge 1 0: Both edges Other settings are prohibited. This bit is valid only in count start mode or count restart mode. (TEECTL[1:0] = 10 or 11)	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Do not rewrite this register while the counter is running. Rewrite this register only when the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0).
Note 1. For details, see section 23.4.1. Count operation.
Note 2. Continuous mode: After starting counting, the counter continues to count down (reload value to underflow) repeatedly until the counter stops.
One-shot mode: After starting counting, the counter counts down once (reload value to underflow).
Note 3. When data is set to the ULPTMR1 register or 1 is set to the ULPTCR.TSTOP bit, the ULPTOn output is initialized.

The ULPTMR3 register sets the operating mode of the ULPT.

The ULPTMR3 register is only available for 8-bit read/write access.

TEECTL[1:0] bits (ULPTEEn Function Select)

The ULPTEEn pin functions differently depending on the mode. For details, see section 23.4.1. Count operation.

In count enable mode, external events are counted while the ULPTEEn pin is valid. (The duration can be selected by the ULPTIOC.TIOGT0 bit, and the polarity can be selected by the ULPTISR.RCCPSEL2 bit.)

In count start mode, the counter starts counting at an edge trigger at the ULPTEEn pin. The operation of the ULPTEEn pin differs depending on the TSTART and TCSTF bits of the ULPTCR register.

In count restart mode, the counter starts counting at the first edge trigger at the ULPTEEn pin and resets the count at the second or following edge trigger. The operation of the ULPTEEn pin differs depending on the TSTART and TCSTF bits of the ULPTCR register.

Table 23.3 shows the operation of the ULPTEEn pin in count start or count restart mode.

Table 23.3 Operation of the ULPTEEn pin in count start mode or count restart mode

TEECTL[1:0]	ULPTCR.TSTART	ULPTCR.TCSTF	Function of the ULPTEEn pin and counter operation
10 (count start mode)	0 (stop the counter)	—	The edge input at the ULPTEEn pin is invalid. The counter remains stopped.
	1 (start the counter)	0 (counter stopped)	The counter starts counting from the reload value on the edge input at the ULPTEEn pin.
	1 (start the counter)	1 (counter running)	The edge input at the ULPTEEn pin is invalid. The counter continues to count.
11 (count restart mode)	0 (stop the counter)	—	The edge input at the ULPTEEn pin is invalid. The counter remains stopped.
	1 (start the counter)	0 (counter stopped)	The counter starts counting from the reload value on the edge input at the ULPTEEn pin.
	1 (start the counter)	1 (counter running)	The counter is reset on the edge input at the ULPTEEn pin, and it starts counting again from the reload value.

Note: The edge polarity of the ULPTEEn pin can be selected by the TEEPOL[1:0] bits.

23.3.8 ULPTIOC : ULPT I/O Control Register

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	TIOGT 0	TIPF[1:0]		—	TOE	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	TOE	ULPTOn Output Enable 0: Disable the ULPTOn output. 1: Enable the ULPTOn output.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	TIPF[1:0]	ULPTEVIn Input Filter*2 0 0: No filter 0 1: Filter sampling at PCLKB 1 0: Filter sampling at PCLKB/8 1 1: Filter sampling at PCLKB/32 This bit is valid only in event counter mode (ULPTMR1.TMOD1 = 1).	R/W
6	TIOGT0	ULPTEVIn Count Control*1 0: Always count external events. 1: Count external events while the ULPTEVIn pin is valid. This bit is valid only in event counter mode (ULPTMR1.TMOD1 = 1) and count enable mode (ULPTMR3.TEECTL[1:0] = 00).	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Do not rewrite this register while the counter is running. Rewrite this register only when the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0).
Note 1. The polarity for counting external events can be selected by the ULPTISR.RCCPSEL2 bit.
Note 2. Set the TIPF[1:0] bits to 00 before entering Software Standby mode or Deep Software Standby mode1.

The ULPTIOC register controls the input/output of the ULPT.

The ULPTIOC register is only available for 8-bit read/write access.

TIPF[1:0] bits (ULPTEVIn Input Filter)

These bits are used to specify the sampling frequency of the ULPTEVIn input filter. Input to the ULPTEVIn pin is sampled. When a value matches three consecutive times, the value is regarded as the input value.

23.3.9 ULPTISR : ULPT Event Pin Select Register

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x11

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	RCCP SEL2	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
2	RCCPSEL2	ULPTEEn Polarity Select 0: Count external events when low level. 1: Count external events when high level. This bit is valid only in event counter mode (ULPTMR1.TMOD1 = 1), count enable mode (ULPTMR3.TEECTL[1:0] = 00), and when ULPTIOC.TIOGT0 = 1.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Do not rewrite this register while the counter is running. Rewrite this register only when the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0).

The ULPTISR register sets the polarity of the ULPTEEn pin.

The ULPTISR register is only available for 8-bit read/write access.

23.3.10 ULPTCMSR : ULPT Compare Match Function Select Register

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x12

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	TOPOLB	TOEB	TCMEB	—	TOPOLA	TOEA	TCMEA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCMEA	Compare Match A Register Enable*1 0: Disable compare match A register. 1: Enable compare match A register.	R/W
1	TOEA	ULPTOAn Output Enable 0: Disable the ULPTOAn output. 1: Enable the ULPTOAn output.	R/W
2	TOPOLA	ULPTOAn Polarity Select*2 0: Start the ULPTOAn output with low. 1: Start the ULPTOAn output with high.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	TCMEB	Compare Match B Register Enable*1 0: Disable compare match B register. 1: Enable compare match B register.	R/W
5	TOEB	ULPTOBn Output Enable 0: Disable the ULPTOBn output. 1: Enable the ULPTOBn output.	R/W
6	TOPOLB	ULPTOBn Polarity Select*2 0: Start the ULPTOBn output with low. 1: Start the ULPTOBn output with high.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Do not rewrite this register while the counter is running. Rewrite this register only when the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0).
Note 1. When 1 (stop the counter forcibly) is set to the ULPTCR.TSTOP bit, the TCMEA, and TCMEB bits are set to 0.
Note 2. When data is set to the ULPTMR1 register or 1 is set to the ULPTCR.TSTOP bit, the ULPTOAn and ULPTOBn output is initialized.

The ULPTCMSR register controls the interrupt request signal and input/output of compare match A and compare match B.

The ULPTCMSR register is only available for 8-bit read/write access.

23.4 Operation

23.4.1 Count operation

The counter is decremented by one on each edge of the count source. When the count value reaches 0x00000000 and the next edge of the count source is input, an interrupt request of underflow is generated. The count operation is set by the ULPTMR1 register and the ULPTMR3 register. Table 23.4 lists the count operations.

Table 23.4 Count operation

Section No.	Count operation			Count source	ULPTMR1 register		ULPTMR3 register	
	Mode 1	Mode 2	Mode 3		TMOD1	TCK1	TCNTCTL	TEECTL[1:0]
section 23.4.1.1. Timer & continuous mode	Timer	Continuous	—	ULPTLCLK/ ULPTSCLK	0	0 : ULPTLCLK 1 : ULPTSCLK	0	00
section 23.4.1.5. Common & continuous & count start mode			Count start					10
section 23.4.1.6. Common & continuous & count restart mode			Count restart					11
section 23.4.1.2. Timer & one-shot mode		One-shot	—				1	00
section 23.4.1.7. Common & one-shot & count start mode			Count start					10
section 23.4.1.8. Common & one-shot & count restart mode			Count restart					11
section 23.4.1.3. Event counter & continuous & count enable mode	Event counter	Continuous	Count enable	ULPTEVIn pin	1	Don't care	0	00
section 23.4.1.5. Common & continuous & count start mode			Count start					10
section 23.4.1.6. Common & continuous & count restart mode			Count restart					11
section 23.4.1.4. Event counter & one-shot & count enable mode		One-shot	Count enable				1	00
section 23.4.1.7. Common & one-shot & count start mode			Count start					10
section 23.4.1.8. Common & one-shot & count restart mode			Count restart					11

Note: For details about register settings, see section 23.3.5. ULPTMR1 : ULPT Mode Register 1 and section 23.3.7. ULPTMR3 : ULPT Mode Register 3 .

[Mode 1]

The clock used as the count source is selected by the TMOD1 and TCK1 bits of the ULPTMR1 register.

Timer mode: The internal clock (ULPTLCLK or ULPTSCLK) is used as the count source.

The counter is decremented on each rising edge of the count source.

Event counter mode: The external event input (ULPTEVIn pin) is used as the count source.

The counter is decremented on each edge of the count source.

The edge polarity can be selected by the ULPTMR1.TEDGPL bit and the ULPTMR3.TEVPOL bit.

[Mode 2]

The counter behavior is selected by the TCNTCTL bit of the ULPTMR3 register.

Continuous mode: After starting counting, the counter continues to count down (reload value to underflow) repeatedly until the counter stops.

One-shot mode: After starting counting, the counter counts down once (reload value to underflow).

[Mode 3]

The function of the ULPTEEn pin is selected by the TEECTL[1:0] bits of the ULPTMR3 register.

Count enable mode: External events are counted while the ULPTEEn pin is valid.

(This bit is valid only in event counter mode.)

Note: Because the ULPTEEn pin is synchronized with the count source, external events are counted after three cycles of the count source.

Count start mode: The counter starts counting at an edge trigger of the ULPTEEn pin.

The edge polarity of the ULPTEEn pin can be selected by the ULPTMR3.TEEPOL[1:0] bits.

Note: Because the ULPTEEn pin is synchronized and held with the count source, external events are counted after four cycles of the count source.

Count restart mode: The counter starts counting at the first edge trigger of the ULPTEEn pin.

The counter resets the count at the second or following edge trigger.

The edge polarity of the ULPTEEn pin can be selected by the ULPTMR3.TEEPOL[1:0] bits.

Note: Because the ULPTEEn pin is synchronized and held with the count source, external events are counted after four cycles of the count source.

23.4.1.1 Timer & continuous mode

This mode operates in timer mode only.

The counter continues to count down (reload value to underflow) repeatedly in synchronization with the count source.

When 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1).

When an underflow occurs, the counter counts down again from the reload value.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 23.2 shows an operation example.

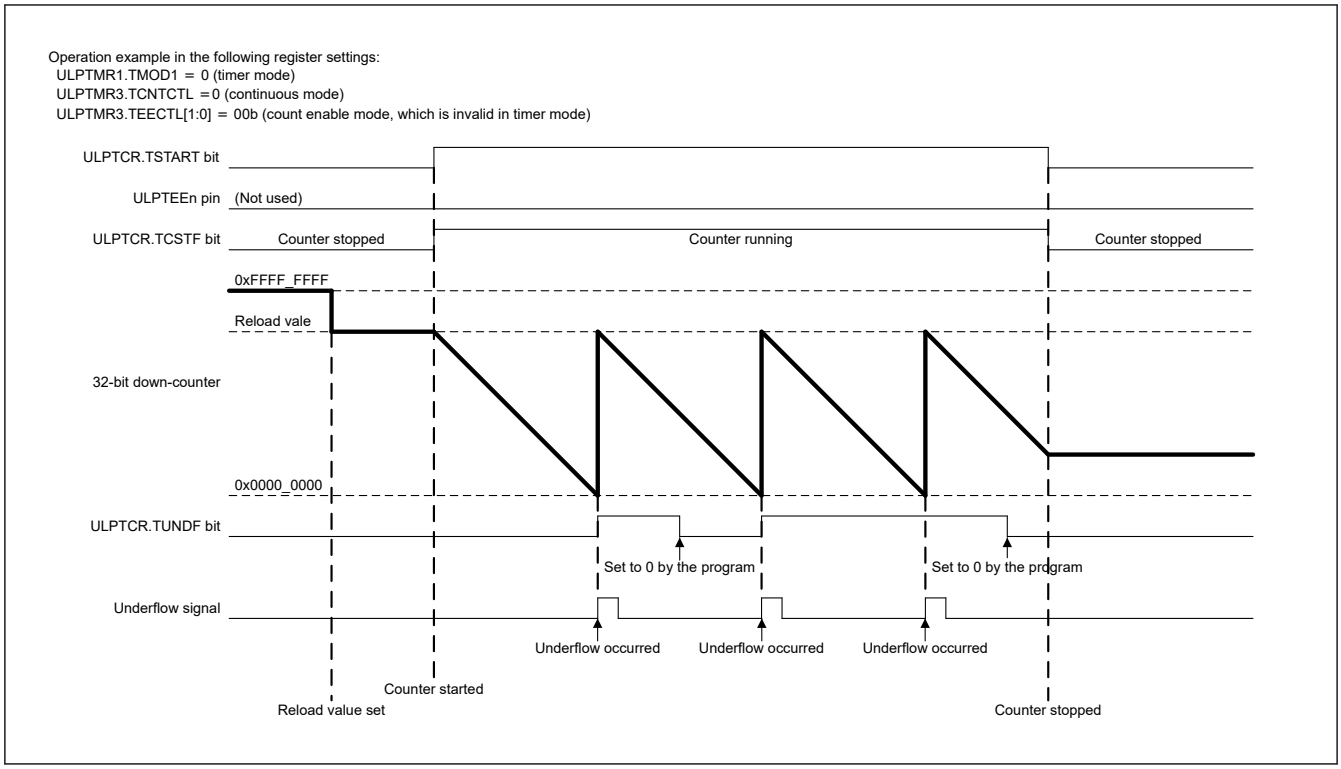


Figure 23.2 Operation example in timer & continuous mode

23.4.1.2 Timer & one-shot mode

This mode operates in timer mode only.

The counter counts down (reload value to underflow) once in synchronization with the count source.

When 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1).

When an underflow occurs, the counter is stopped (with the ULPTCR.TSTART bit set to 1 and the ULPTCR.TCSTF bit set to 0).

To restart the counter after the counter is stopped, perform the following procedure:

1. Read the ULPTCR.TUNDF bit to confirm that it is set to 1 (underflow occurred).
   This is to make sure that counting down is finished.
2. Write 0 to the ULPTCR.TSTART bit to stop the counter.
3. Write 1 to the ULPTCR.TSTART bit to start the counter.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 23.3 shows an operation example.

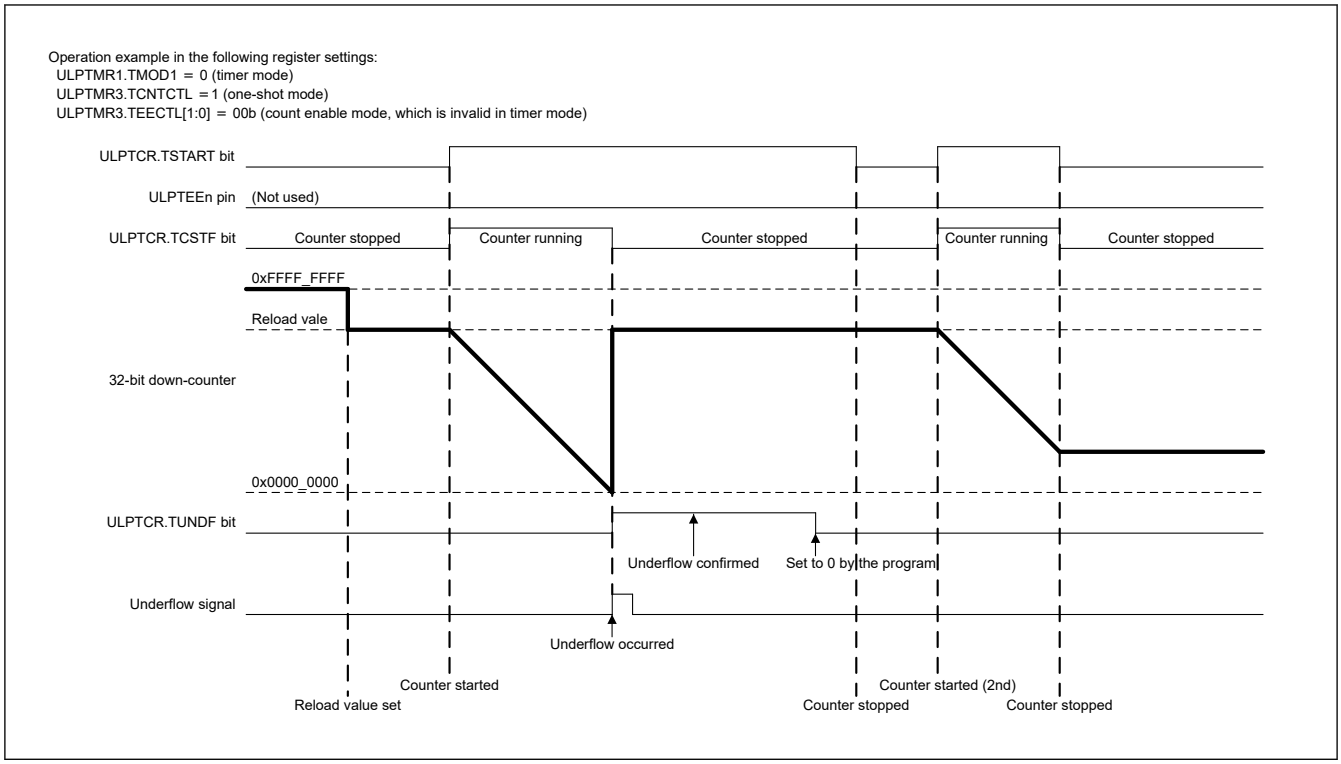


Figure 23.3 Operation example in timer & one-shot mode

23.4.1.3 Event counter & continuous & count enable mode

This mode operates in event counter mode only.

The counter continues to count down (reload value to underflow) repeatedly in synchronization with the count source. The duration in which the count source is counted can be specified by the ULPTIOC.TIOGT0 bit and ULPTISR.RCCPSEL2 bit.

When 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1).

When an underflow occurs, the counter counts down again from the reload value.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 23.4 shows an operation example when the ULPTIOC.TIOGT0 bit is set to 0, and Figure 23.5 and Figure 23.6 show operation examples when the ULPTIOC.TIOGT0 bit is set to 1.

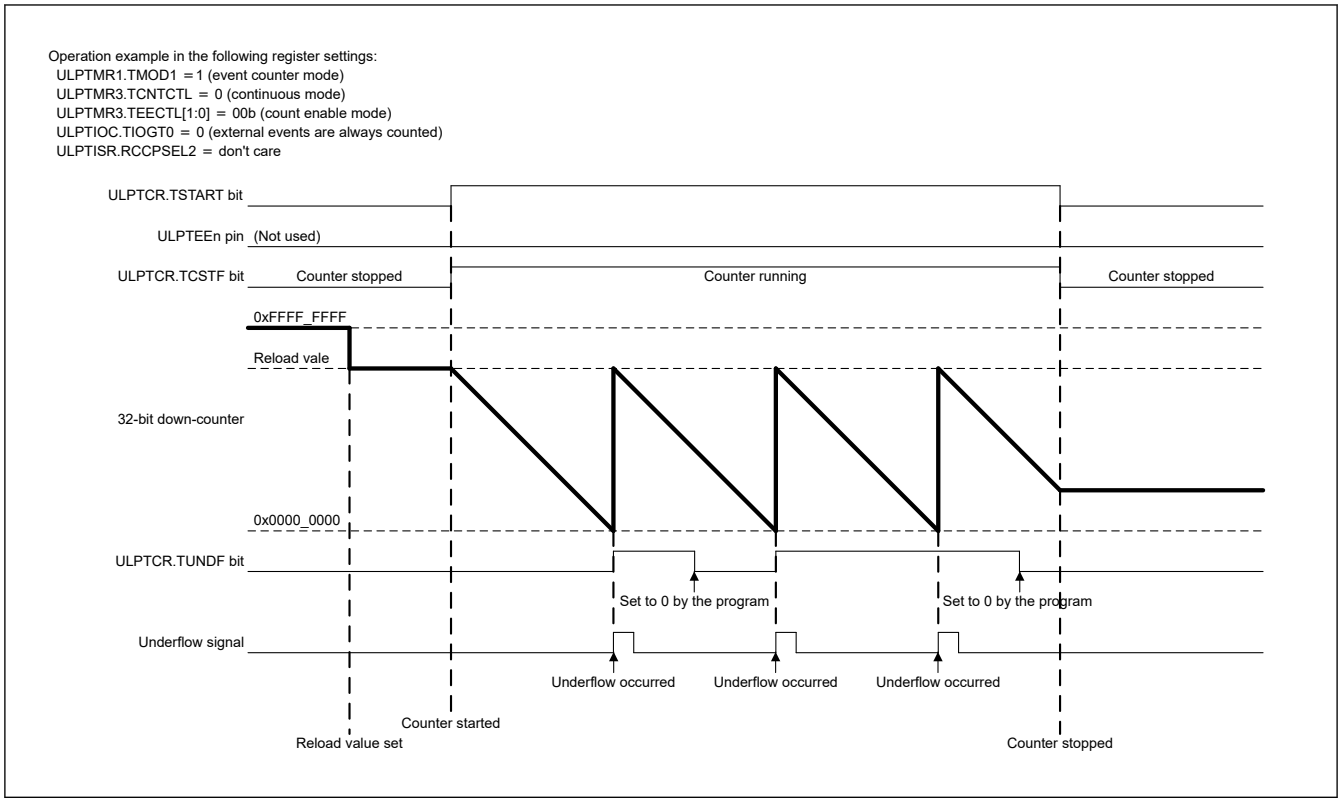


Figure 23.4 Operation example in event counter & continuous & count enable mode (with the ULPTIOC.TIOGT0 bit set to 0)

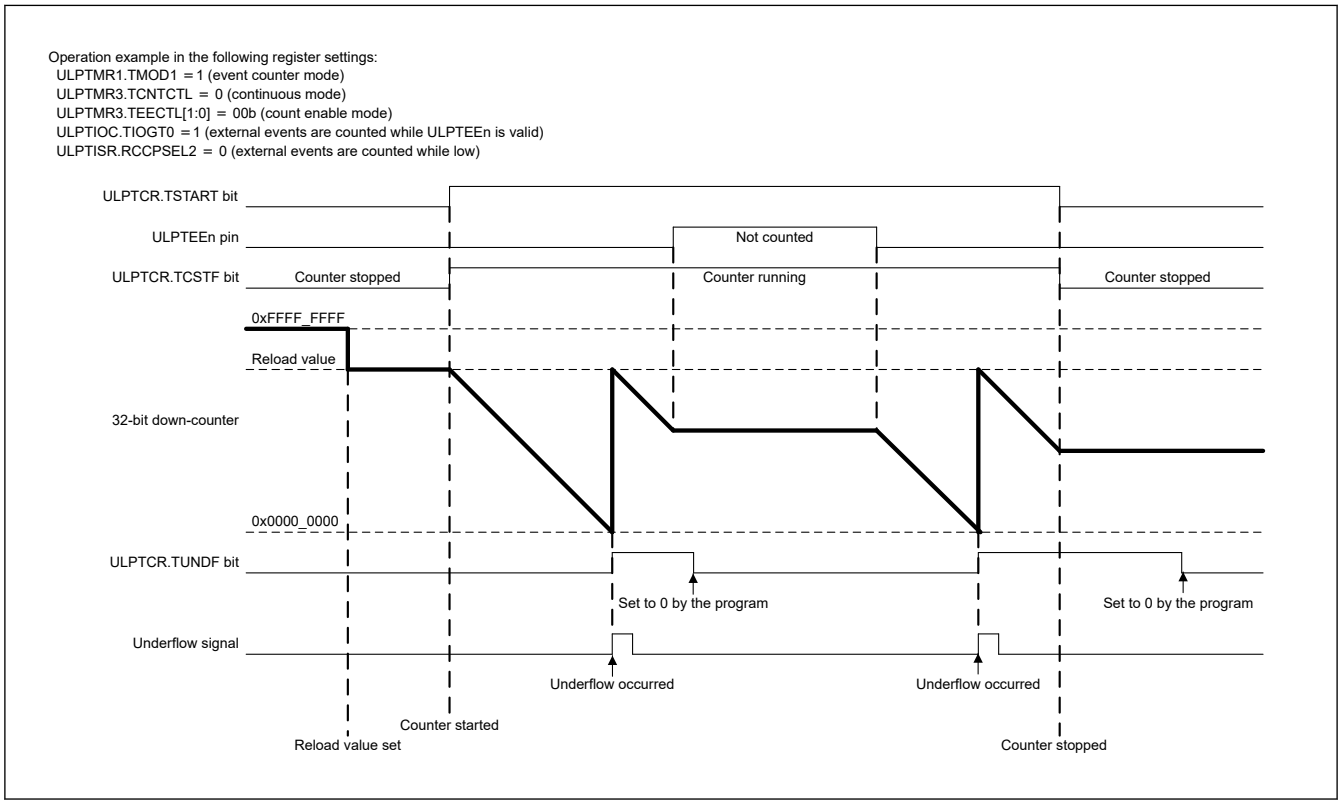
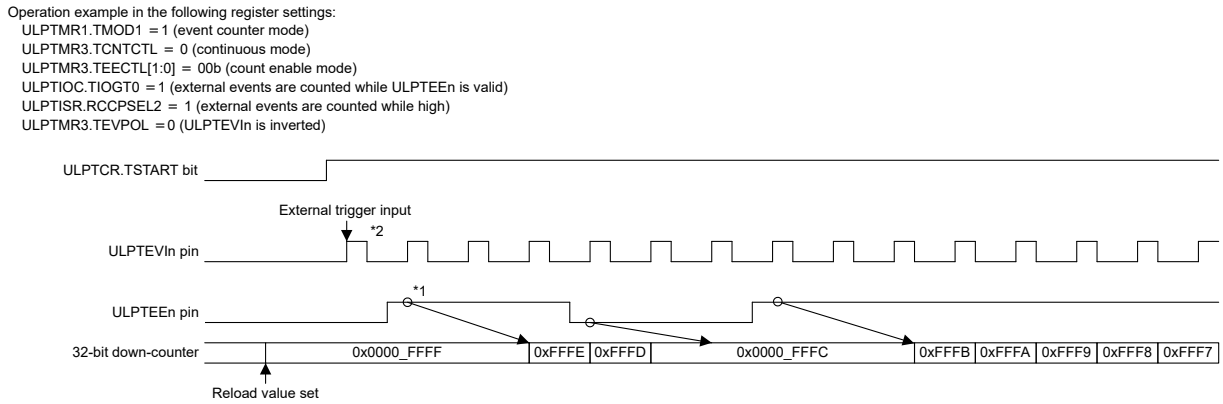


Figure 23.5 Operation example 1 in event counter & continuous & count enable mode (with the ULPTIOC.TIOGT0 bit set to 1)



Note 1. When synchronization is controlled, there is a delay by two cycles of count source until external events are actually counted. In addition, the count start timing may be shifted by one cycle due to the phase difference between the ULPTEEn pin and the sampling clock.

Note 2. Depending on the state before the counter is stopped, counting may be performed during two cycles of the count source immediately after the counter is started.
To disable the counting operation during two cycles immediately after the counter is started, set 1 to the ULPTCR.TSTOP bit to initialize the internal circuits, and then finish the operation settings before starting the counter.

Figure 23.6 Operation example 2 in event counter & continuous & count enable mode (with the ULPTIOC.TIOGT0 bit set to 1)

23.4.1.4 Event counter & one-shot & count enable mode

This mode operates in event counter mode only.

The counter counts down (reload value to underflow) once in synchronization with the count source. The duration in which the count source is counted can be specified by the ULPTIOC.TIOGT0 bit and ULPTISR.RCCPSEL2 bit.

When 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1).

When an underflow occurs, the counter is stopped (with the ULPTCR.TSTART bit set to 1 and the ULPTCR.TCSTF bit set to 0).

To restart the counter after the counter is stopped, perform the following procedure:

1. Read the ULPTCR.TUNDF bit to confirm that it is set to 1 (underflow occurred).
   This is to make sure that counting down is finished.
2. Write 0 to the ULPTCR.TSTART bit to stop the counter.
3. Write 1 to the ULPTCR.TSTART bit to start the counter.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 23.7 shows an operation example when the ULPTIOC.TIOGT0 bit is set to 00, and Figure 23.8 shows an operation example when the ULPTIOC.TIOGT0 bit is set to 01.

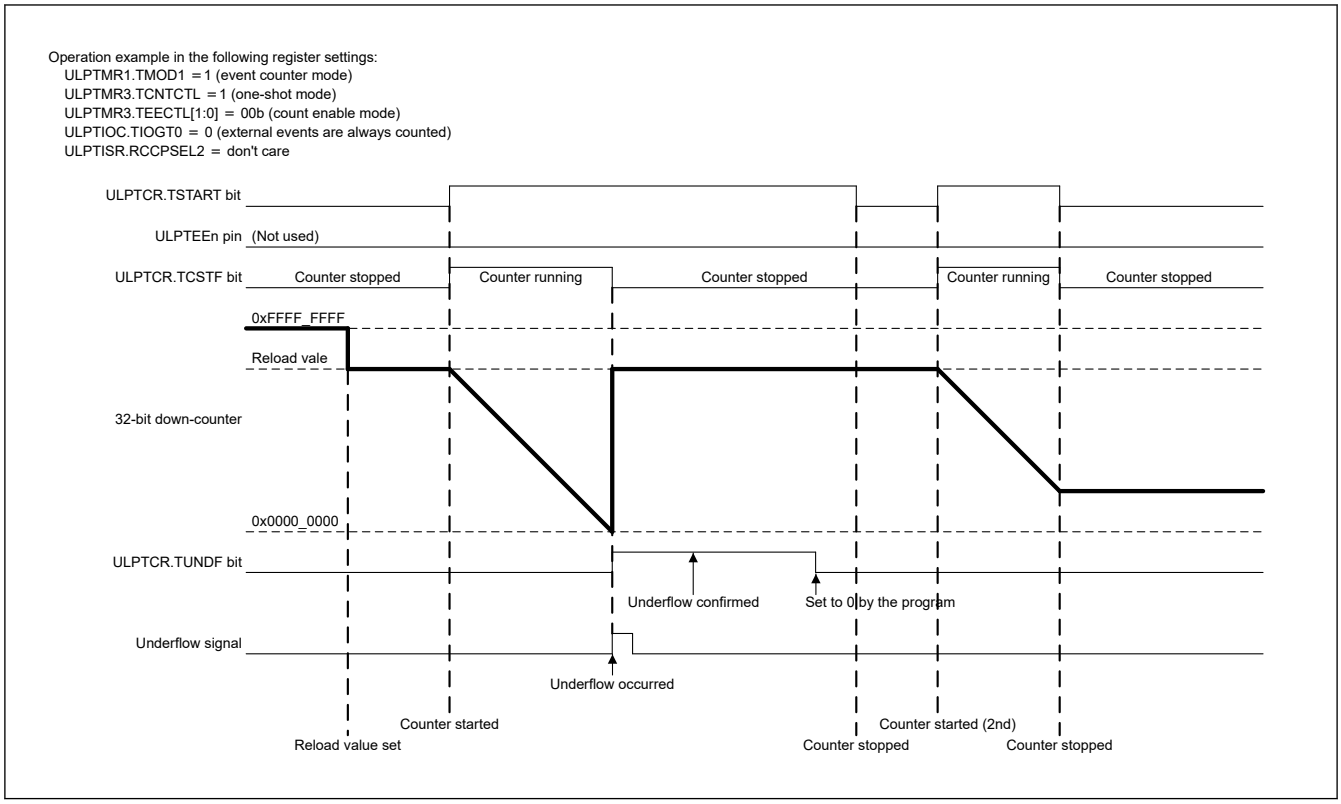


Figure 23.7 Operation example in event counter & one-shot & count enable mode (with the ULPTIOC.TIOGT0 bit set to 0)

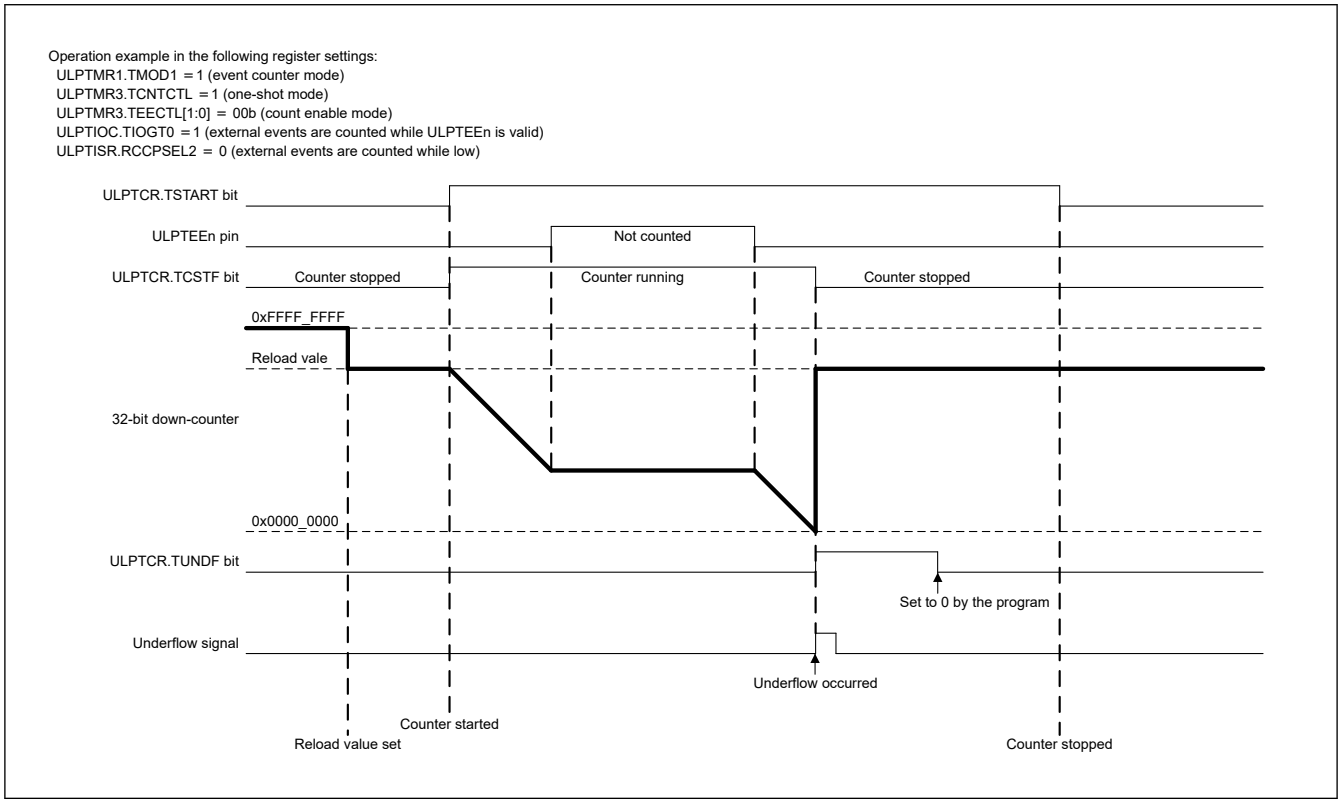


Figure 23.8 Operation example in event counter & one-shot & count enable mode (with the ULPTIOC.TIOGT0 bit set to 1)

23.4.1.5 Common & continuous & count start mode

This mode operates in both timer mode and event counter mode.

The counter continues to count down (reload value to underflow) repeatedly in synchronization with the count source.

After 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1) by an edge trigger at the ULPTEEn pin.

When an underflow occurs, the counter counts down again from the reload value.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 23.9 shows an operation example.

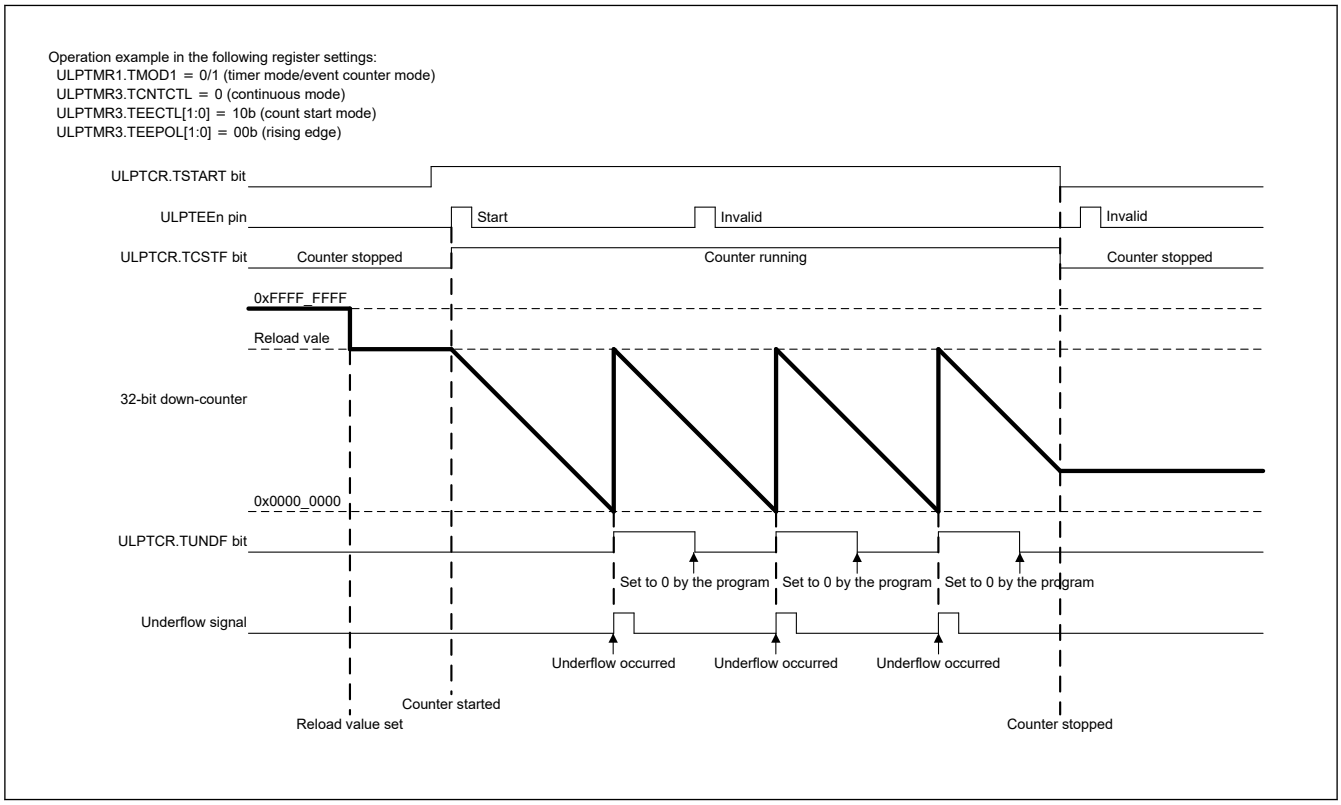


Figure 23.9 Operation example in common & continuous & count start mode

23.4.1.6 Common & continuous & count restart mode

This mode operates in both timer mode and event counter mode.

The counter continues to count down (reload value to underflow) repeatedly in synchronization with the count source.

After 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1) by an edge trigger at the ULPTEEn pin.

On an edge trigger input at the ULPTEEn pin when the counter is running (with ULPTCR.TCSTF bit set to 1), the counter is reset and it starts counting again from the reload value.

If no edge trigger is input from the ULPTEEn pin and an overflow occurs, the counter starts counting again from the reload value.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 23.10 shows an operation example.

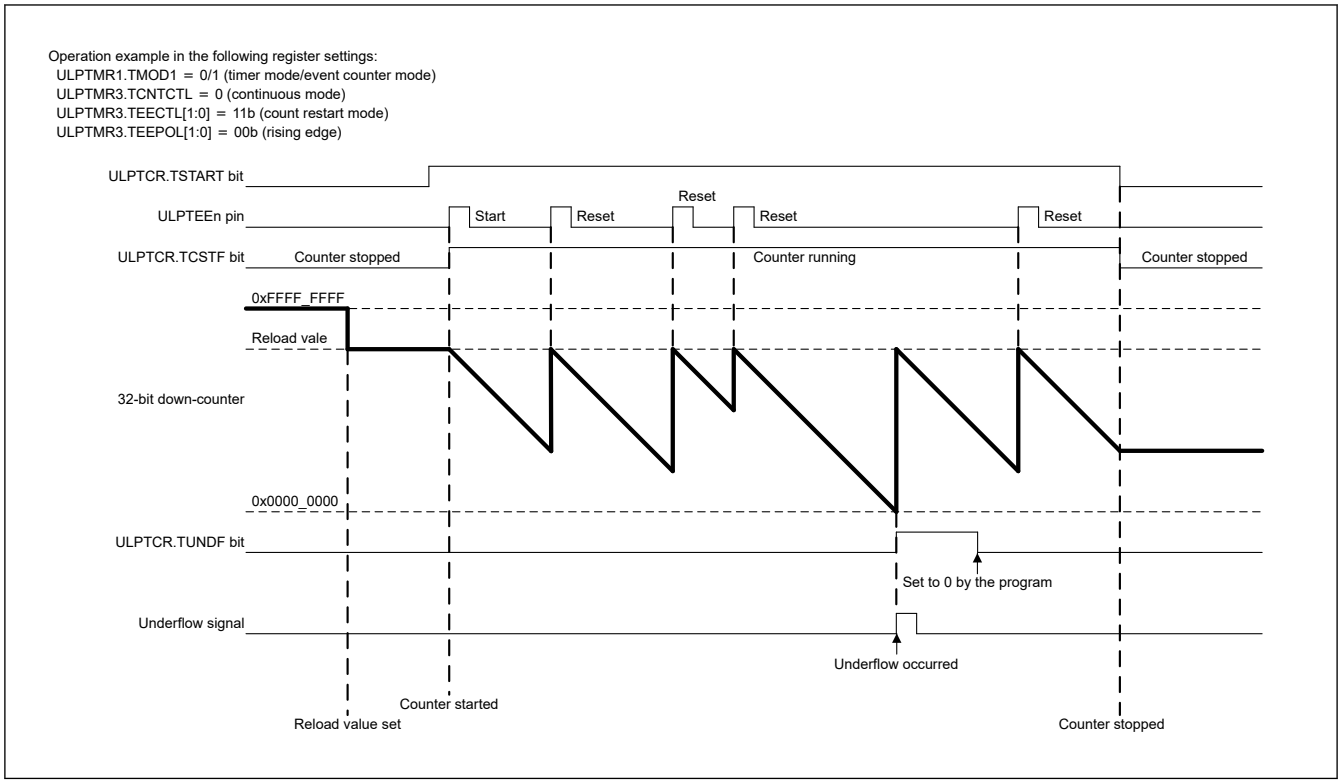


Figure 23.10 Operation example in common & continuous & count restart mode

23.4.1.7 Common & one-shot & count start mode

This mode operates in both timer mode and event counter mode.

The counter counts down (reload value to underflow) once in synchronization with the count source.

After 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1) by an edge trigger at the ULPTEEn pin.

When an underflow occurs, the counter is stopped (with the ULPTCR.TSTART bit set to 1 and the ULPTCR.TCSTF bit set to 0).

On an edge trigger input at the ULPTEEn pin when the counter is running (with ULPTCR.TSTART bit set to 1 and the ULPTCR.TCSTF bit set to 0), the counter starts counting down again from the reload value.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 23.11 shows an operation example.

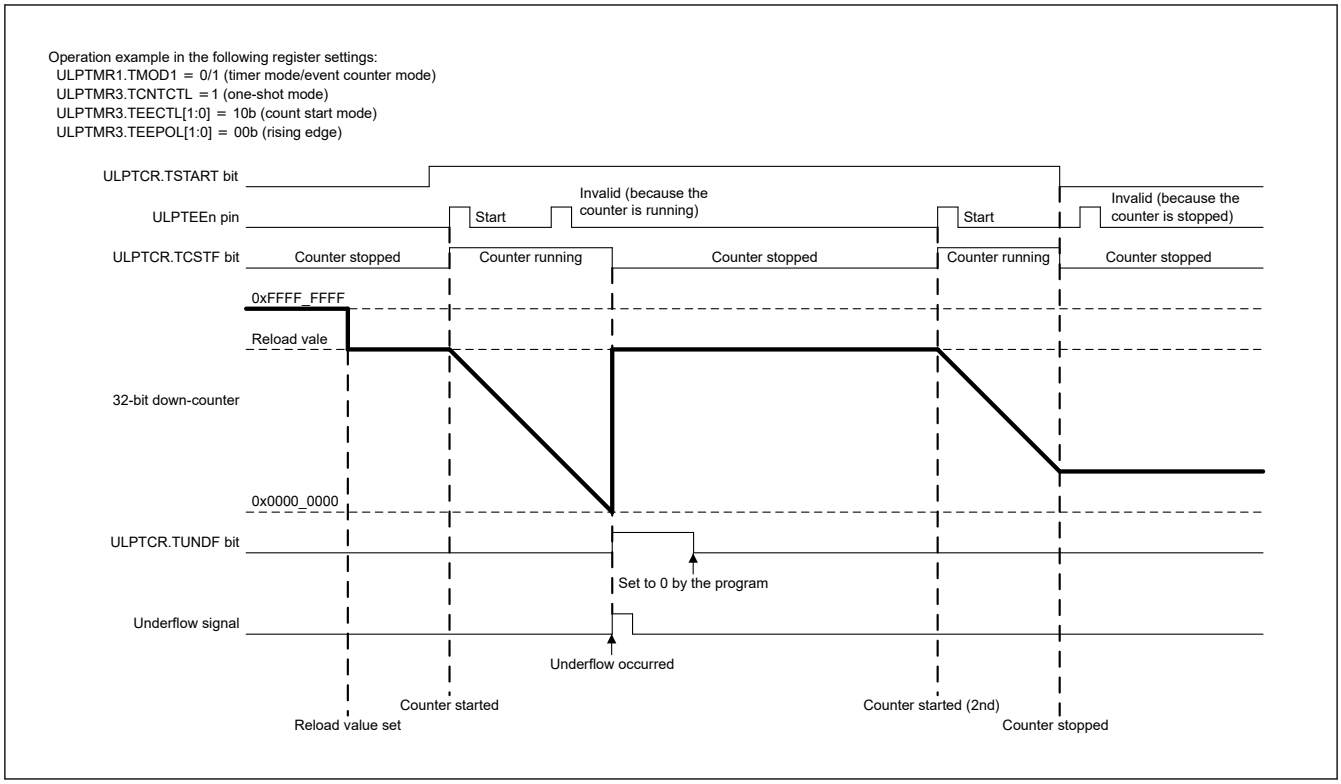


Figure 23.11 Operation example in common & one-shot & count start mode

23.4.1.8 Common & one-shot & count restart mode

This mode operates in both timer mode and event counter mode.

The counter counts down (reload value to underflow) once in synchronization with the count source.

After 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1) by an edge trigger of the ULPTEEn pin.

On an edge trigger input at the ULPTEEn pin when the counter is running (with ULPTCR.TCSTF bit set to 1), the counter is reset and it starts counting again from the reload value.

If no edge trigger is input at the ULPTEEn pin and an underflow occurs, the counter is stopped (with the ULPTCR.TSTART bit set to 1 and the ULPTCR.TCSTF bit set to 0).

On an edge trigger input at the ULPTEEn pin when the counter is running (with ULPTCR.TSTART bit set to 1 and the ULPTCR.TCSTF bit set to 0), the counter starts counting down again from the reload value.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 23.12 shows an operation example.

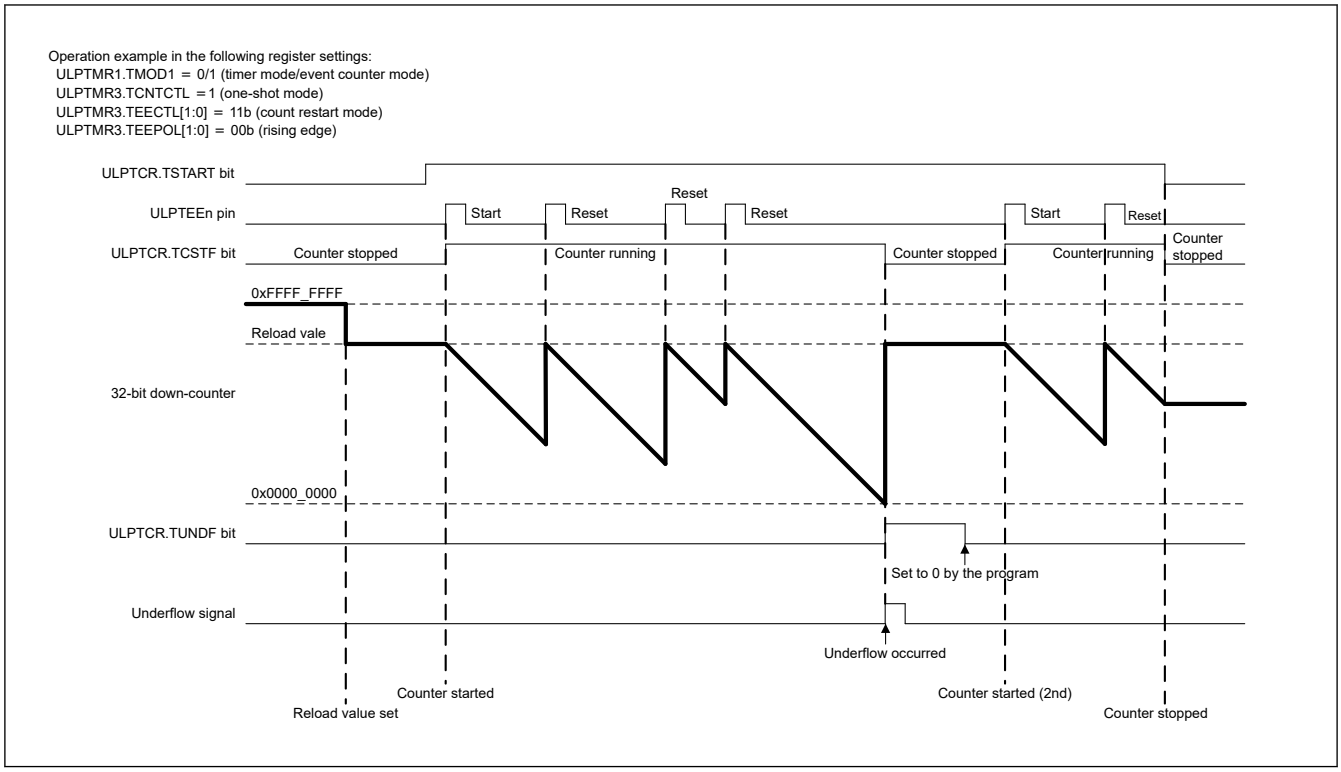


Figure 23.12 Operation example in common & one-shot & count restart mode

23.4.2 Rewriting the counter and reload register

Regardless of the operating mode, the counter and reload register rewrite timing changes depending on the TSTART and TCSTF bits of the ULPTCR register, the TCMEA and TCMEB bits of the ULPTCMSR register, and the TEECTL[1:0] bits of the ULPTMR3 register.

Condition A: When the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0)

A reload value is written directly to the counter and the reload register.

Condition B: When the counter is running (with the ULPTCR.TCSTF bit set to 1) and the compare match A/B registers are disabled (with both the TCMEA and TCMEB bits of the ULPTCR register set to 0)

After a reload value is written to the ULPTCNT register, the reload value is set first to the reload register, then to the counter in synchronization with the count source.

Condition C: When the counter is running (with the ULPTCR.TCSTF bit set to 1) and the compare match A/B registers are enabled (with either the TCMEA bit or the TCMEB bit of the ULPTCR register set to 1)

After a reload value is written to the ULPTCNT register, the reload value is written to the reload register in synchronization with the count source. Then, the reload value is set to the counter in synchronization with an underflow.

Condition D: When the counter is running (with ULPTCR.TCSTF bit set to 1) and when the count restart mode is enabled (with the ULPTMR3.TEECTL[1:0] bits set to 11)

After an edge trigger is an input from the ULPTEEn pin, the reload value set in the reload register is set to the counter in synchronization with the count source. In this condition, the reload register is not rewritten.

Figure 23.13 to Figure 23.15 show the counter and reload register rewrite timing.

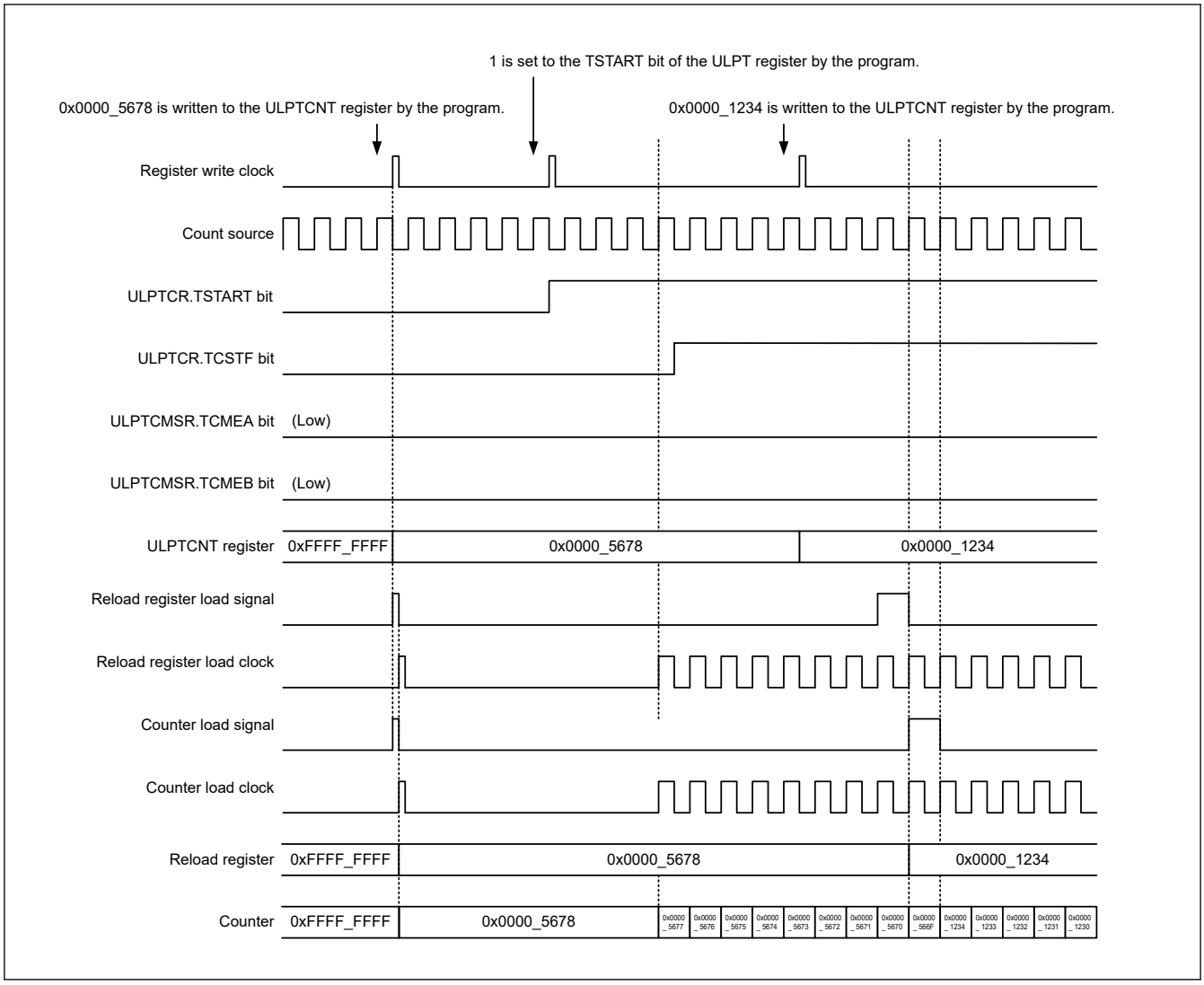


Figure 23.13 Counter and reload rewrite timing (under condition A or B)

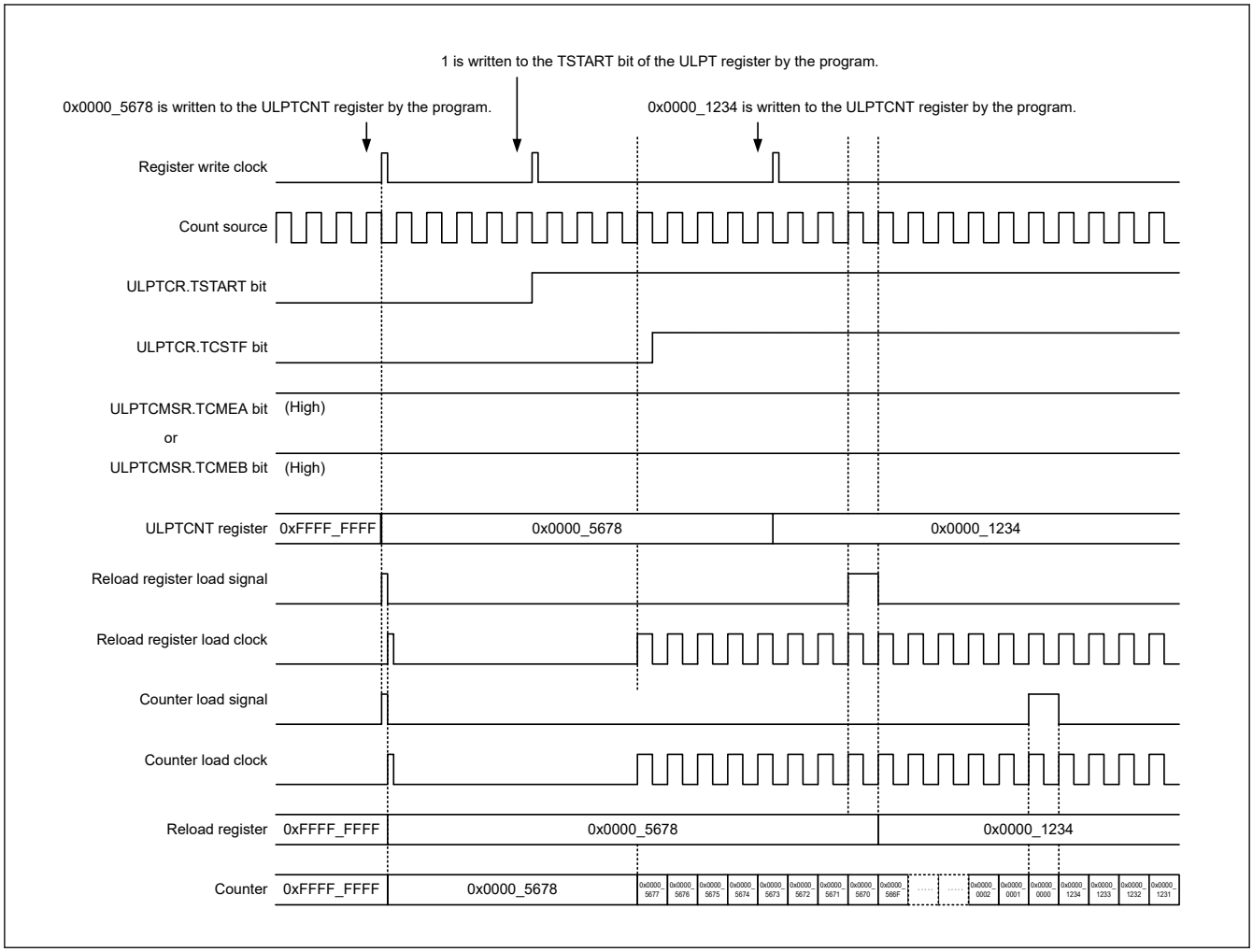


Figure 23.14 Counter and reload rewrite timing (under condition A or C)

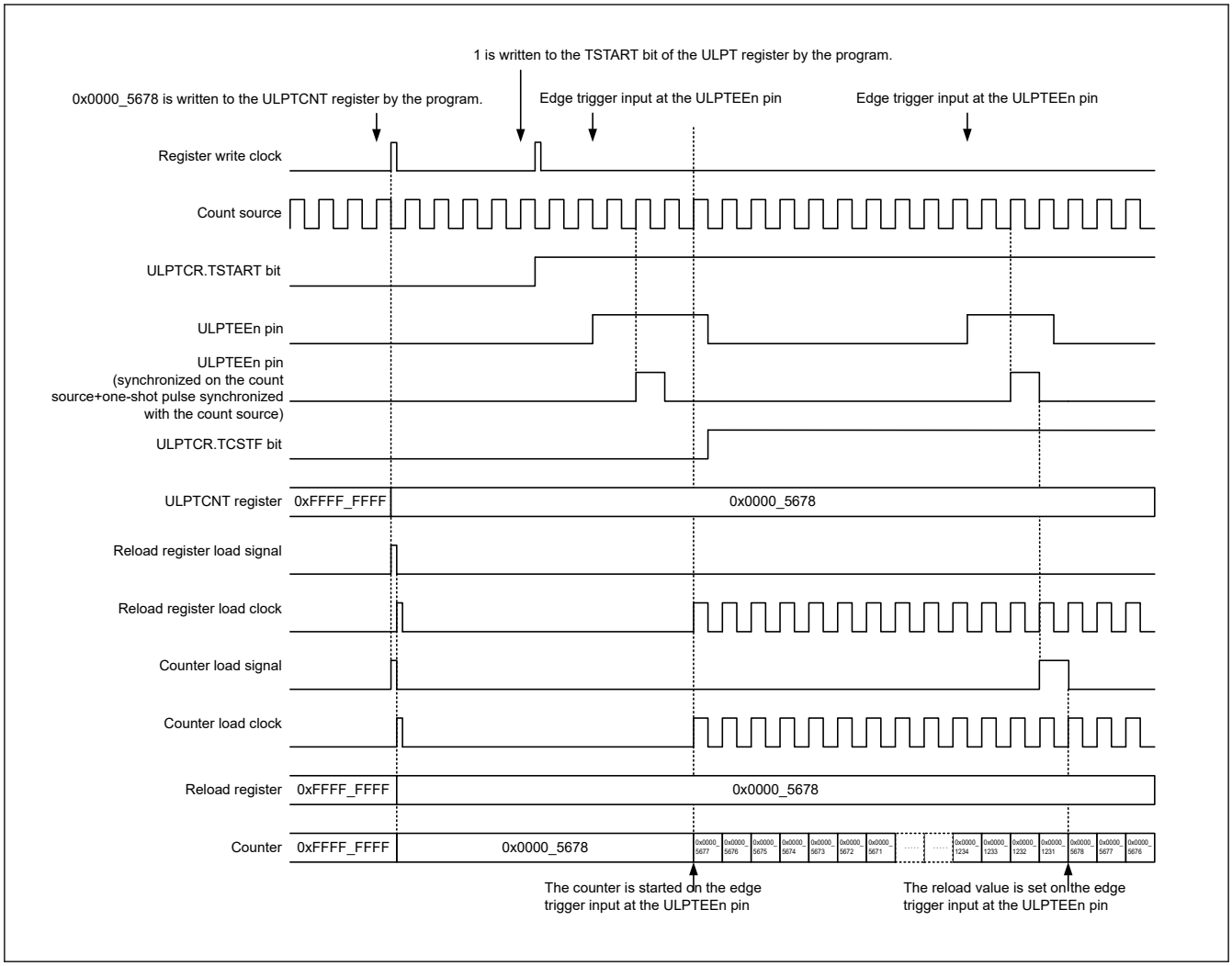


Figure 23.15 Counter and reload rewrite timing (under condition D)

23.4.3 Rewriting the compare circuits and reload registers for compare match A/B

Regardless of the operating mode, the rewrite timing for the compare circuits and reload registers for compare match A/B changes depending on the TSTART and TCSTF bits of the ULPTCR register.

Condition A: When the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0)

Compare match A/B values are written directly to the compare circuits and reload registers.

Condition B: When the counter is running (with the ULPTCR.TCSTF bit set to 1)

After compare match A/B values are written to the ULPTCMA and ULPTCMB registers, the compare match A/B values are written to the reload registers in synchronization with the count source. Then, the compare match A/B values are written to the compare circuits in synchronization with an underflow.

Figure 23.16 shows the rewrite timing for the compare circuit and reload register for compare match A.

The timing for compare match B is similar to that for compare match A.

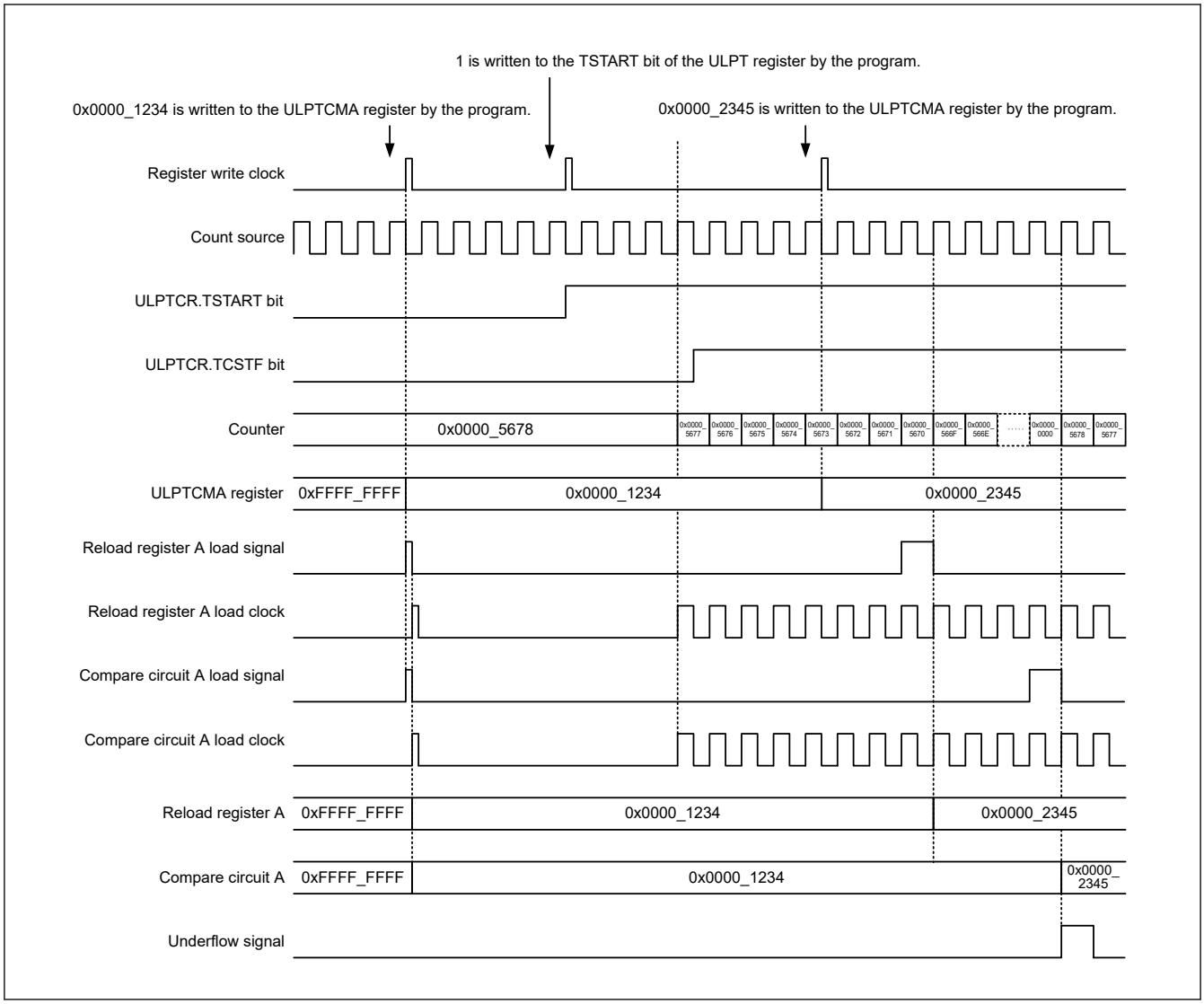


Figure 23.16 Rewrite timing for the compare circuit and reload register for compare match A

23.4.4 Pulse output

Regardless of the operating mode, the ULPTOn pin can output pulses. The output level is inverted each time the counter underflows. The pulse output from the ULPTOn pin can be stopped by the ULPTIOC.TOE bit. The initial value of the output level can be selected by the ULPTMR3.TOPOL bit.

Figure 23.17 shows an operation example of pulse output.

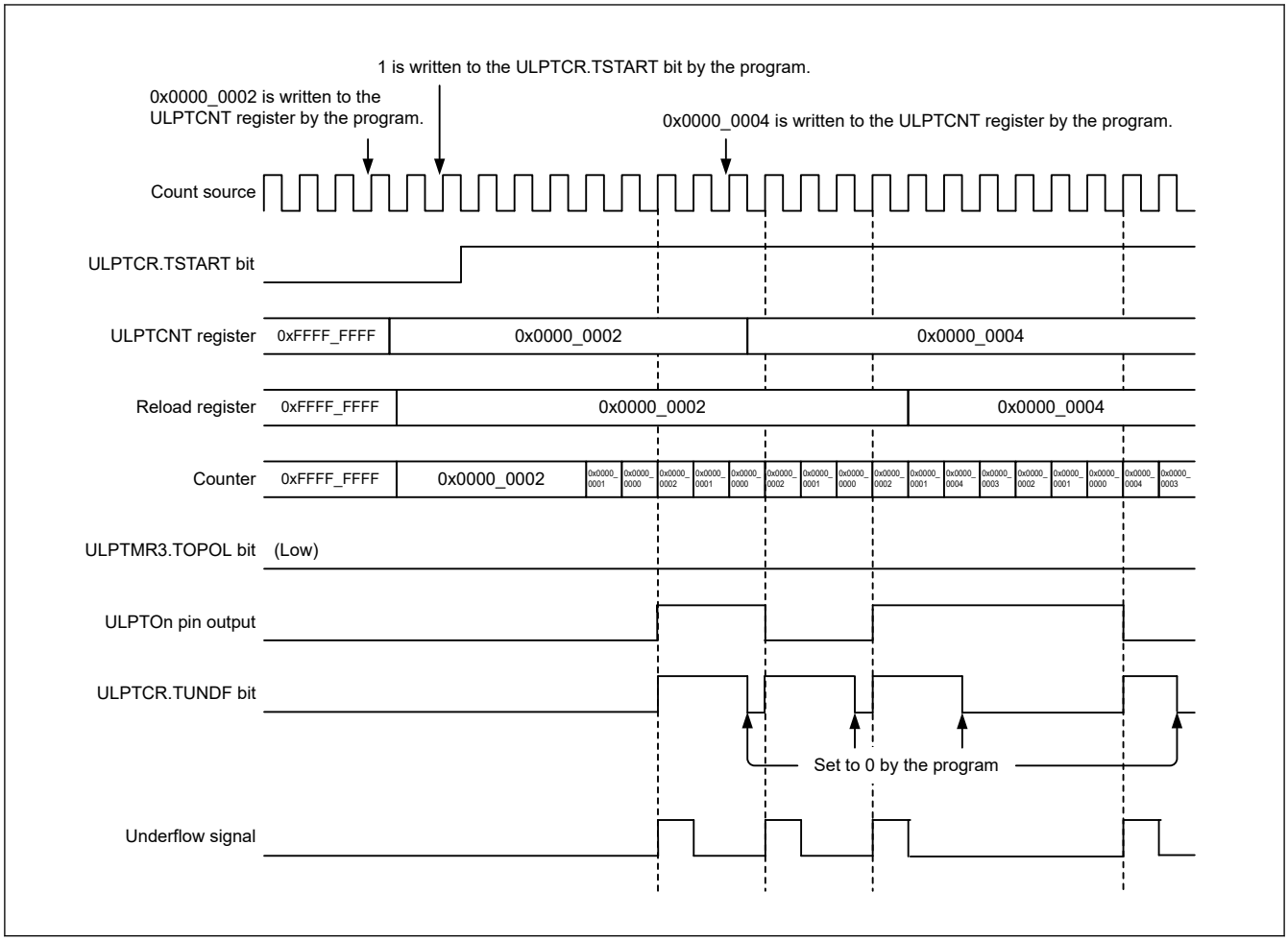


Figure 23.17 Operation example of pulse output

23.4.5 Compare match function

This function detects a match (compare match) between the ULPTCMA/ULPTCMB register contents and the ULPTCNT register contents. This function is enabled when the TCMEA or TCMEB bit of the ULPTCMSR register is set to 1 (the compare match A/B registers are enabled).

The counter is decremented in synchronization with the count source selected by the TMOD1/TCK1 bits of the ULPTMR1 register. When the ULPTCNT register value matches the ULPTCMA or ULPTCMB register value, the TCMAF or TCMBF bit of the ULPTCR register is set to 1 and an interrupt request is generated.

The counter and reload register rewrite timing differs depending on whether the compare match function is enabled. For details, see section 23.4.2. Rewriting the counter and reload register.

The output level of the ULPTOAn/ULPTOBn pin is inverted upon a match or underflow. The output level can be selected by the ULPTOAn/ULPTOBn bit of the ULPTCMSR register.

Figure 23.18 shows an operation example of the compare match function.

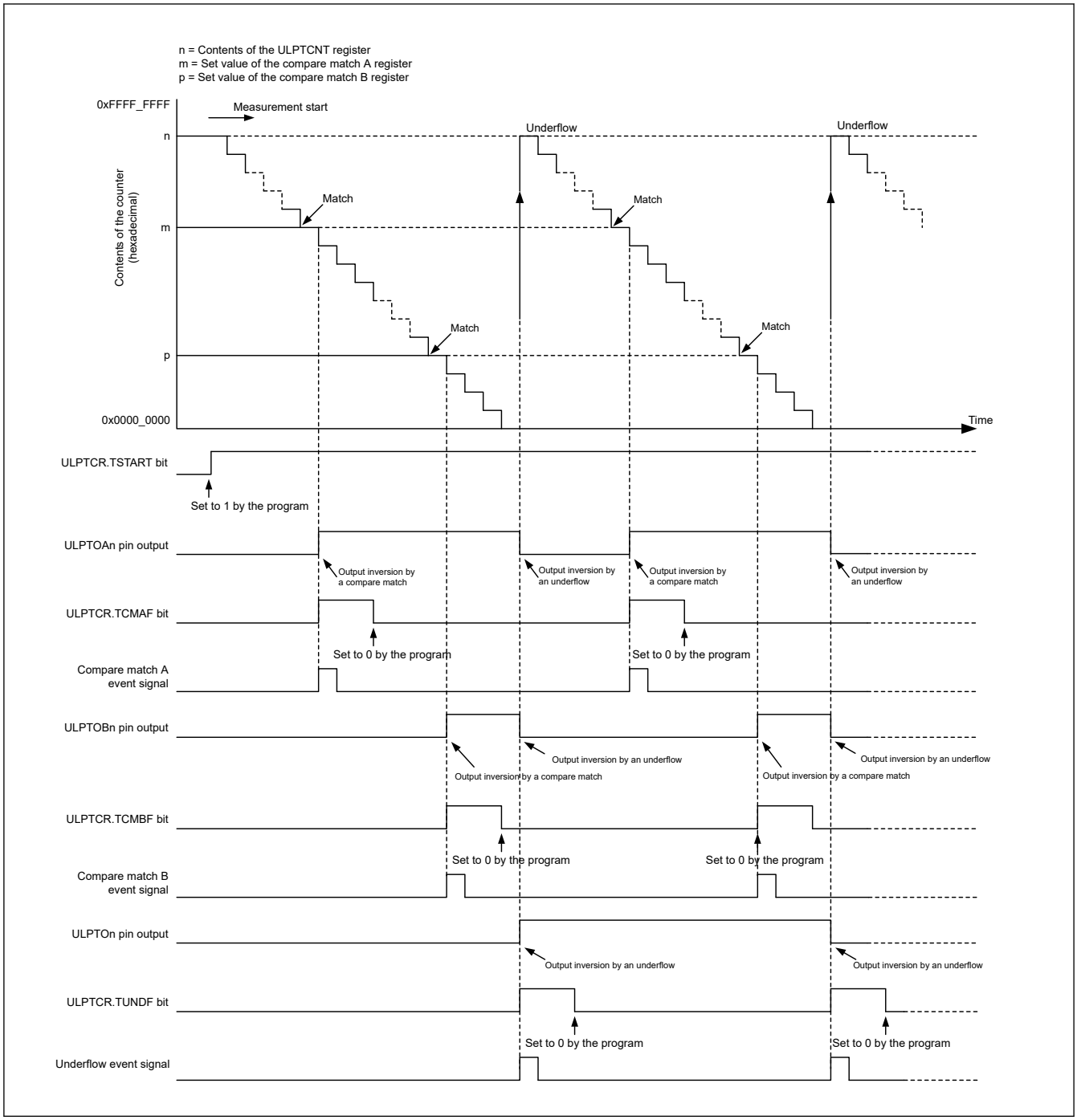


Figure 23.18 Operation example of the compare match function (both the TOPOLA and TOPOLB bits of the ULPTCMSR register are set to 0)

23.4.6 Input and output settings for each mode

Table 23.5 to Table 23.9 show the states of the input pins (ULPTEEn and ULPTEVIn) and output pins (ULPTOn, ULPTOAn, and ULPTOBn) in each mode.

Table 23.5 ULPTEEn pin settings

Operating mode		ULPTMR1 register TMOD1 bit	ULPTMR3 register TEECTL[1:0] bit	ULPTEEn pin input
Timer mode	-	0	00	Input disabled
	Count start		10	Normal input
	Count restart		11	
Event counter mode	Count enable	1	00	
	Count start		10	
	Count restart		11	

Table 23.6 ULPTEVIn pin settings

Operating mode	ULPTMR1 register TMOD1 bit	ULPTMR3 register TEVPOL bit	ULPTEVIn pin input
Timer mode	0	0 or 1	Input disabled
Event counter mode	1	0	Normal input
		1	Inverted input

Table 23.7 ULPTOn pin settings

Operating mode	ULPTIOC register TOE bit	ULPTMR3 register TOPOL bit	ULPTOn pin output
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

Table 23.8 ULPTOAn pin settings

Operating mode	ULPTCMSR register TOEA bit	TOPOLA bit	ULPTOAn pin output
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

Table 23.9 ULPTOBn pin settings

Operating mode	ULPTCMSR register TOEB bit	TOPOLB bit	ULPTOBn pin output
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

23.4.7 Standby mode

The ULPT can operate in Software Standby mode or Deep Software Standby mode1. Set it to each Standby mode with count operation start (ULPTCR.TSTART = 1, ULPTCR.TCSTF = 1).

Table 23.10 lists the settings available in Software Standby mode or Deep Software Standby mode1.

Table 23.10 Settings available in Software Standby mode or Deep Software Standby mode1

Section No.	Count operation			Count source	CPU recovery source
	Mode 1	Mode 2	Mode 3		
section 23.4.1.1. Timer & continuous mode	Timer	Continuous	—	ULPTLCLK, ULPTSCLK	• Underflow • Compare match A/B
section 23.4.1.5. Common & continuous & count start mode			Count start		
section 23.4.1.6. Common & continuous & count restart mode			Count restart		
section 23.4.1.2. Timer & one-shot mode		One-shot	—		
section 23.4.1.7. Common & one-shot & count start mode			Count start		
section 23.4.1.8. Common & one-shot & count restart mode			Count restart		
section 23.4.1.3. Event counter & continuous & count enable mode	Event counter	Continuous	Count enable	ULPTEVIn pin*1	• Underflow • Compare match A/B
section 23.4.1.5. Common & continuous & count start mode			Count start		
section 23.4.1.6. Common & continuous & count restart mode			Count restart		
section 23.4.1.4. Event counter & one-shot & count enable mode		One-shot	Count enable		
section 23.4.1.7. Common & one-shot & count start mode			Count start		
section 23.4.1.8. Common & one-shot & count restart mode			Count restart		

Note: The ULPT can operate in all modes even in Software Standby mode or Deep Software Standby mode1. For details about the count operations and register settings, see section 23.4.1. Count operation.

Note 1. Only ULPTEVIn pin can be used in Deep Software Standby mode1.

Only in event counter mode (ULPTMR1.TMOD1 = 1), set the ULPTIOC.TIPF[1:0] bits to 00 (no filter) before entering standby mode.

In Software Standby mode or Deep Software Standby mode1, the digital filter clock (PCLKB) stops (the digital filter does not function).

It is prohibited to rewrite the ULPTCNT, ULPTCMA, and ULPTCMB registers immediately before setting each standby mode.

If the ULPTCNT, ULPTCMA, and ULPTCMB registers are rewritten while the counter is running, set each standby mode after four or more cycles of the count source.

23.4.8 Interrupt sources

The ULPT has three interrupt sources: ULPTI, ULPTCMAI, and ULPTCMBI.

Table 23.11 lists the interrupt sources.

Table 23.11 ULPT interrupt sources

Name	Interrupt source	DMAC/DTC activation
ULPTn_ULPTI	When the counter underflows	Possible
ULPTn_ULPTCMAI	When the ULPTCNT value matches the ULPTCMA value	Possible
ULPTn_ULPTCMBI	When the ULPTCNT value matches the ULPTCMB value	Possible

23.4.9 Event output to the ELC

The ULPT is capable of link operation for a specified module via the event link controller (ELC) by using the interrupt request signal as an event signal.

The ULPT outputs an event signal when compare match A/compare match B or an underflow occurs. For details, see section 18, Event Link Controller (ELC).

23.5 Usage notes

23.5.1 Start and stop control of the counter

Condition A: In timer mode (ULPTMR1.TMOD1 = 0), continuous mode (ULPTMR3.TCNTCTL = 0), and count enable mode (ULPTMR3.TEECTL[1:0] = 00)

When the counter is stopped (with the TCSTF bit of the ULPTCR register set to 0), even if 1 (start the counter) is set to the TSTART bit of the ULPTCR register, the TCSTF bit remains at 0 (counter stopped) for five cycles of the count source.

Do not access the registers*1 related to the ULPT other than the TCSTF bit until the TCSTF bit is set to 1 (counter running).

When the counter is running (with the TCSTF bit set to 1), even if 0 (stop the counter) is set to the TSTART bit, the TCSTF bit remains at 1 (counter running) for five cycles of the count source.

Do not access the registers*1 related to the ULPT other than the TCSTF bit until the TCSTF bit is set to 0 (counter stopped).

Before changing the TSTART bit from 0 to 1, clear the interrupt register. For details, see section 13, Interrupt Controller Unit (ICU).

Note 1. Registers related to the ULPT: ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR, ULPTMR1, ULPTMR2, ULPTMR3, ULPTIOC, ULPTISR, and ULPTCMSR

Condition B: In event counter mode (ULPTMR1.TMOD1 = 1), continuous mode (ULPTMR3.TCNTCTL = 0), and count enable mode (ULPTMR3.TEECTL[1:0] = 00)

When the counter is stopped (with the TCSTF bit of the ULPTCR register set to 0), even if 1 (start the counter) is set to the TSTART bit of the ULPTCR register, the TCSTF bit remains at 0 (counter stopped) for two cycles of the count source.

Do not access the registers*1 related to the ULPT other than the TCSTF bit until the TCSTF bit is set to 1 (counter running).

When the counter is running (with the TCSTF bit set to 1), even if 0 (stop the counter) is set to the TSTART bit, the TCSTF bit remains at 1 (counter running) for two cycles of the count source.

Do not access the registers*1 related to the ULPT other than the TCSTF bit until the TCSTF bit is set to 0 (counter stopped).

Before changing the TSTART bit from 0 to 1, clear the interrupt register. For details, see section 13, Interrupt Controller Unit (ICU).

Note 1. Registers related to the ULPT: ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR, ULPTMR1, ULPTMR2, ULPTMR3, ULPTIOC, ULPTISR, and ULPTCMSR

Condition C: In count start mode or count restart mode (ULPTMR3.TEECTL[1:0] bits = 10 or 11)

When the counter is stopped (with the TCSTF bit of the ULPTCR register set to 0), even if 1 is set to the TSTART bit of the ULPTCR register and then the counter is started on an edge trigger of the ULPTEEn pin, the TCSTF bit remains at 0 (counter stopped) for five cycles of the count source.

Do not access the registers*1 related to the ULPT other than the TCSTF bit until the TCSTF bit is set to 1 (counter running).

When the counter is running (with the TCSTF bit set to 1), even if 0 (stop the counter) is set to the TSTART bit, the TCSTF bit remains at 1 (counter running) for five cycles of the count source.

Do not access the registers*1 related to the ULPT other than the TCSTF bit until the TCSTF bit is set to 0 (counter stopped).

Before changing the TSTART bit from 0 to 1, clear the interrupt register. For details, see section 13, Interrupt Controller Unit (ICU).

Note 1. Registers related to the ULPT: ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR, ULPTMR1, ULPTMR2, ULPTMR3, ULPTIOC, ULPTISR, and ULPTCMSR

23.5.2 Access to flags (TUNDF, TCMAF, and TCMBF bits in ULPTCR register)

It is recommended to set 1 to the TUNDF, TCMAF and TCMBF bits when changing the TSTART or TSTOP bits to prevent clearing TUNDF, TCMAF, or TCMBF by mistake. At this time, setting 1 to TUNDF, TCMAF and TCMBF bits is ignored.

23.5.3 Access to the ULPTCNT, ULPTCMA, and ULPTCMB registers

When setting to the same register sequentially while the counter is running (ULPTCR.TCSTF bit = 1), leave three or more cycles of count source between writes. The following registers are applicable:

- ULPTCNT
- ULPTCMA
- ULPTCMB

23.5.4 Mode changes

Registers related to the operating mode of the ULPT (ULPTMR1, ULPTMR2, ULPTMR3, ULPTIOC, ULPTISR, and ULPTCMSR) can be changed only while the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0). Do not change these registers while the counter is running.

If a register related to the operating mode of the ULPT is changed, the values of the TUNDF, TCMAF, and TCMBF bits of the ULPTCR register are undefined. Before starting the counter, write 0 to the TUNDF, TCMAF, and TCMBF bits.

- TUNDF (no underflow)
- TCMAF (no match for compare match A)
- TCMBF (no match for compare match B)

23.5.5 Setting the ULPTOn, ULPTOAn, and ULPTOBn pins

To use the ULPTOn, ULPTOAn, and ULPTOBn pins as output pins, set up the ULPT, and then set PmnPFS.PMR bit to 1 after determining the initial output values.

23.5.6 Calculating the number of events

In event counter mode, the number of events is calculated by using the following formula:

Number of events = Initial counter value (ULPTCNT register write value) - Counter value at the end of a valid event

23.5.7 When the counter is stopped forcibly by the TSTOP bit

Do not access the ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR, ULPTMR1, ULPTMR2, or ULPTMR3 register during one cycle of the count source after stopping the counter forcibly by the ULPTCR.TSTOP bit.

23.5.8 Digital filter

When using a digital filter, do not start the timer within five cycles of the digital filter clock after setting the ULPTIOC.TIPF[1:0] bits.

Also, when ULPTMR3.TEVPOL bit is changed while a digital filter is in use, do not start the timer within five cycles of the digital filter clock after setting the bit.

In Software Standby mode or Deep Software Standby mode1, the digital filter clock (PCLKB) stops and the digital filter does not function. Before entering Software standby mode or Deep Software Standby mode1, set the ULPTIOC.TIPF[1:0] bits to 00 (no filter).

23.5.9 Restrictions on Event Counting during Deep Software Standby Mode 1

If external events are to be counted continuously during Deep Software Standby mode 1, the ULPTEEn-DS and ULPTEVIn-DS pins should be held low during the transition to Deep Software Standby mode 1.

The device that generates the external events must be notified of the transition to Deep Software Standby mode 1 using the general I/O port.

An example of the procedure is shown below.(Examples of flow)

(a) Procedure on the MCU

1. During initial setup, set the port for notifying the external device as a general output port and output low.
2. Before transitioning to Deep Software Standby mode 1, set the PODR bit of the above port to 1.
3. Read the PODR bit and confirm that it is set to 1.
4. After 10 µs has passed, execute the WFI instruction.

(b) Procedure on the External Device

1. When the above port goes high, hold the ULPTEEn-DS and ULPTEVIn-DS pins low within 10 µs. If the time exceeds 10 µs, add the excess time to the time in 4 above. If the time is less than 10 µs, subtract the reduced time.
2. After 1 ms or more has passed, resume the event output.

23.5.10 Module-stop function

ULPT operation can be disabled or enabled using Module Stop Control Register E (MSTPCRE). The ULPT module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

23.5.11 Setting the ULPTEEn and ULPTEVIn pins

To use the ULPTEEn and ULPTEVIn pins as input pins, set up the ULPT, and then set PmnPFS.PMR bit to 1.

24. Realtime Clock (RTC)

24.1 Overview

The realtime clock (RTC) has two counting modes, calendar count mode and binary count mode, that are used by switching register settings. For calendar count mode, the RTC has a 100-year calendar from 2000 to 2099 and automatically adjusts dates for leap years. For binary count mode, the RTC counts seconds and retains the information as a serial value. Binary count mode can be used for calendars other than the Gregorian (Western) calendar.

The sub-clock oscillator or LOCO can be selected as the count source of the time counters. The RTC uses a 128-Hz clock acquired by dividing the count source by a prescaler. Year, month, date, day-of-week, a.m. /p.m. (in 12-hour mode), hour, minute, second, or 32-bit binary is counted by 1/128 second.

Table 24.1 lists the RTC specifications, Figure 24.1 shows a block diagram, and Table 24.2 lists the I/O pins.

Table 24.1 RTC specifications

Parameter	Specifications
Count mode	Calendar count mode/binary count mode
Count source*1	Sub-clock(XCIN), External clock input(EXCIN) or LOCO
Clock and calendar functions	● Calendar count mode – Year, month, date, day of week, hour, minute, second are counted, BCD display – 12 hours/24 hours mode switching function – 30 seconds adjustment function (a number less than 30 is rounded down to 00 seconds, and 30 seconds or more are rounded up to 1 minute) – Automatic adjustment function for leap years ● Binary count mode – Count seconds in 32 bits, binary display ● Shared by both modes – Start/stop function – The sub-second digit is displayed in binary units (1 Hz, 2 Hz, 4 Hz, 8 Hz, 16 Hz, 32 Hz, or 64 Hz) – Clock error correction function – Clock (1-Hz/64-Hz) output
Interrupts	● Alarm interrupt (RTC_ALM) – As an alarm interrupt condition, selectable for comparison with the following: • Calendar count mode: Year, month, date, day-of-week, hour, minute, or second can be selected • Binary count mode: Each bit of the 32-bit binary counter ● Periodic interrupt (RTC_PRD) – 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second can be selected as an interrupt period. ● Carry interrupt (RTC_CUP) – An interrupt is generated at either of the following conditions: • When a carry from the 64-Hz counter to the second counter is generated. • When the 64-Hz counter is changed and the R64CNT register is read at the same time. (32-KHz count mode is only for 64-Hz counter reading) ● Return from Software Standby or Deep Software Standby mode can be performed by the alarm interrupt or periodic interrupt
Time capture function	● Times can be captured when the edge of the time capture event input pin is detected. For every event input, month, date, hour, minute, and second are captured or the 32-bit binary counter value is captured. ● Interrupt can be generated when the edge of the time capture event input is detected. The time capture event input pin and IRQ are shared.
Event link function	Periodic event output (RTC_PRD)
TrustZone filter	Security and Privilege attribution can be set

Note 1. The frequency of the peripheral module clock (PCLKB) ≥ the frequency of the count source should be satisfied.

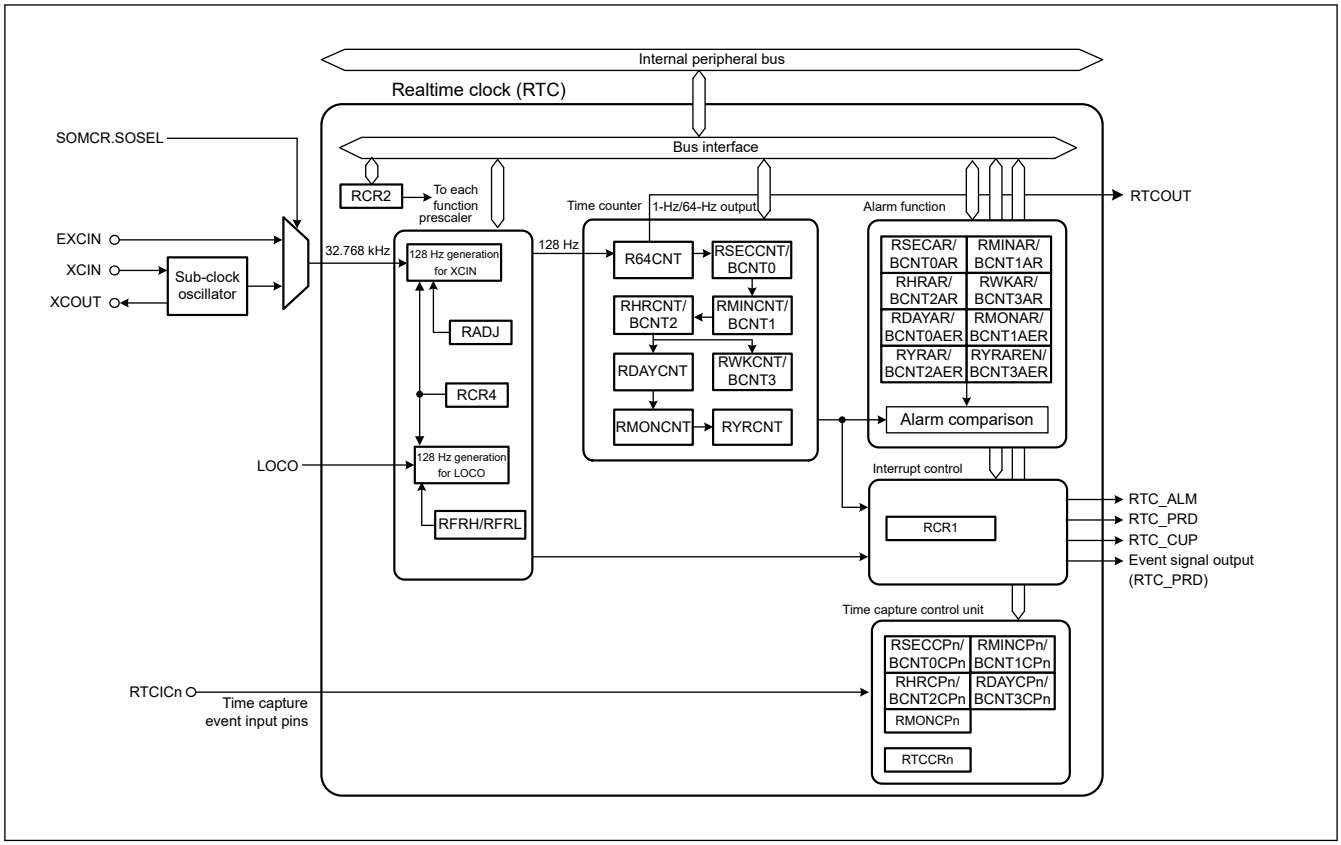


Figure 24.1 RTC block diagram

Table 24.2 RTC I/O pins

Pin name	I/O	Description
XCIN	Input	Connect a 32.768-kHz crystal to these pins
XCOUT	Output	
EXCIN	Input	External sub-clock input
RTCOUT	Output	This pin is used to output a 1-Hz/64-Hz waveform, but not in Deep Software Standby mode
RTCICn (n = 0 to 2)	Input	Time capture event input pins RTCICn can be controlled by the VBTICTLR register. For more information, see section 11, Battery Backup Function and section 19, I/O Ports.

24.2 Register Descriptions

Write or read from the RTC registers as described in section 24.6.5. Notes on Writing to and Reading from Registers.

If the value in an RTC register after a reset is given as x (undefined bits) in the list, it is not initialized by a reset. When RTC enters the reset state or a low power state during counting operations, for example, while the RCR2.START bit is 1, the year, month, day of the week, date, hours, minutes, seconds, and 64-Hz counters continue to operate.

Note: A reset generated while writing to a register might destroy the register value. In addition, do not allow the MCU to enter Software Standby mode, Deep Software Standby mode, or battery backup state immediately after setting any of these registers. For details, see section 24.6.4. Transitions to Low Power Modes after Setting Registers.

24.2.1 R64CNT : 64-Hz Counter

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	F1HZ	F2HZ	F4HZ	F8HZ	F16HZ	F32HZ	F64HZ
Value after reset:	0	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
0	F64HZ	64-Hz Flag This bit indicates the 64-Hz state of the sub-second digit.	R
1	F32HZ	32-Hz Flag This bit indicates the 32-Hz state of the sub-second digit.	R
2	F16HZ	16-Hz Flag This bit indicates the 16-Hz state of the sub-second digit.	R
3	F8HZ	8-Hz Flag This bit indicates the 8-Hz state of the sub-second digit.	R
4	F4HZ	4-Hz Flag This bit indicates the 4-Hz state of the sub-second digit.	R
5	F2HZ	2-Hz Flag This bit indicates the 2-Hz state of the sub-second digit.	R
6	F1HZ	1-Hz Flag This bit indicates the 1-Hz state of the sub-second digit.	R
7	—	This bit is read as 0.	R

Note: S-TYPE3, P-TYPE3

The R64CNT counter is used in both calendar count mode and binary count mode. The 64-Hz counter (R64CNT) generates the period for a second by counting up periods of the 128-Hz clock. The state in the sub-second range can be confirmed by reading this counter.

This counter is set to 0x00 by an RTC software reset or an execution of a 30-second adjustment. To read this counter, follow the procedure in section 24.3.5. Reading 64-Hz Counter and Time.

24.2.2 RSECCNT : Second Counter (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	SEC10[2:0]			SEC1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	SEC1[3:0]	1-Second Count Counts from 0 to 9 every second. When a carry is generated, 1 is added to the tens place.	R/W
6:4	SEC10[2:0]	10-Second Count Counts from 0 to 5 for 60-second counting.	R/W
7	—	The read value is undefined. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RSECCNT counter sets and counts the BCD-coded second value. It counts the carries generated once per second in the 64-Hz counter.

The setting range is decimal 00 to 59. The RTC does not operate normally if any other value is set. Before writing to this register, you must stop the count operation using the START bit in RCR2.

To read this counter, follow the procedure in section 24.3.5. Reading 64-Hz Counter and Time.

24.2.3 RMINCNT : Minute Counter (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	MIN10[2:0]			MIN1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	MIN1[3:0]	1-Minute Count Counts from 0 to 9 every minute. When a carry is generated, 1 is added to the tens place.	R/W
6:4	MIN10[2:0]	10-Minute Count Counts from 0 to 5 for 60-minute counting.	R/W
7	—	The read value is undefined. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RMINCNT counter sets and counts the BCD-coded minute value. It counts the carries generated once every minute in the second counter.

A value from 00 through 59 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 24.3.5. Reading 64-Hz Counter and Time.

24.2.4 RHRCNT : Hour Counter (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	HR1[3:0]	1-Hour Count Counts from 0 to 9 once per hour. When a carry is generated, 1 is added to the tens place.	R/W
5:4	HR10[1:0]	10-Hour Count Counts from 0 to 2 once per carry from the ones place.	R/W
6	PM	AM/PM select for time counter setting. 0: AM 1: PM	R/W
7	—	The read value is undefined. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RHRCNT counter sets and counts the BCD-coded hour value. It counts the carries generated once per hour in the minute counter. The specifiable time differs based on the setting in the hours mode bit (RCR2.HR24):

- When the RCR2.HR24 bit is 0 – from 00 to 11 (in BCD).
- When the RCR2.HR24 bit is 1 – from 00 to 23 (in BCD).

If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, you must stop the count operation using the START bit in RCR2. The PM bit is only enabled when the RCR2.HR24 bit is 0.

Otherwise, the setting in the PM bit has no effect. To read this counter, follow the procedure in section 24.3.5. Reading 64-Hz Counter and Time.

24.2.5 RWKCNT : Day-of-Week Counter (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	DAYW[2:0]		
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
2:0	DAYW[2:0]	Day-of-Week Counting 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting prohibited	R/W
7:3	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RWKCNT counter sets and counts in the coded day-of-week value. It counts the carries generated once per day in the hour counter. A value from 0 through 6 can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 24.3.5. Reading 64-Hz Counter and Time.

24.2.6 BCNTn : Binary Counter n (n = 0 to 3) (in Binary Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x02 + 0x02 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	BCNT[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	BCNT[7:0]	Binary Counter	R/W

Note: S-TYPE3, P-TYPE3

BCNTn is a read/write 8-bit register to access BCNT[31:0] that is a 32-bit binary counter. BCNT3 is assigned to the BCNT[31:24] bits, BCNT2 is assigned to the BCNT[23:16] bits, BCNT1 is assigned to the BCNT[15:8] bits, and BCNT0 is assigned to the BCNT[7:0] bits. BCNTn performs count operation by a carry generated for each second of the 64-Hz counter. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 24.3.5. Reading 64-Hz Counter and Time.

24.2.7 RDAYCNT : Day Counter

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x0A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	0	0	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	DATE1[3:0]	1-Day Count Counts from 0 to 9 once per day. When a carry is generated, 1 is added to the tens place.	R/W
5:4	DATE10[1:0]	10-Day Count Counts from 0 to 3 once per carry from the ones place.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RDAYCNT counter is used in calendar count mode to set and count the BCD-coded date value. It counts the carries generated once per day in the hour counter. The count operation depends on the month and whether the year is a leap year. Leap years are determined according to whether the year counter (RYRCNT) value is divisible by 400, 100, and 4.

A value from 01 through 31 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. When specifying a value, the range of specifiable days depends on the month and whether the year is a leap year. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 24.3.5. Reading 64-Hz Counter and Time.

24.2.8 RMONCNT : Month Counter

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	MON10	MON1[3:0]			
Value after reset:	0	0	0	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	MON1[3:0]	1-Month Count Counts from 0 to 9 once per month. When a carry is generated, 1 is added to the tens place.	R/W
4	MON10	10-Month Count Counts from 0 to 1 once per carry from the ones place.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RMONCNT counter is used in calendar count mode to set and count the BCD-coded month value. It counts the carries generated once per month in the date counter.

A value from 01 through 12 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 24.3.5. Reading 64-Hz Counter and Time.

24.2.9 RYRCNT : Year Counter

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x0E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	YR1[3:0]	1-Year Count Counts from 0 to 9 once per year. When a carry is generated, 1 is added to the tens place.	R/W
7:4	YR10[3:0]	10-Year Count Counts from 0 to 9 once per carry from ones place. When a carry is generated in the tens place, 1 is added to the hundreds place.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RYRCNT counter is used in calendar count mode to set and count the BCD-coded year value. It counts the carries generated once per year in the month counter.

A value from 00 through 99 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 24.3.5. Reading 64-Hz Counter and Time.

24.2.10 RSECAR : Second Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB	SEC10[2:0]			SEC1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	SEC1[3:0]	1 Second Value for the ones place of seconds.	R/W
6:4	SEC10[2:0]	10 Seconds Value for the tens place of seconds.	R/W
7	ENB	ENB 0: Do not compare register value with RSECCNT counter value 1: Compare register value with RSECCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

RSECAR is an alarm register associated with the BCD-coded second counter RSECCNT. When the ENB bit is set to 1, the RSECAR value is compared with the RSECCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR

- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. RSECAR values from 00 through 59 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0x00 by an RTC software reset.

24.2.11 RMINAR : Minute Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x12

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB	MIN10[2:0]			MIN1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	MIN1[3:0]	1 Minute Value for the ones place of minutes.	R/W
6:4	MIN10[2:0]	10 Minutes Value for the tens place of minutes.	R/W
7	ENB	ENB 0: Do not compare register value with RMINCNT counter value 1: Compare register value with RMINCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

RMINAR is an alarm register associated with the BCD-coded minute counter RMINCNT. When the ENB bit is set to 1, the RMINAR value is compared with the RMINCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. RMINAR values from 00 through 59 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0x00 by an RTC software reset.

24.2.12 RHRAR : Hour Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x14

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	HR1[3:0]	1 Hour Value for the ones place of hours.	R/W
5:4	HR10[1:0]	10 Hours Value for the tens place of hours.	R/W
6	PM	AM/PM select for alarm setting. 0: AM 1: PM	R/W
7	ENB	ENB 0: Do not compare register value with RHRCNT counter value 1: Compare register value with RHRCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

RHRAR is an alarm register associated with the BCD-coded hour counter RHRCNT. When the ENB bit is set to 1, the RHRAR value is compared with the RHRCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24):

- When the RCR2.HR24 bit is 0 – From 00 to 11 (in BCD).
- When the RCR2.HR24 bit is 1 – From 00 to 23 (in BCD).

If a value outside of this range is specified, the RTC does not operate correctly. When the RCR2.HR24 bit is 0, you must set the PM bit. When the RCR2.HR24 bit is 1, the setting in the PM bit has no effect. This register is set to 0x00 by an RTC software reset.

24.2.13 RWKAR : Day-of-Week Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x16

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB	—	—	—	—	DAYW[2:0]		
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
2:0	DAYW[2:0]	Day-of-Week Setting 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting prohibited	R/W
6:3	—	The read values are undefined. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	ENB	ENB 0: Do not compare register value with RWKCNT counter value 1: Compare register value with RWKCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

RWKAR is an alarm register associated with the coded day-of-week counter RWKCNT. When the ENB bit is set to 1, the RWKAR value is compared with the RWKCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. RWKAR values from 0 through 6 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0x00 by an RTC software reset.

24.2.14 BCNTnAR : Binary Counter n Alarm Register (n = 0 to 3) (in Binary Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x10 + 0x02 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	BCNTAR							
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	BCNTAR	Alarm register associated with the 32-bit binary counter	R/W

Note: S-TYPE3, P-TYPE3

BCNTnAR is a read/write alarm register associated with the 32-bit binary counter. BCNT3AR is assigned to the BCNTAR[31:24] bits, BCNT2AR is assigned to the BCNTAR[23:16] bits, BCNT1AR is assigned to the BCNTAR[15:8] bits, and BCNT0AR is assigned to the BCNTAR[7:0]. This register is set to 0x00 by an RTC software reset.

24.2.15 RDAYAR : Date Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x18

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	DATE1[3:0]	1 Day Value for the ones place of days.	R/W

Bit	Symbol	Function	R/W
5:4	DATE10[1:0]	10 Days Value for the tens place of days.	R/W
6	—	The read value is undefined. The write value should be 0.	R/W
7	ENB	ENB 0: Do not compare register value with RDAYCNT counter value 1: Compare register value with RDAYCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

RDAYAR is an alarm register associated with the BCD-coded date counter RDAYCNT. When the ENB bit is set to 1, the RDAYAR value is compared with the RDAYCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. The RDAYAR values from 01 through 31 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0x00 by an RTC software reset.

24.2.16 RMONAR : Month Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x1A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB	—	—	MON10	MON1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	MON1[3:0]	1 Month Value for the ones place of months.	R/W
4	MON10	10 Months Value for the tens place of months.	R/W
6:5	—	The read values are undefined. The write value should be 0.	R/W
7	ENB	ENB 0: Do not compare register value with RMONCNT counter value 1: Compare register value with RMONCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

RMONAR is an alarm register associated with the BCD-coded month counter RMONCNT. When the ENB bit is set to 1, the RMONAR value is compared with the RMONCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR

- RDAYAR
- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. The RMONAR values from 01 through 12 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0x00 by an RTC software reset.

24.2.17 RYRAR : Year Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x1C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	YR1[3:0]	1 Year Value for the ones place of years.	R/W
7:4	YR10[3:0]	10 Years Value for the tens place of years.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

RYRAR is an alarm register associated with the BCD-coded year counter RYRCNT. The RYRAR values from 00 through 99 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0x0000 by an RTC software reset.

24.2.18 RYRAREN : Year Alarm Enable Register (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x1E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
6:0	—	The read values are undefined. The write value should be 0.	R/W
7	ENB	ENB 0: Do not compare register value with the RYRCNT counter value 1: Compare register value with the RYRCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

When the ENB bit in the RYRAREN register is set to 1, the RYRAR value is compared with the RYRCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR

- RDAYAR
- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. This register is set to 0x00 by an RTC software reset.

24.2.19 BCNTnAER : Binary Counter n Alarm Enable Register (n = 0 to 3) (in Binary Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x18 + 0x02 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	ENB[7:0]	Setting the alarm enable associated with the 32-bit binary counter	R/W

Note: S-TYPE3, P-TYPE3

BCNTnAER is a read/write register for setting the alarm enable (BCNTAER) associated with the 32-bit binary counter. BCNT3AER is assigned to the BCNTAER.ENB[31:24] bits, BCNT2AER register is assigned to the BCNTAER.ENB[23:16] bits, BCNT1AER is assigned to the BCNTAER.ENB[15:8] bits, and BCNT0AER is assigned to the BCNTAER.ENB[7:0] bits. The binary counter (BCNT[31:0]) associated with the BCNTAER.ENB[31:0] bits that are set to 1 is compared with the binary alarm register (BCNTAR) and, when all match, the IR flag associated with the RTC_ALM interrupt is set to 1. This register is set to 0x00 by an RTC software reset.

24.2.20 RCR1 : RTC Control Register 1

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x22

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PES[3:0]				RTCOS	PIE	CIE	AIE
Value after reset:	x	x	x	x	0	x	0	x

Bit	Symbol	Function	R/W
0	AIE	Alarm Interrupt Enable 0: Disable alarm interrupt requests 1: Enable alarm interrupt requests	R/W
1	CIE	Carry Interrupt Enable 0: Disable carry interrupt requests 1: Enable carry interrupt requests	R/W
2	PIE	Periodic Interrupt Enable 0: Disable periodic interrupt requests 1: Enable periodic interrupt requests	R/W
3	RTCOS	RTCOUT Output Select 0: Outputs 1 Hz on RTCOUT 1: Outputs 64 Hz RTCOUT	R/W

Bit	Symbol	Function	R/W
7:4	PES[3:0]	Periodic Interrupt Select 0x6: Generate periodic interrupt every 1/256 second*1 0x7: Generate periodic interrupt every 1/128 second 0x8: Generate periodic interrupt every 1/64 second 0x9: Generate periodic interrupt every 1/32 second 0xA: Generate periodic interrupt every 1/16 second 0xB: Generate periodic interrupt every 1/8 second 0xC: Generate periodic interrupt every 1/4 second 0xD: Generate periodic interrupt every 1/2 second 0xE: Generate periodic interrupt every 1 second 0xF: Generate periodic interrupt every 2 seconds Others: Do not generate periodic interrupts	R/W

Note: S-TYPE3, P-TYPE3
Note 1. When LOCO is selected (RCR4.RCKSEL = 1) while PES[3:0] = 0x6, a periodic interrupt is generated every 1/128 second.

The RCR1 register is used in both calendar count mode and binary count mode. Bits AIE, PIE, and PES[3:0] are updated synchronously with the count source. When the RCR1 register is modified, check that all the bits are updated before proceeding.

AIE bit (Alarm Interrupt Enable)

The AIE bit enables or disables alarm interrupt requests.

If the times indicated in the counters and alarm settings match in Deep Software Standby mode, the MCU returns from Deep Software Standby mode regardless of the AIE bit value.

CIE bit (Carry Interrupt Enable)

The CIE bit enables or disables interrupt requests when a carry to the RSECCNT/BCNT0 register occurs, or when a carry to the 64-Hz counter (R64CNT) occurs while reading the 64-Hz counter.

PIE bit (Periodic Interrupt Enable)

The PIE bit enables or disabled a periodic interrupt.

If the periods indicated in the counters and PES[3:0] settings match in Deep Software Standby mode, the MCU returns from Deep Software Standby mode regardless of the PIE bit value.

RTCOS bit (RTCOUT Output Select)

The RTCOS bit selects the RTCOUT output period. The RTCOS bit must be rewritten while the count operation is stopped (RCR2.START = 0) and the RTCOUT output is disabled (RCR2.RTCOE = 0). When RTCOUT is output to an external pin, the RCR2.RTCOE bit must be enabled.

PES[3:0] bits (Periodic Interrupt Select)

The PES[3:0] bits specify the period for the periodic interrupt. A periodic interrupt is generated with the period specified in these bits.

24.2.21 RCR2 : RTC Control Register 2 (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x24

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CNTMD	HR24	AADJP	AADJE	RTCOE	ADJ30	RESET	START
Value after reset:	x	x	x	x	0	0	0	x

Bit	Symbol	Function	R/W
0	START	Start 0: Stop prescaler and time counter 1: Operate prescaler and time counter normally	R/W

Bit	Symbol	Function	R/W
1	RESET	RTC Software Reset 0: In writing: Invalid (writing 0 has no effect). In reading: Normal time operation in progress, or an RTC software reset has completed. 1: In writing: Initialize the prescaler and target registers for RTC software reset*1. In reading: RTC software reset in progress.	R/W
2	ADJ30	30-Second Adjustment 0: In writing: Invalid (writing 0 has no effect). In reading: Normal time operation in progress, or 30-second adjustment has completed. 1: In writing: Execute 30-second adjustment. In reading: 30-second adjustment in progress.	R/W
3	RTCOE	RTCOUT Output Enable 0: Disable RTCOUT output 1: Enable RTCOUT output	R/W
4	AADJE	Automatic Adjustment Enable*2*3 0: Disable automatic adjustment 1: Enable automatic adjustment	R/W
5	AADJP	Automatic Adjustment Period Select*2*3 0: The RADJ.ADJ[5:0] setting from the count value of the prescaler every minute. 1: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every 10 seconds.	R/W
6	HR24	Hours Mode*3 0: Operate RTC in 12-hour mode 1: Operate RTC in 24-hour mode	R/W
7	CNTMD	Count Mode Select*4 0: Calendar count mode 1: Binary count mode	R/W

Note: S-TYPE3, P-TYPE3
Note 1. R64CNT, RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, RYRAR, RYRAREN, RADJ, RTCCRn, RSECCPn, RMINCPn, RHRCPn, RDAYCPn, RMONCPn, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP.
Note 2. When LOCO is selected, the setting of this bit is disabled.
Note 3. When rewriting this bit, confirm that the value has been rewritten before performing the following processing. See section 24.6.5. Notes on Writing to and Reading from Registers for notes on register writing/reading.
Note 4. When rewriting this bit, confirm that the value has been rewritten before performing the following processing.

The RCR2 register is related to hours mode, automatic adjustment function, enabling RTCOUT output, 30-second adjustment, RTC software reset, and controlling count operation.

START bit (Start)

The START bit stops or restarts the prescaler or time counter operation. This bit is updated in synchronization with the next cycle of the count source. When the START bit is modified, check that the bit is updated before proceeding.

RESET bit (RTC Software Reset)

The RESET bit initializes the prescaler and registers to be reset by RTC software. When 1 is written to this bit, initialization starts in synchronization with the count source. When the initialization is complete, the RESET bit is automatically set to 0. Check that this bit is 0 before proceeding.

ADJ30 bit (30-Second Adjustment)

The ADJ30 bit is for 30-second adjustment.

When 1 is written to the ADJ30 bit, the RSECCNT value of less than 30 seconds is rounded down to 00 second and the value of 30 seconds or more is rounded up to 1 minute.

The 30-second adjustment is performed in synchronization with the count source. When 1 is written to this bit, the ADJ30 bit is automatically set to 0 after the 30-second adjustment completes. If 1 is written to the ADJ30 bit, check that the bit is 0 before proceeding. When the 30-second adjustment is performed, the prescaler and R64CNT are also reset. The ADJ30 bit is set to 0 by an RTC software reset.

RTCOE bit (RTCOUT Output Enable)

The RTCOE bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin.

Use the START bit to stop counting before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time.

When RTCOUT is to be output from an external pin, enable the RTCOE bit and set up the port control for the pin.

AADJE bit (Automatic Adjustment Enable)

The AADJE bit controls (enables or disables) automatic adjustment.

Set the plus-minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit.

The AADJE bit is set to 0 by an RTC software reset.

AADJP bit (Automatic Adjustment Period Select)

The AADJP bit selects the automatic-adjustment period.

Set the plus-minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJP bit.

The AADJP bit is set to 0 by an RTC software reset.

HR24 bit (Hours Mode)

The HR24 bit specifies whether the RTC operates in 12- or 24-hour mode.

Use the START bit to stop counting before changing the value of the HR24 bit. Do not stop counting (write 0 to the START bit) and change the value of the HR24 bit at the same time.

CNTMD bit (Count Mode Select)

The CNTMD bit specifies whether the RTC count mode operates in calendar count mode or in binary count mode.

When setting the count mode, execute an RTC software reset and start again from the initial settings. This bit is updated in synchronization with the count source. However, the count mode switches only after the RTC software reset. (Bit switches before RTC reset, mode switches after RTC reset.)

For details on initial settings, see section 24.3.1. Outline of Initial Settings of Registers after Power On.

24.2.22 RCR2 : RTC Control Register 2 (in Binary Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x24

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CNTMD	—	AADJP	AADJE	RTCOE	—	RESET	START
Value after reset:	x	x	x	x	0	0	0	x

Bit	Symbol	Function	R/W
0	START	Start 0: Stop the 32-bit binary counter, 64-Hz counter, and prescaler 1: Operate the 32-bit binary counter, 64-Hz counter, and prescaler normally	R/W
1	RESET	RTC Software Reset 0: In writing: Invalid (writing 0 has no effect). In reading: Normal time operation in progress, or an RTC software reset has completed. 1: In writing: Initialize the prescaler and target registers for RTC software reset*1. In reading: RTC software reset in progress.	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
3	RTCOE	RTCOUT Output Enable 0: Disable RTCOUT output 1: Enable RTCOUT output	R/W
4	AADJE	Automatic Adjustment Enable*2*3 0: Disable automatic adjustment 1: Enable automatic adjustment	R/W
5	AADJP	Automatic Adjustment Period Select*2*3 0: Add or subtract RADJ.ADJ [5:0] bits from prescaler count value every 32 seconds 1: Add or subtract RADJ.ADJ [5:0] bits from prescaler count value every 8 seconds.	R/W
6	—	The read value is undefined. The write value should be 0.	R/W
7	CNTMD	Count Mode Select*4 0: Calendar count mode 1: Binary count mode	R/W

Note: S-TYPE3, P-TYPE3
Note 1. R64CNT, BCNTnAR, BCNTnAER, RADJ, RTCCRn, BCNTnCPm, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP.
Note 2. When LOCO is selected, the setting of this bit is disabled.
Note 3. When rewriting this bit, confirm that the value has been rewritten before performing the following processing. See section 24.6.5. Notes on Writing to and Reading from Registers for notes on register writing/reading.
Note 4. When rewriting this bit, confirm that the value has been rewritten before performing the following processing.

RCR2 in the binary count mode is a register related to the automatic correction function, RTCOUT output enable, RTC software reset, and count mode control.

START bit (Start)

The START bit stops or restarts the prescaler or counter (clock) operation. This bit is updated in synchronization with the count source. When the START bit is modified, check that the bit is updated before proceeding.

RESET bit (RTC Software Reset)

The RESET bit initializes the prescaler and registers to be reset by RTC software. When 1 is written to this bit, initialization starts in synchronization with the count source. When the initialization is complete, the RESET bit is automatically set to 0. When 1 is written to the RESET bit, check that the bit is 0 before proceeding.

RTCOE bit (RTCOUT Output Enable)

The RTCOE bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin.

Use the START bit to stop counting before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time. When an RTCOUT signal is to be output from an external pin, enable the port control in addition to setting this bit.

AADJE bit (Automatic Adjustment Enable)

The AADJE bit controls (enables or disables) automatic adjustment.

Set the plus-minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit. The AADJE bit is set to 0 by an RTC software reset.

AADJP bit (Automatic Adjustment Period Select)

The AADJP bit selects the automatic-adjustment period.

Correction period can be selected from 32 second units or 8 second units in binary count mode.

Set the plus-minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJP bit. The AADJP bit is set to 0 by an RTC software reset.

CNTMD bit (Count Mode Select)

The CNTMD bit specifies whether the RTC count mode operates in calendar count mode or in binary count mode.

When setting the count mode, execute an RTC software reset and start again from the initial settings. This bit is updated in synchronization with the count source. However, the count mode switches only after the RTC software reset. (Bit switches before RTC reset, mode switches after RTC reset.)

For details on initial settings, see section 24.3.1. Outline of Initial Settings of Registers after Power On .

24.2.23 RCR4 : RTC Control Register 4

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x28

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	RCKSEL
Value after reset:	0	0	0	0	0	0	0	x

Bit	Symbol	Function	R/W
0	RCKSEL	Count Source Select 0: Sub-clock oscillator is selected 1: LOCO is selected	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RCR4 register is used in both calendar count mode and binary count mode.

RCKSEL bit (Count Source Select)

The RCKSEL bit selects the count source from the sub-clock oscillator and LOCO.

The RCKSEL bit is only used in normal operation mode. When the RCKSEL bit is set to 0, the time is counted with the sub-clock oscillator. When the bit is set to 1, the time is counted with LOCO.

For details on count source setting, see section 24.3.1. Outline of Initial Settings of Registers after Power On and section 24.3.2. Clock and Count Mode Setting Procedure. The count source must be selected only once before specifying the initial settings of the RTC registers at power on.

24.2.24 RFRL : Frequency Register L

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x2C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RFC[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
15:0	RFC[15:0]	Frequency Comparison Value Write 0x00FF to this register when using the LOCO.	R/W

Note: S-TYPE3, P-TYPE3

RFRL is a register for controlling the prescaler when LOCO is selected.

The RTC time counter operates on a 128-Hz clock signal as the base clock. Therefore, when LOCO is selected, LOCO is divided by the prescaler to generate a 128-Hz clock signal. Set the frequency comparison value in the RFC[15:0] bits to generate a 128-Hz clock from the LOCO frequency. Before writing to RFC[15:0] after a cold start, write 0x0000 to the RFRH register.

A value from 0x0007 through 0x01FF can be specified as the frequency comparison value. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2. The operating frequency of the peripheral module clock and the LOCO should be such that the peripheral module clock is ≥ LOCO.

Calculation method of frequency comparison value:

RFC[15:0] = (LOCO clock frequency) / 128 - 1

When the LOCO frequency is 32.768 kHz, the RFRL register should be set to 0x00FF.

24.2.25 RFRH : Frequency Register H

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x2A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

Bit	Symbol	Function	R/W
0	RFC16	Write 0 before writing to the RFRL register after a cold start.	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

Before writing to RFRHL.RFC[15:0] after a cold start, write 0x0000 to the RFRH register.

24.2.26 RADJ : Time Error Adjustment Register

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x2E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PMADJ[1:0]		ADJ[5:0]					
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
5:0	ADJ[5:0]	Adjustment Value These bits specify the adjustment value from the prescaler.	R/W
7:6	PMADJ[1:0]	Plus-Minus 0 0: Do not perform adjustment. 0 1: Adjustment is performed by the addition to the prescaler 1 0: Adjustment is performed by the subtraction from the prescaler 1 1: Setting prohibited.	R/W

Note: S-TYPE3, P-TYPE3

The RADJ register is used in both calendar count mode and binary count mode. Adjustment is performed by the addition to or subtraction from the prescaler or 64-Hz counter. If the Automatic Adjustment Enable (RCR2.AADJE) bit is 0, adjustment is performed when writing to the RADJ. If the RCR2.AADJE bit is 1, adjustment is performed in the interval specified in the Automatic Adjustment Period Select (RCR2.AADJP) bit.

The current adjustment by software (disabling automatic adjustment) may be invalid if the following adjustment value is specified within 320 cycles of the count source after the register setting. To perform adjustment consecutively, wait for 320 cycles or more of the count source after the register setting, then specify the next adjustment value.

RADJ is updated in synchronization with the count source. When RADJ is modified, check that all the bits are updated before continuing with more processing. This register is set to 0x00 by an RTC software reset. The setting of this register is enabled only when the sub-clock oscillator is selected. When LOCO is selected, adjustment is not performed.

ADJ[5:0] bits (Adjustment Value)

The ADJ[5:0] bits specify the adjustment value (number of sub-clock cycles) from the prescaler.

PMADJ[1:0] bits (Plus-Minus)

The PMADJ[1:0] bits select whether the clock is set ahead or back depending on the error-adjustment value set in the ADJ[5:0] bits.

24.2.27 RTCCRn : Time Capture Control Register n (n = 0 to 2)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x40 + 0x02 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TCEN	—	TCNF[1:0]		—	TCST	TCCT[1:0]	
Value after reset:	x	0	x	x	0	x	x	x

Bit	Symbol	Function	R/W
1:0	TCCT[1:0]	Time Capture Control 0 0: Do not detect events 0 1: Detect rising edge 1 0: Detect falling edge 1 1: Detect both edges	R/W
2	TCST	Time Capture Status 0: No event detected 1: Event detected*1	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	TCNF[1:0]	Time Capture Noise Filter Control 0 0: Turn noise filter off 0 1: Setting prohibited 1 0: Turn noise filter on (count source) 1 1: Turn noise filter on (count source by divided by 32)	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	TCEN	Time Capture Event Input Pin Enable 0: Disable the RTCICn pin as the time capture event input pin 1: Enable the RTCICn pin as the time capture event input pin	R/W

Note: S-TYPE3, P-TYPE3
Note 1. Indicates that an event is detected. Writing 1 to this bit has no effect. Writing 0 sets this bit to 0.

The RTCCRn register is used both in calendar count mode and in binary count mode. RTCCR0, RTCCR1, and RTCCR2 control the RTCIC0, RTCIC1, and RTCIC2 pins, respectively.

RTCCRn is updated in synchronization with the count source. When RTCCRn is modified, check that all the bits except the TCST bit are updated before continuing with additional processing. This register is cleared to 0x00 by an RTC software reset. When RTCICm is used as the time capture pin, VBTICTLR.VCHnIEN (n = 0 to 2) must be set to 1.

TCCT[1:0] bits (Time Capture Control)

The TCCT[1:0] bits control the edge detection of the time capture event input pins, RTCIC0, RTCIC1, and RTCIC2. The detection edge is selectable. The TCCT[1:0] bits must be set while the VBTICTLR.VCHnIEN bit is 1.

TCST bit (Time Capture Status)

The TCST bit indicates that an event on the time capture event input pins, RTCIC0, RTCIC1, and RTCIC2, was detected. When the TCST bit is 0, no event is detected. When the TCST bit is 1, this bit indicates that an event was detected on the associated pin and the capture register is valid. When multiple events are detected, the capture time for the first event is retained.

The event is detected only during count operation (RCR2.START bit = 1). Before reading the capture register, make sure that this bit is set to 1.

Set the TCST bit while the TCCT[1:0] bits are 00b (no event is detected). The TCST bit is set to 0 in synchronization with the count source. When the TCST bit is set to 0, check that the bit is updated before continuing with additional processing.

TCNF[1:0] bits (Time Capture Noise Filter Control)

The TCNF[1:0] bits control the noise filter of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2).

When the noise filter is on, the count source divided by 1 or divided by 32 is selectable. In this case, when the input level on the time capture event input pin matches three consecutive times at the set sampling period, the input level is determined.

Set the TCNF[1:0] bits while the TCCT[1:0] bits are 00b (no event is detected). When the noise filter is used, set the TCNF[1:0] bits, wait for 3 cycles of the specified sampling period, then set the TCCT[1:0] bits. Set the TCNF[1:0] bits when the VBTICTLR.VCHnIEN bit is 1.

TCEN bit (Time Capture Event Input Pin Enable)

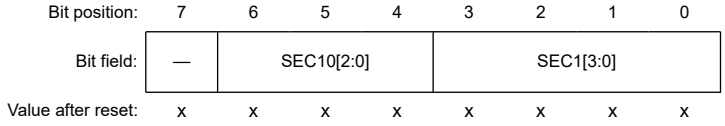
The TCEN bit enables or disables the time capture event input pins RTCIC0, RTCIC1, and RTCIC2. When the functions of the time capture event input pins are multiplexed, set VBTICTLR first. If the TCEN bit is set to 0, also set the TCCT[1:0] bits to 00b.

Before setting this bit to 1, be sure to set the count source setting bit (RCR4.RCKSEL), RTC time capture event enable bit (RCPE.RTCEN), port control setting bits (PmnPFS.PDR, and PmnPFS.PMR). For details on the port control setting bits (PmnPFS.PDR and PmnPFS.PMR), see section 19, I/O Ports.

24.2.28 RSECCPn : Second Capture Register n (n = 0 to 2) (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x52 + 0x10 × n



Bit	Symbol	Function	R/W
3:0	SEC1[3:0]	1-Second Capture Capture value for the ones place of seconds.	R
6:4	SEC10[2:0]	10-Second Capture Capture value for the tens place of seconds.	R
7	—	The read value is undefined.	R

Note: S-TYPE3, P-TYPE3

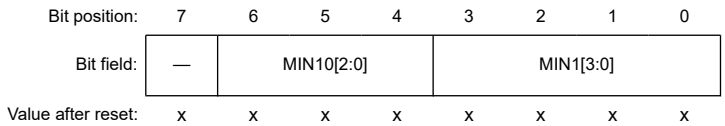
RSECCPn is a read-only register that captures the RSECCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins ares stored in the RSECCP0, RSECCP1, and RSECCP2 registers, respectively. This register is cleared to 0x00 by an RTC software reset. Before reading from this register, the time capture event detection should be stopped using the RTCCRn.TCCT[1:0] bits.

24.2.29 RMINCPn : Minute Capture Register n (n = 0 to 2) (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x54 + 0x10 × n



Bit	Symbol	Function	R/W
3:0	MIN1[3:0]	1-Minute Capture Capture value for the ones place of minutes.	R

Bit	Symbol	Function	R/W
6:4	MIN10[2:0]	10-Minute Capture Capture value for the tens place of minutes.	R
7	—	The read value is undefined.	R

Note: S-TYPE3, P-TYPE3

RMINCPn is a read-only register that captures the RMINCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMINCP0, RMINCP1, and RMINCP2 registers, respectively.

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, the time capture event detection should be stopped using the RTCCRn.TCCT[1:0] bits.

24.2.30 RHRCPn : Hour Capture Register n (n = 0 to 2) (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x56 + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	HR1[3:0]	1-Hour Capture Capture value for the ones place of hours	R
5:4	HR10[1:0]	10-Hour Capture Capture value for the tens place of hours	R
6	PM	PM 0: AM 1: PM	R
7	—	The read value is undefined.	R

Note: S-TYPE3, P-TYPE3

RHRCPn is a read-only register that captures the RHRCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RHRCP0, RHRCP1, and RHRCP2 registers, respectively.

The PM bit is only enabled when the RCR2.HR24 bit is 0 (in 12-hour mode).

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, you must stop the time capture event detection using the RTCCRn.TCCT[1:0] bits.

24.2.31 RDAYCPn : Date Capture Register n (n = 0 to 2) (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x5A + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	DATE1[3:0]	1-Day Capture Capture value for the ones place of days.	R

Bit	Symbol	Function	R/W
5:4	DATE10[1:0]	10-Day Capture Capture value for the tens place of days.	R
7:6	—	The read value is undefined.	R

Note: S-TYPE3, P-TYPE3

RDAYCPn is a read-only register that captures the RDAYCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RDAYCP0, RDAYCP1, and RDAYCP2 registers, respectively.

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, the time capture event detection should be stopped using the RTCCRn.TCCT[1:0] bits.

24.2.32 RMONCPn : Month Capture Register n (n = 0 to 2) (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x5C + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	MON10	MON1[3:0]			
Value after reset:	0	0	0	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	MON1[3:0]	1-Month Capture Capture value for the ones place of months.	R
4	MON10	10-Month Capture Capture value for the tens place of months.	R
7:5	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3

RMONCPn is a read-only register that captures the RMONCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMONCP0, RMONCP1, and RMONCP2 registers, respectively.

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, the time capture event detection should be stopped using the RTCCRn.TCCT[1:0] bits.

24.2.33 BCNTnCPm : BCNTn Capture Register m (n = 0 to 3, m = 0 to 2) (in Binary Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x52 + 0x10 × m (BCNT0CPm)
0x54 + 0x10 × m (BCNT1CPm)
0x56 + 0x10 × m (BCNT2CPm)
0x5A + 0x10 × m (BCNT3CPm)

Bit position:	7	6	5	4	3	2	1	0
Bit field:								
Value after reset:	x	x	x	x	x	x	x	x

BCNTnCPm is a read-only register that captures the BCNTn value when a time capture event is detected. BCNT3CPm is assigned to the BCNTCPm[31:24] bits, BCNT2CPm is assigned to the BCNTCPm[23:16] bits, BCNT1CPm is assigned to the BCNTCPm[15:8] bits and BCNT0CPm is assigned to the BCNTCPm[7:0] bits. The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNTnCP0, BCNTnCP1, and BCNTnCP2 registers, respectively.

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, you must stop the time capture event detection using the RTCCRn.TCCT[1:0] bits.

24.3 Operation

24.3.1 Outline of Initial Settings of Registers after Power On

After the power is turned on, perform the initial settings for the clock, count mode, time error adjustment, time, alarm, interrupts, and time capture.

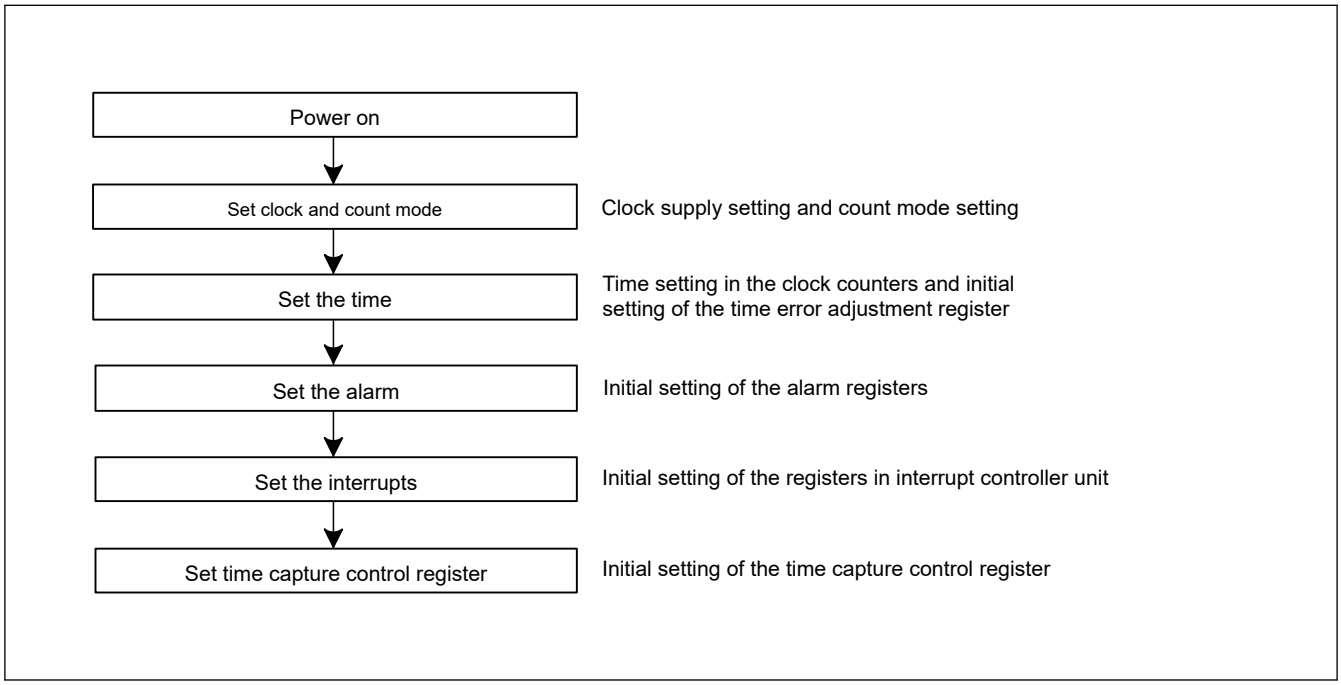
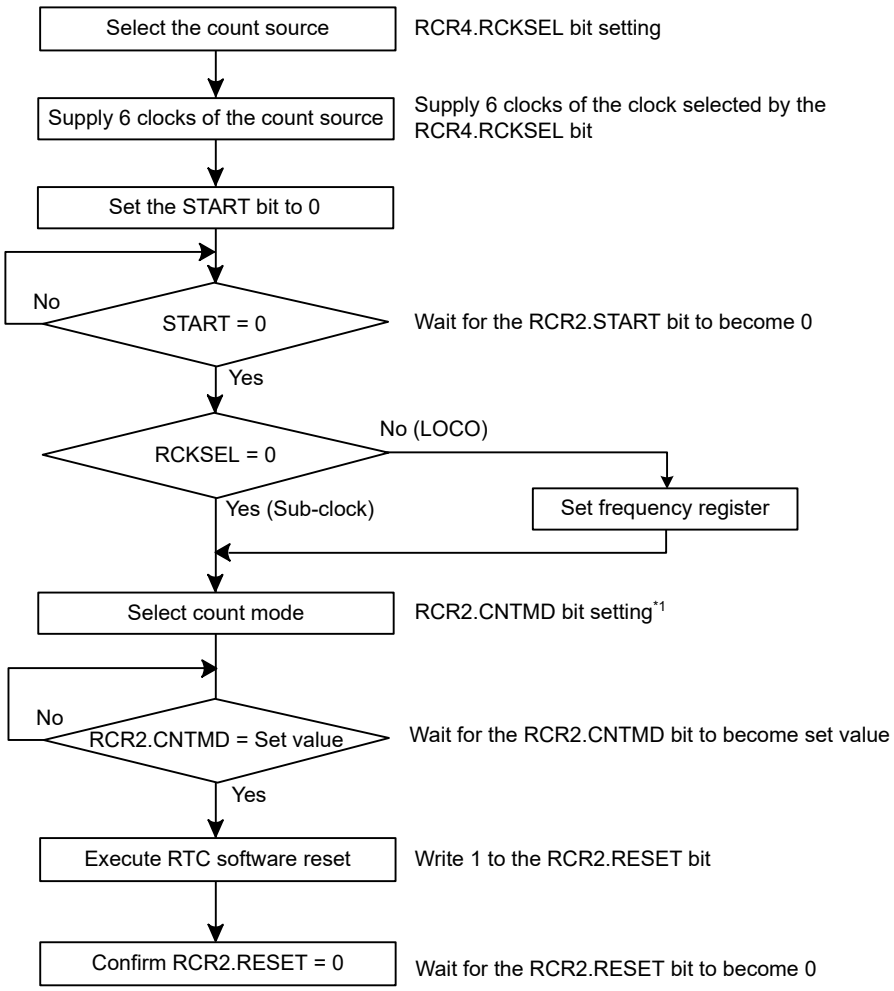


Figure 24.2 Outline of initial settings after a power on

24.3.2 Clock and Count Mode Setting Procedure

Figure 24.3 shows how to set the clock and the count mode.

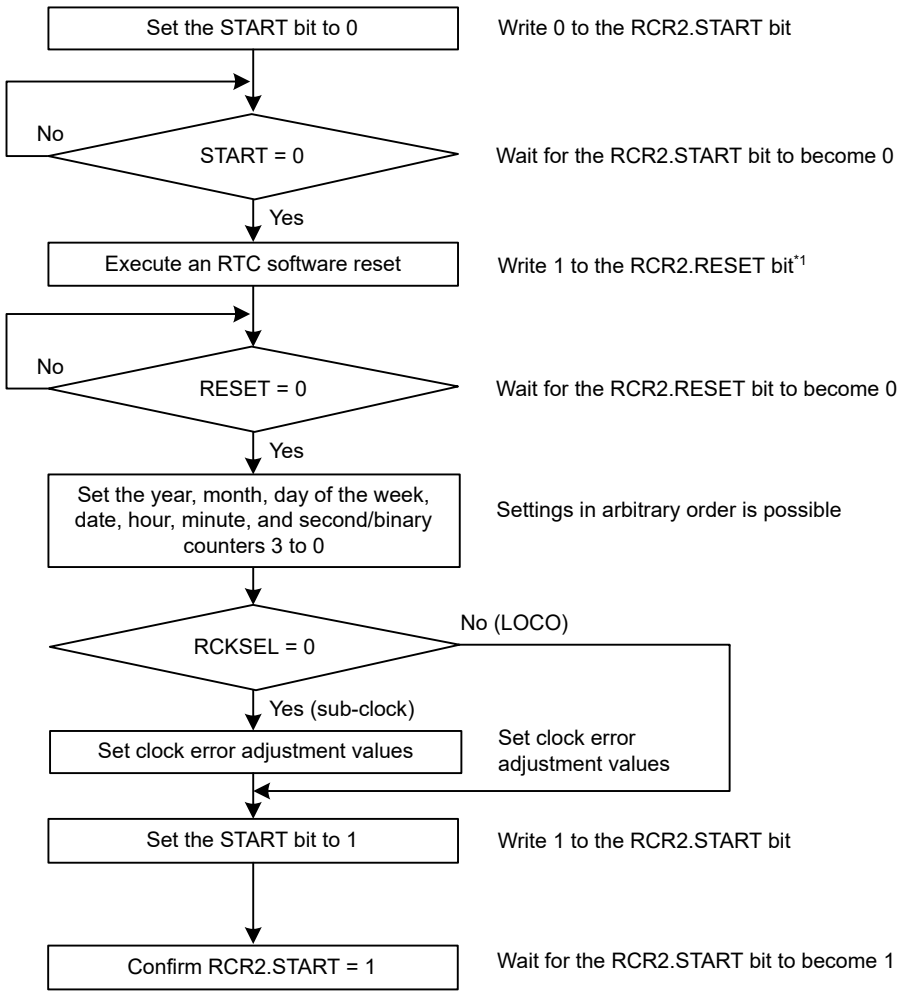


Note 1. This step is not required if the count mode is concurrently set by setting the START bit to 0. A value associated with the count mode setting must be written to the RCR2.CNTMD bit.

Figure 24.3 Clock and count mode setting procedure

24.3.3 Setting the Time

Figure 24.4 shows how to set the time.



Note 1. This step can be omitted in initial settings after power-on because an RTC software reset is executed in the clock setting procedure that is performed during initial settings at power-on.
Note 2. Set in any order.

Figure 24.4 Setting the time

24.3.4 30-Second Adjustment

Figure 24.5 shows how to execute a 30-second adjustment.

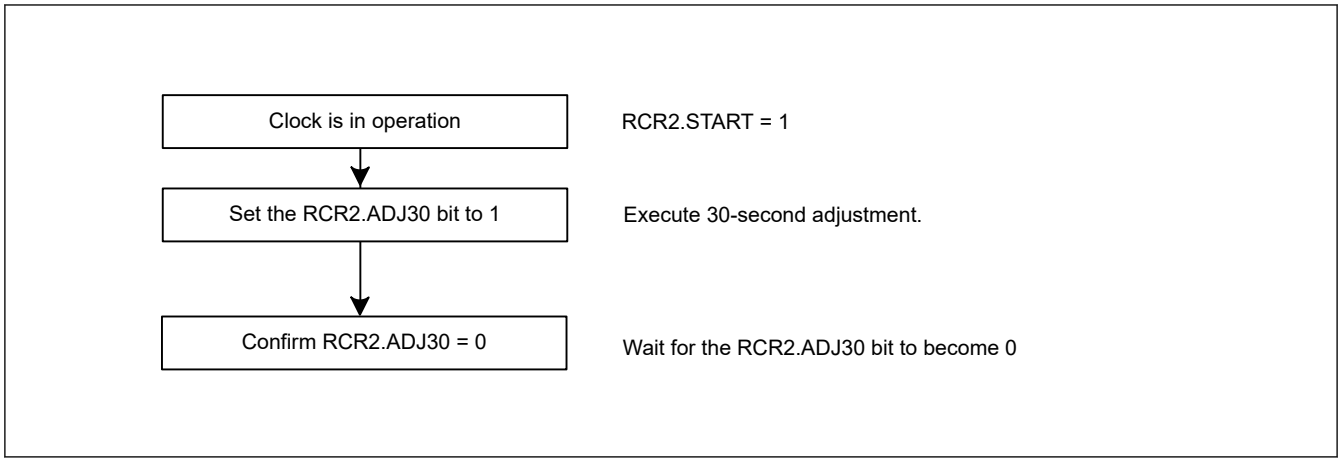
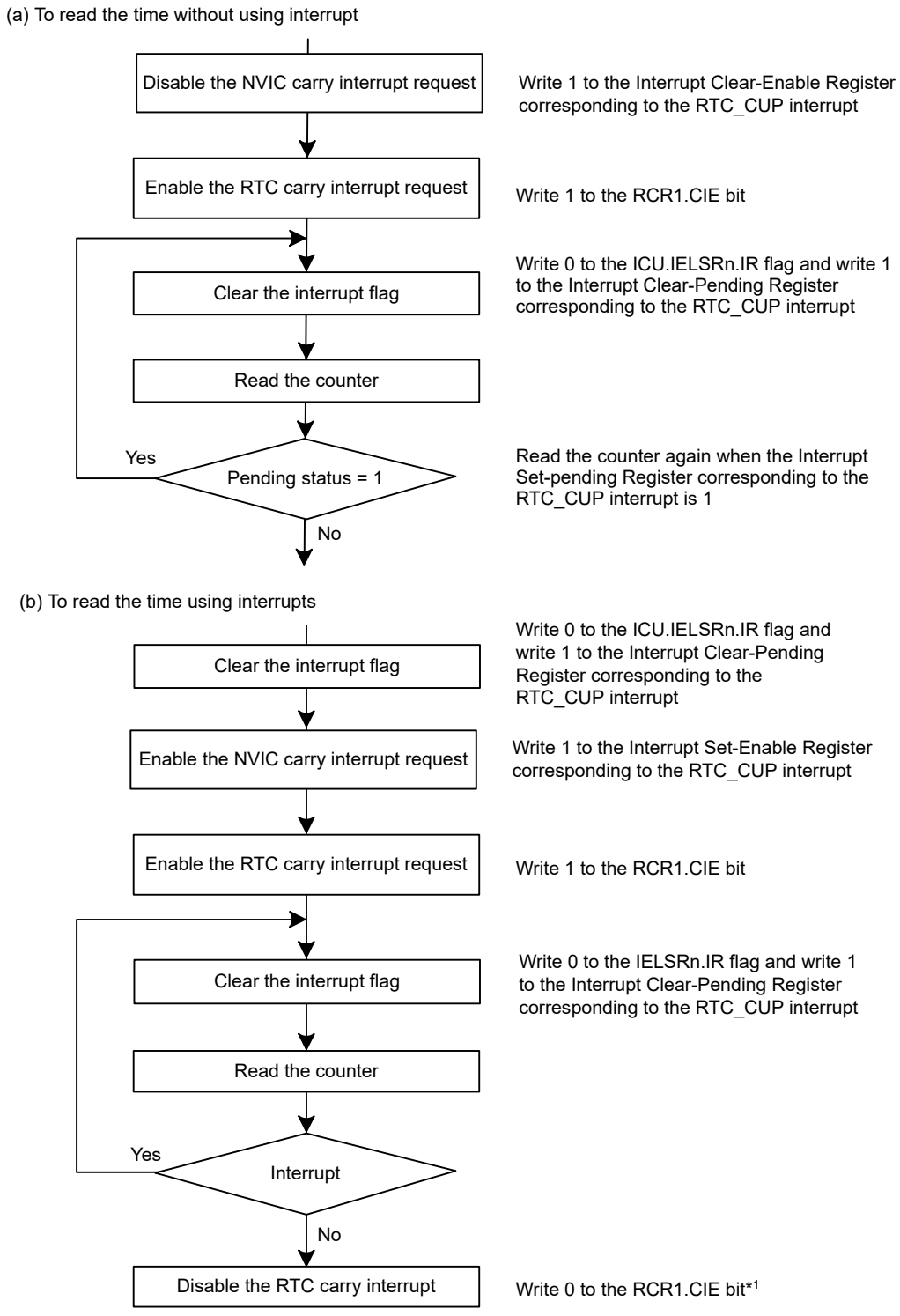


Figure 24.5 30-second adjustment

24.3.5 Reading 64-Hz Counter and Time

Figure 24.6 shows how to read a 64-Hz counter and time.



Note 1. Disable interrupts if required.

Figure 24.6 Reading time

If a carry occurs while the 64-Hz counter and time are read, the correct time is not obtained, therefore they must be read again. The procedure for reading the time without using interrupts is shown in (a) in Figure 24.6, and the procedure using carry interrupts is shown in (b). To keep the program simple, method (a) should be used in most cases.

24.3.6 Alarm Function

Figure 24.7 shows how to use the alarm function.

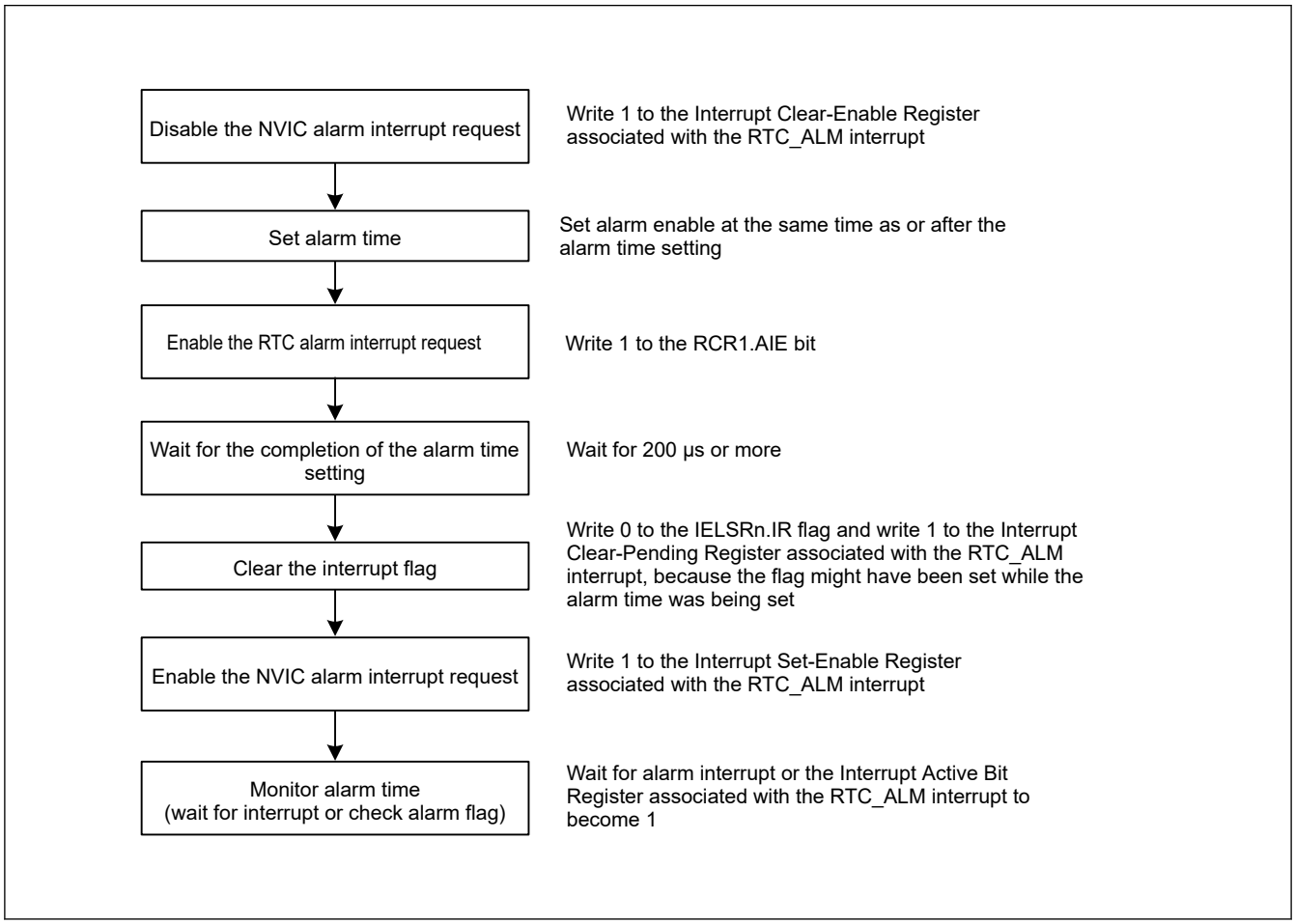


Figure 24.7 Using the alarm function

In calendar count mode, an alarm can be generated by any one of year, month, date, day-of-week, hour, minute or second, or any combination of those. Write 1 to the ENB bit in the alarm registers involved in the alarm setting, and set the alarm time in the lower bits. Write 0 to the ENB bit in registers not involved in the alarm setting.

In binary count mode, an alarm can be generated in any bit combination of 32 bits. Write 1 to the ENB bit of the Alarm Enable register associated with the target bit of the alarm, and set the alarm time in the alarm register. For bits that are not the target of the alarm, write 0 to the ENB bit of the Alarm Enable register.*1

For any of the ENB[31:0] bits that are set to 1, the bits in the corresponding positions in the binary counter (BCNT[31:0]) are compared with the values of the corresponding bits in the binary alarm registers*1. When all such bits match, the IR flag associated with the RTC_ALM interrupt is set to 1 and the corresponding bits in the Interrupt Set-Pending/Clear-Pending Registers are set to 1. Alarm detection can be confirmed by reading the Interrupt Set-Pending Register associated with the RTC_ALM interrupt, but an interrupt should be used in most cases. If 1 is set in the Interrupt Set-Enable Register associated with the RTC_ALM interrupt, an alarm interrupt is generated in the event of the alarm, enabling the alarm to be detected.

Writing 0 sets the IELSRn.IR flag associated with the RTC_ALM interrupt to 0. If interrupt is enabled, the Interrupt Set-Pending/Clear-Pending Register associated with the RTC_ALM interrupt is cleared automatically after exiting the interrupt handler. Otherwise, write 1 to the Interrupt Clear-Pending Register associated with the RTC_ALM interrupt to clear it.

When the counter and the alarm time match in a low power state, the MCU returns from the low power state.

Note 1. For any bits in the ENB bits that are set to 1, the values in the corresponding positions in the alarm registers from the following registers are compared with the corresponding bits of the counted values.

Counter registers: RSECCNT, RMINCNT, RHRCNT, RWKCNT, RDAYCNT, RMONCNT, RYRCNT

Alarm registers: RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, RYRAREN

24.3.7 Procedure for Disabling Alarm Interrupt

Figure 24.8 shows the procedure for disabling the enabled alarm interrupt request.

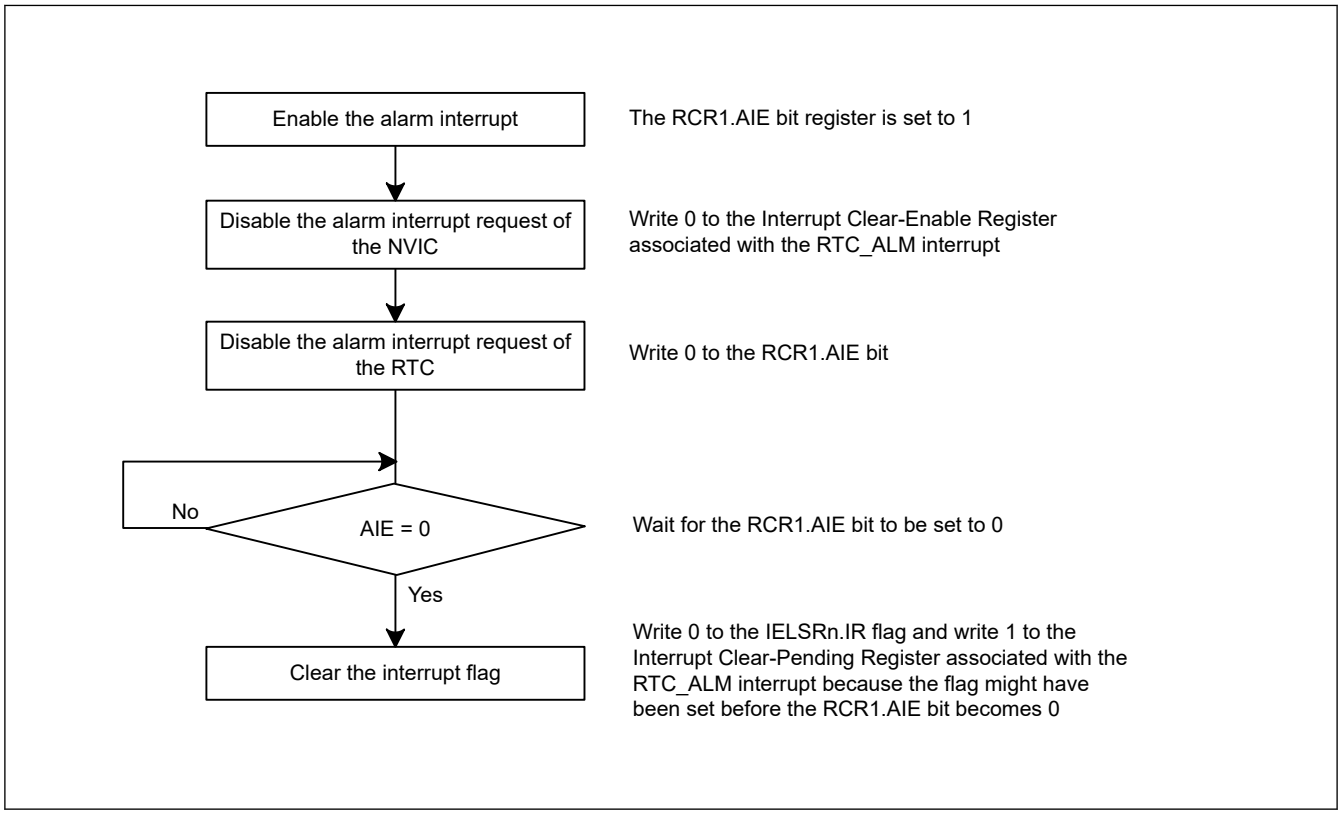


Figure 24.8 Procedure for disabling alarm interrupt request

24.3.8 Time Error Adjustment Function

The time error adjustment function is used to correct errors, running fast or slow, in the time caused by variation in the precision of oscillation by the sub-clock oscillator. Because 32768 cycles of the sub-clock oscillator constitute 1 second of operation when the sub-clock oscillator is selected, the clock runs fast if the sub-clock frequency is high and slow if the sub-clock frequency is low.

The time error adjustment functions include:

- Automatic adjustment
- Adjustment by software

Use the RCR2.AADJE bit to select automatic adjustment or adjustment by software.

24.3.8.1 Automatic adjustment

Enable automatic adjustment by setting the RCR2.AADJE bit to 1.

Automatic adjustment is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register every time the adjustment period selected by the RCR2.AADJP bit elapses.

(1) Example 1: Sub-clock oscillator running at 32.769 kHz

Adjustment procedure

When the sub-clock oscillator is running at 32.769 kHz, 1 second elapses every 32769 clock cycles. The RTC is meant to run at 32768 clock cycles, so the clock runs fast by 1 clock cycle every second. The time on the clock is fast by 60 clock cycles per minute, so adjustment can take the form of setting the clock back by 60 cycles every minute.

Register settings when RCR2.CNTMD = 0:

- RCR2.AADJP = 0 (adjustment every minute)
- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler)
- RADJ.ADJ[5:0] = 60 (0x3C)

(2) Example 2: Sub-clock oscillator running at 32.766 kHz

Adjustment procedure

When the sub-clock oscillator is running at 32.766 kHz, 1 second elapses every 32766 clock cycles. The RTC is meant to run at 32768 clock cycles, so the clock runs slow by 2 clock cycles every second. The time on the clock is slow by 20 clock cycles every 10 seconds, so adjustment can take the form of setting the clock forward by 20 cycles every 10 seconds.

Register settings when RCR2.CNTMD = 0:

- RCR2.AADJP = 1 (adjustment every 10 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler)
- RADJ.ADJ[5:0] = 20 (0x14)

(3) Example 3: Sub-clock oscillator running at 32.764 kHz

Adjustment procedure

When the sub-clock oscillator is running at 32.764 kHz, 1 second elapses on 32764 clock cycles. Because the RTC operates for 32768 clock cycles as 1 second, the clock is delayed for 4 clock cycles per second. In 8 seconds, the delay is 32 clock cycles, therefore correction can be made by advancing the clock 32 clock cycles every 8 seconds.

Register settings when RCR2.CNTMD = 1:

- RCR2.AADJP = 1 (adjustment every 8 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler)
- RADJ.ADJ[5:0] = 32 (0x20)

24.3.8.2 Adjustment by software

Enable adjustment by software by setting the RCR2.AADJE bit to 0. Adjustment by software is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register on execution of a write instruction to the RADJ register.

(1) Example 1: Sub-clock oscillator running at 32.769 kHz

Adjustment procedure

When the sub-clock oscillator is running at 32.769 kHz, 1 second elapses every 32769 clock cycles. The RTC is meant to run at 32768 clock cycles, so the clock runs fast by 1 clock cycle every second. The time on the clock is fast by 1 clock cycle per second, so adjustment can take the form of setting the clock back by 1 cycle every second.

Register settings

- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler)
- RADJ.ADJ[5:0] = 1 (0x01)
  This is written to the RADJ register once per 1-second interrupt.

24.3.8.3 Procedure to change the mode of adjustment

When changing the mode of adjustment, change the value of the AADJE bit in RCR2 after setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

To change adjustment by software to automatic adjustment:

1. Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
2. Set the RCR2.AADJE bit to 1 (automatic adjustment is enabled).
3. Use the RCR2.AADJP bit to select the period of adjustment.
4. In RADJ, set the PMADJ[1:0] bits for addition or subtraction and the ADJ[5:0] bits to the value for use in time error adjustment.

To change automatic adjustment to adjustment by software:

1. Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
2. Set the RCR2.AADJE bit to 0 (adjustment by software is enabled).
3. Proceed with the adjustment by setting the RADJ.PMADJ[1:0] bits for addition or subtraction and the RADJ.ADJ[5:0] bits to the value for use in time error adjustment at the wanted time. After that, the time is adjusted every time a value is written to the RADJ register.

24.3.8.4 Procedure to stop adjustment

Stop the adjustment by setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

24.3.9 Capturing the time

The RTC is capable of storing the month, date, hour, minute and second/binary counters 3 to 0 by detecting an edge of a signal on a time capture event input pin in calendar count mode or binary count mode.

A noise filter can also be used on a time capture event input pin. If the noise filter is enabled, the RTCCRn.TCST bit is set to 1 when the input level on the pin matches three times.

The noise filter can be switched on or off for each of the time capture event input pins. Set VBTICTLR.VCHnIEN (n = 0 to 2) to 1 to enable the RTCICn input. Operation when the noise filter is off is shown in Figure 24.9 and operation when the noise filter is on is shown in Figure 24.10.

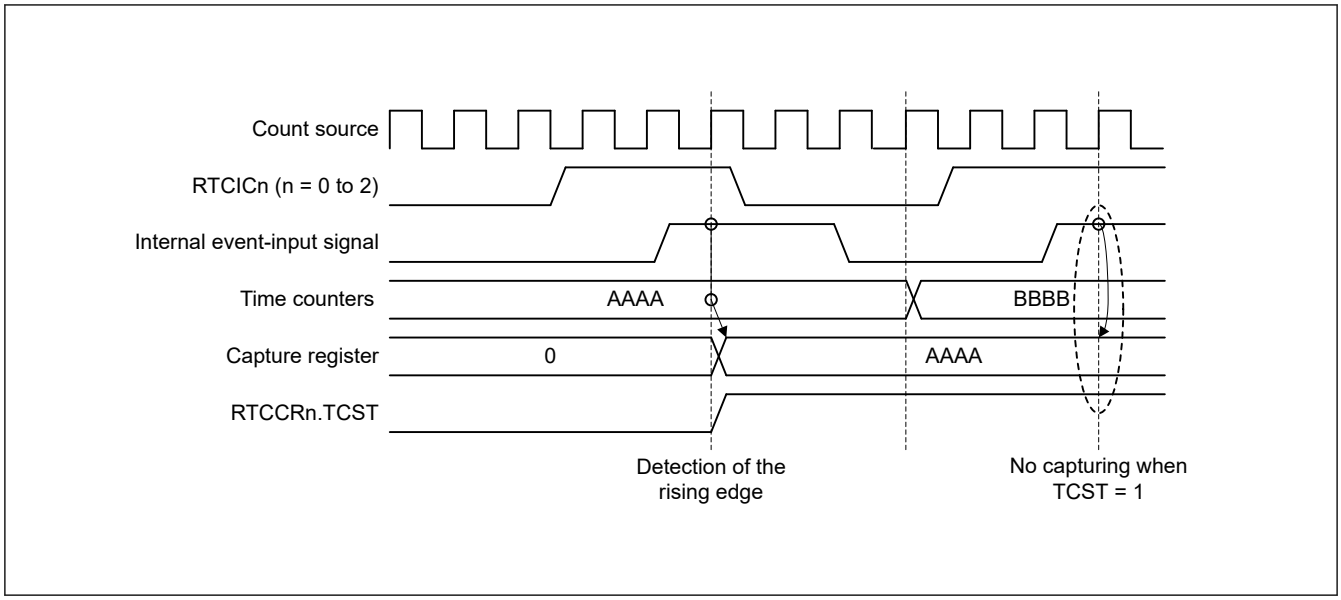


Figure 24.9 Timing of a time capture operation with the noise filter off

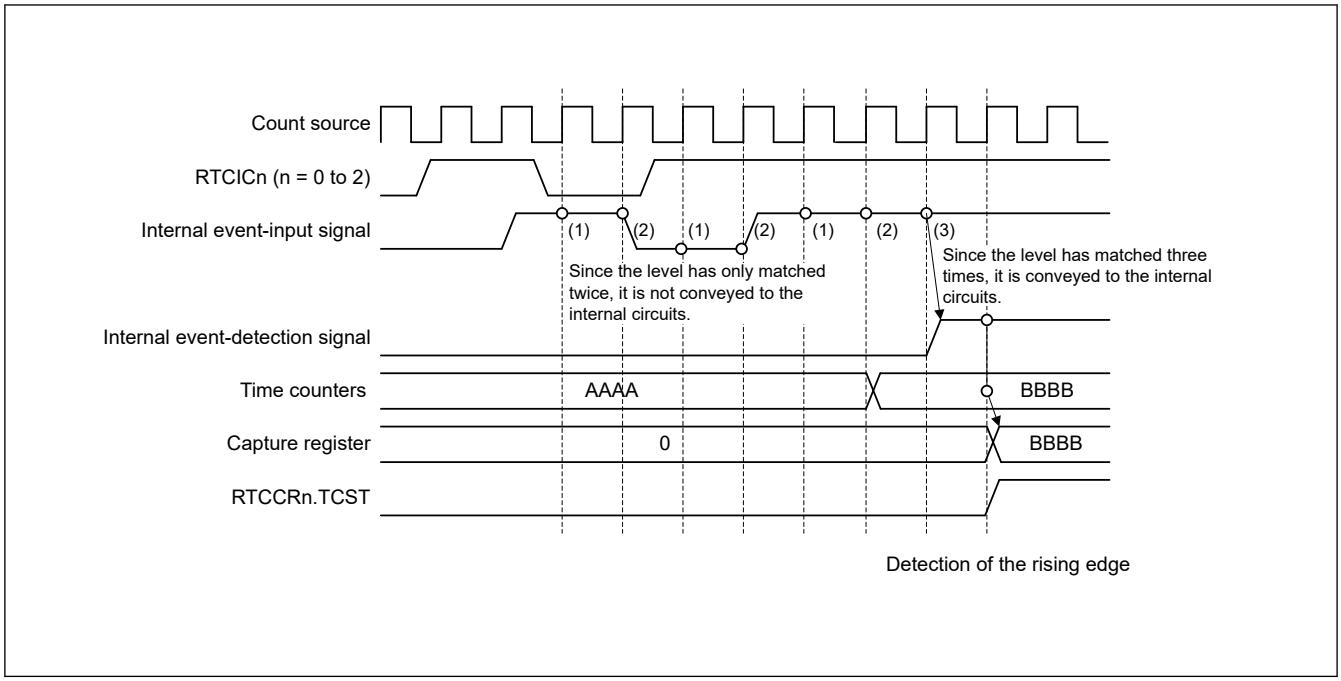


Figure 24.10 Timing of a time capture operation with the noise filter on

24.4 Interrupt Sources

The RTC has three interrupt sources, as listed in Table 24.3.

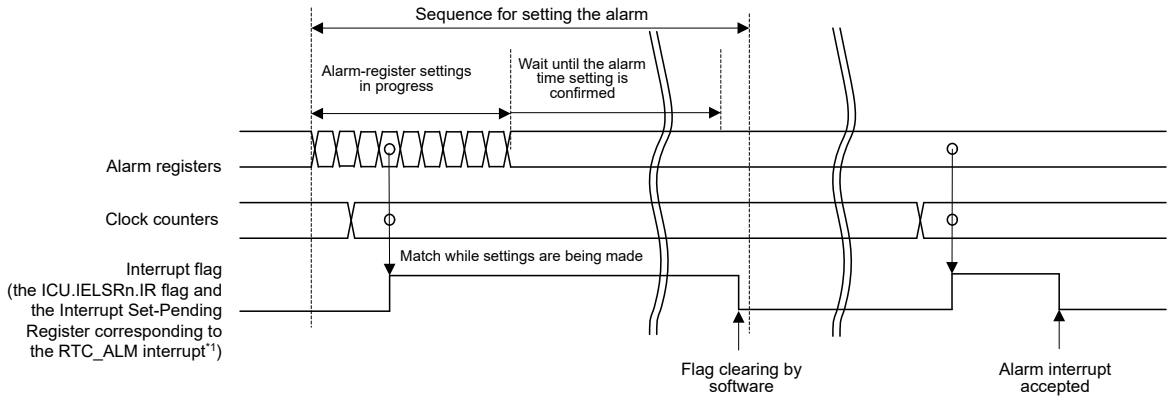
Table 24.3 RTC interrupt sources

Name	Interrupt source
RTC_ALM	Alarm interrupt
RTC_PRD	Periodic interrupt
RTC_CUP	Carry interrupt

(1) Alarm interrupt (RTC_ALM)

This interrupt is generated based on the comparison result between the alarm registers and RTC counters. For details, see section 24.3.6. Alarm Function.

Because there is a possibility that the interrupt flag might be set to 1 when the settings of the alarm registers match the clock counters, wait for the alarm time settings to be confirmed and clear the IELSRn.IR flag and the interrupt Set-Pending Register associated with the RTC_ALM interrupt to 0 again after modifying values of the alarm registers. After the interrupt flag for the alarm interrupt is set to 1 and the state is returned to mismatching of the alarm registers and clock counters, the flag is not 1 again until there is another match or the values of the alarm registers are modified again.



Note 1. See section 13, Interrupt Controller Unit (ICU) for details on the associated interrupt vector number.

Figure 24.11 Timing for the alarm interrupt (RTC_ALM)

(2) Periodic interrupt (RTC_PRD)

This interrupt is generated at intervals of 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second. The interrupt interval can be selected in the RCR1.PES[3:0] bits.

(3) Carry interrupt (RTC_CUP)

This interrupt is generated when a carry to the second counter/binary counter 0 occurred or a carry to the R64CNT counter occurred during read access to the 64-Hz counter.

Figure 24.12 shows the timing of the carry interrupt (RTC_CUP).

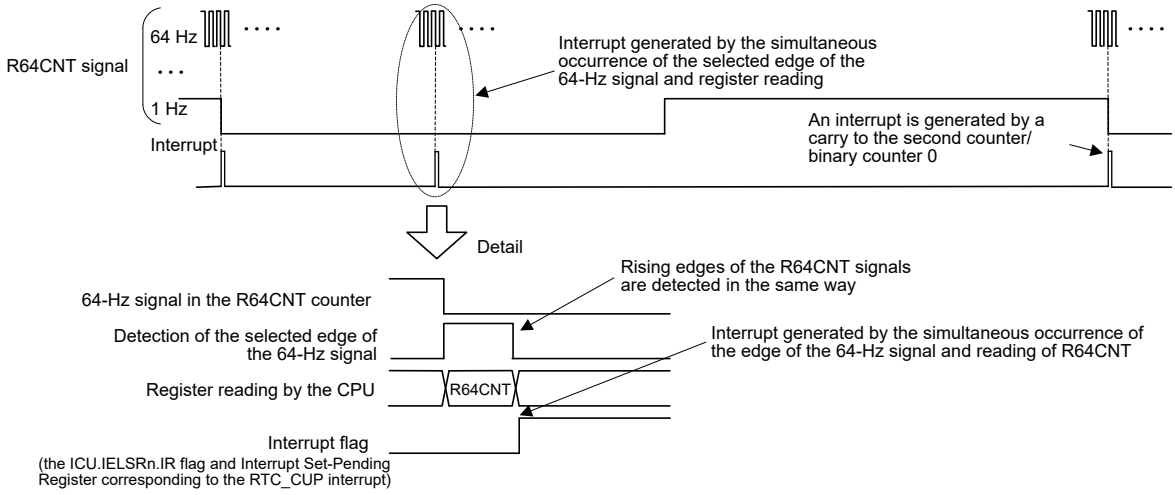


Figure 24.12 Timing for the carry interrupt (RTC_CUP)

24.5 Event Link Output

The RTC generates periodic event output (RTC_PRD) event signal for the ELC that can be used to initiate operations by other modules selected in advance.

The periodic event signal is output at the interval selected from 1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, and 2 seconds by setting the RCR1.PES[3:0] bits.

The event generation period immediately after the event generation is selected is not guaranteed.

Note: If event linking from the RTC is used, only set the ELC after setting the RTC, for example initialization and time settings. Setting the RTC after the ELC can lead to output of unexpected event signals.

24.5.1 Interrupt Handling and Event Linking

The RTC has a bit to enable or disable periodic interrupts. An interrupt request signal is output to the CPU when an interrupt source is generated while the associated enable bit is enabled.

In contrast, an event link output signal is sent to other modules as an event signal through the ELC when an interrupt source is generated, regardless of the setting of the associated interrupt enable bit.

Note: Although alarm and periodic interrupts can still be output during Software Standby or Deep Software Standby mode, the periodic event signals for the ELC are not output.

24.6 Usage Notes

24.6.1 Register Writing during Counting

The following registers should not be written to during counting, that is, while the RCR2.START bit is 1:

- RSECCNT/BCNT0
- RMINCNT/BCNT1
- RHRCNT/BCNT2
- RDAYCNT
- RWKCNT/BCNT3
- RMONCNT
- RYRCNT
- RCR1.RTCOS
- RCR2.RTCOE
- RCR2.HR24
- RFRL

The counter should be stopped before writing to any of these registers.

24.6.2 Use of Periodic Interrupts

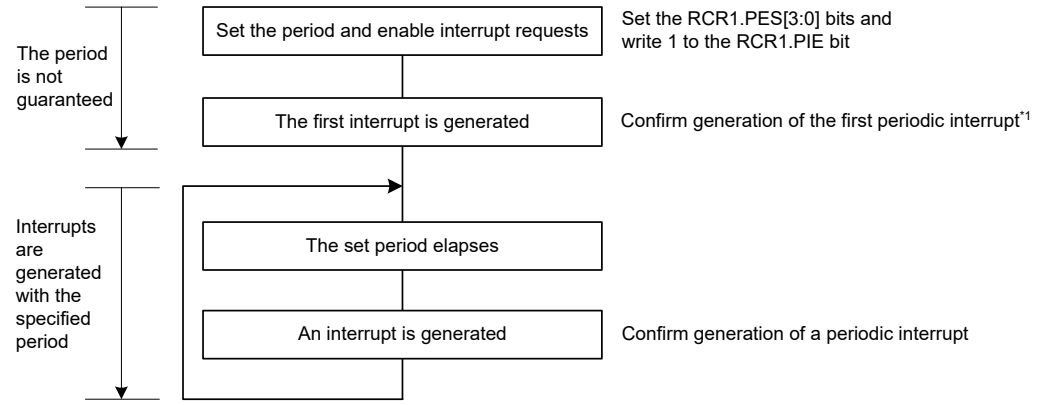
Figure 24.13 shows the procedure for using periodic interrupts.

The generation and period of the periodic interrupt can be changed by setting the RCR1.PES[3:0] bits. However, because the prescaler R64CNT and RSECCNT/BCNT0 are used to generate interrupts, the interrupt period is not guaranteed immediately after setting the RCR1.PES[3:0] bits.

In addition, any of the following operation can affect the interrupt period:

- Stopping/restarting or resetting counter operation
- Reset by RTC software
- 30-second adjustment by changing the RCR2 value

When the time error adjustment function is used, the interrupt generation period after adjustment is added or subtracted based on the adjustment value.



Note 1. When an interrupt generation period changes while the periodic interrupt is used, an interrupt might be generated at the completion of the setting. If the interrupt is generated immediately after the setting, the period is not guaranteed for two interrupts including the current interrupt.

Figure 24.13 Using the periodic interrupt function

24.6.3 RTCOUT (1-Hz/64-Hz) Clock Output

Stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the period of RTCOUT (1-Hz/64-Hz) output. When the time error adjustment function is used, the period of RTCOUT (1-Hz/64-Hz) output after adjustment is added or subtracted based on the adjustment value.

24.6.4 Transitions to Low Power Modes after Setting Registers

A transition to a low power state (Software Standby mode, Deep Software Standby mode, or battery backup state) during a write to an RTC register might corrupt the value of the register. After setting the register, confirm that the setting is in place before initiating a transition to a low power state.

24.6.5 Notes on Writing to and Reading from Registers

- When reading a counter register such as the second counter after writing to the counter register, follow the procedure in section 24.3.5. Reading 64-Hz Counter and Time.
- The value written to the count registers, alarm registers, year alarm enable register, bits RCR2.AADJE, AADJP, and HR24, RCR4 register, or frequency register is reflected when fourth read operations are performed after writing.
- The values written to the RCR1.CIE, RCR1.RTCOS, and RCR2.RTCOE bits can be read immediately after writing.
- To read the value from the timer counter after returning from a reset or a period in Software Standby mode, Deep Software Standby mode, or battery backup state, wait for 1/128 second while the clock is operating (RCR2.START bit = 1).
- After a reset is generated, write to the RTC register after 6 cycles of the count source clock have elapsed.

24.6.6 Changing the Count Mode

When changing the count mode (calendar count mode/binary count mode), set the RCR2.START bit to 0, stop the counting operation, then start it again from the initial setting. For details on the initial setting, see section 24.3.1. Outline of Initial Settings of Registers after Power On.

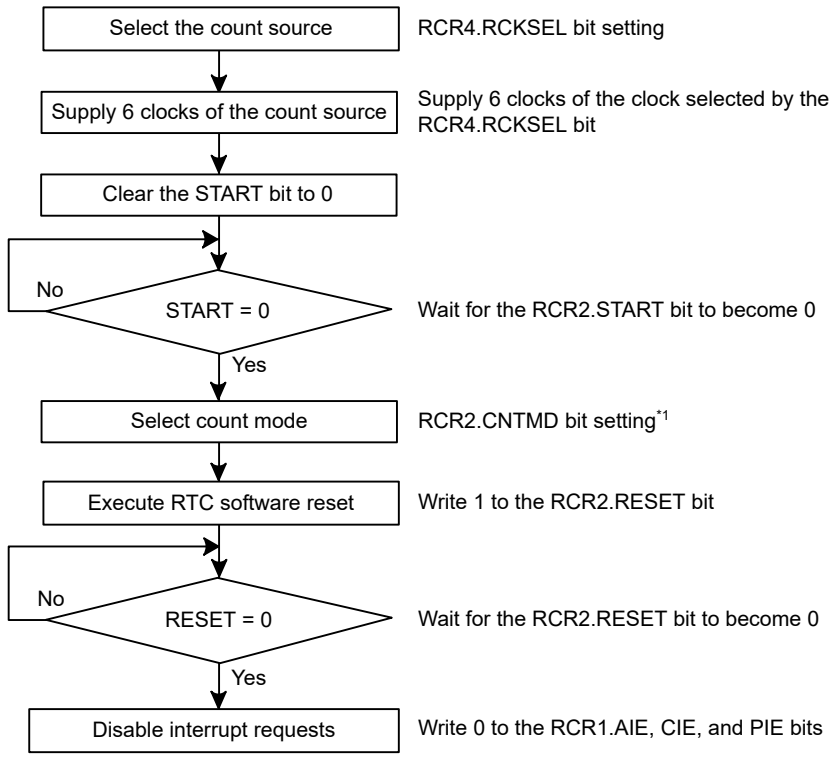
24.6.7 Initialization Procedure When the RTC Is Not to Be Used

Registers in the RTC are not initialized by a reset. Depending on the initial state, the generation of an unintentional interrupt request or operation of the counter might lead to increased power consumption.

For applications that do not require a realtime clock, initialize the registers by following the initialization procedure shown in Figure 24.14.

Alternatively, when the sub-clock oscillator is not used as the system clock source, realtime clock, operating clock of each peripheral modules or FLL function of HOCO, the counter can be stopped by writing 0 (sub-clock oscillator is selected) to the RCR4.RCKSEL bit and stopping the sub-clock oscillator. To stop the sub-clock oscillator, write 1 to the SOSCCR.SOSTP bit.

For details on the setting of the SOSCCR.SOSTP bit, see section 8, Clock Generation Circuit.



Note 1. This step is not necessary if the count mode is concurrently set by setting the START bit to 0.

Figure 24.14 Initialization procedure

24.6.8 When Switching Source Clock

When switching a clock source by changing SCKSCR.CKSEL[2:0], the clock output from the selector stops for 4 cycles of the switched clock. If the RTC periodical interrupt or RTC periodical event output was generated at this time, the interrupt or event is invalid.

24.6.9 When writing to the RTC registers

If VCC voltage of the MCU is lower than 1.8V and write access to the RTC registers is performed, read access should be performed at least once after one write access is performed, or for the consecutive write access, leave an interval of 167ns for each access.

25. Watchdog Timer (WDT)

25.1 Overview

The Watchdog Timer (WDT) is a 14-bit down counter that can be used to reset the MCU when the counter underflows because the system has run out of control and is unable to refresh the WDT. In addition, the WDT can be used to generate a non-maskable interrupt or an underflow interrupt.

Table 25.1 lists the WDT specifications and Figure 25.1 shows a block diagram.

Table 25.1 WDT specifications

Parameter	Specifications
Count source*1	Peripheral clock (PCLKB)
Clock division ratio	Division by 4, 64, 128, 512, 2048, or 8192
Counter operation	Counting down using a 14-bit down-counter
Condition for starting the counter	• Auto start mode: Counting automatically starts after a reset or after an underflow or refresh error occurs • Register start mode: Counting is started with a refresh by writing to the WDTRR register • Only secure developer can select Auto-start mode or Register-start mode
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • A counter underflows or a refresh error is generated
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Watchdog timer reset sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt/interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading of the counter value	The down-counter value can be read by the WDTSR register
Event link function (output)	• Down-counter underflow event output • Refresh error event output
Output signal (internal signal)	• Reset output • Interrupt request output • CPU Sleep mode or CPU Deep Sleep mode count stop control output
TrustZone Filter	Security and Privilege attribution can be set

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ 4 × (the frequency of the count clock source after division).

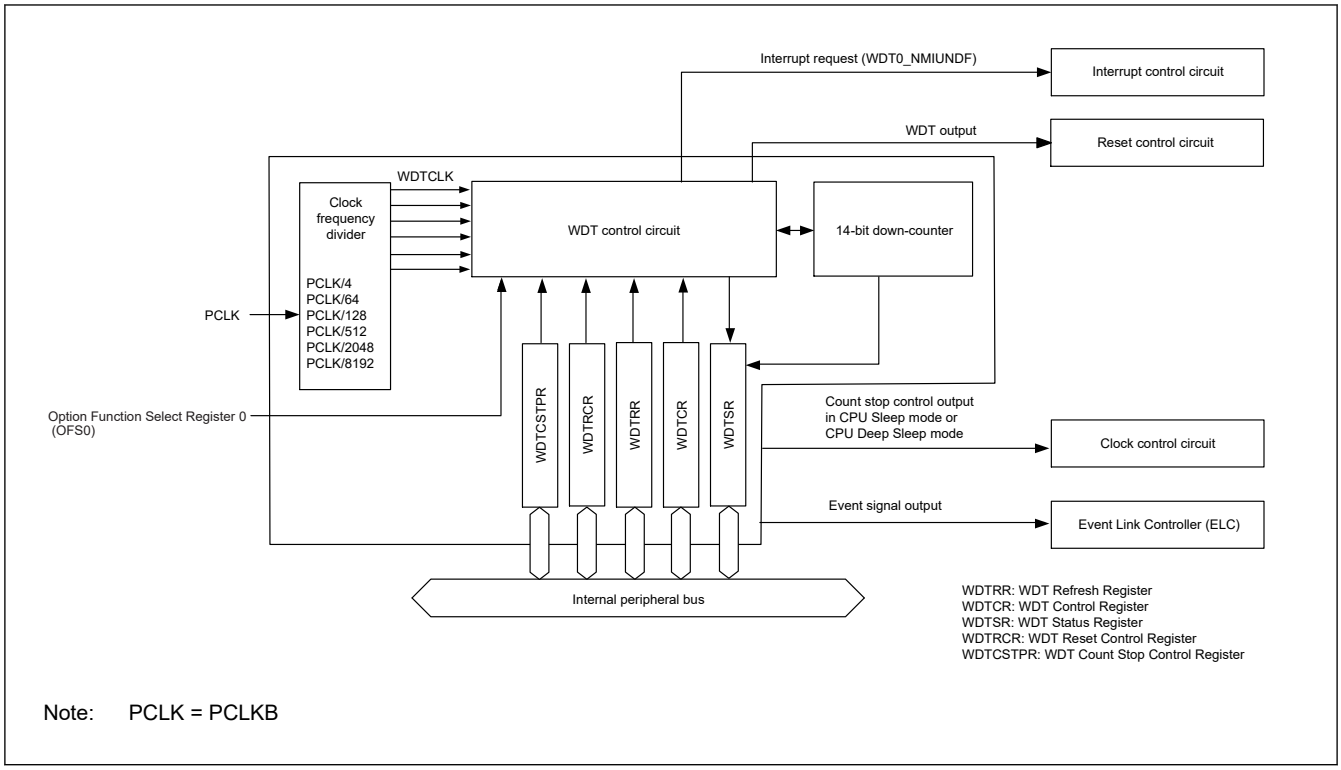


Figure 25.1 WDT block diagram

25.2 Register Descriptions

25.2.1 WDTRR : WDT Refresh Register

Base address: WDT0 = 0x4020_2600
WDT0_NS = 0x5020_2600

Offset address: 0x00

Bit position:	7							0
Bit field:								
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	n/a	The down-counter is refreshed by writing 0x00 and then writing 0xFF to this register.	R/W

Note: S-TYPE-3, P-TYPE-3

The WDTRR register refreshes the down-counter of the WDT.

The down-counter of the WDT is refreshed by writing 0x00 and then writing 0xFF to WDTRR register (refresh operation) within the refresh-permitted period.

After the down-counter is refreshed, it starts counting down from the value selected by setting the WDT Timeout Period Select bits (OFS0.WDTTOPS[1:0]) in the Option Function Select Register 0 in auto start mode. In register start mode, counting down starts from the value selected by setting the Timeout Period Select bits (WDTCR.TOPS[1:0]) in the WDT Control Register.

When 0x00 is written, the read value is 0x00. When a value other than 0x00 is written, the read value is 0xFF. For details of the refresh operation, see section 25.3.3. Refresh Operation.

25.2.2 WDTCR : WDT Control Register

Base address: WDT0 = 0x4020_2600
WDT0_NS = 0x5020_2600

Offset address: 0x02

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	RPSS[1:0]		—	—	RPES[1:0]		CKS[3:0]				—	—	TOPS[1:0]	
Value after reset:	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

Bit	Symbol	Function	R/W
1:0	TOPS[1:0]	Timeout Period Select 0 0: 1024 cycles (0x03FF) 0 1: 4096 cycles (0x0FFF) 1 0: 8192 cycles (0x1FFF) 1 1: 16384 cycles (0x3FFF)	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	CKS[3:0]	Clock Division Ratio Select 0 0 0 1: PCLKB/4 0 1 0 0: PCLKB/64 1 1 1 1: PCLKB/128 0 1 1 0: PCLKB/512 0 1 1 1: PCLKB/2048 1 0 0 0: PCLKB/8192 Others: Setting prohibited	R/W
9:8	RPES[1:0]	Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (do not specify window end position).	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	RPSS[1:0]	Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (do not specify window start position).	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The WDTCR register is used to set the clock division ratio, and window start and end positions for refresh, and the timeout period until the down-counter underflows in register start mode.

Some constraints apply to writes to the WDTCR register. For details, see section 25.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the settings in the WDTCR register are disabled, and the settings in the Option Function Select Register 0 (OFS0) are enabled. The settings for the WDTCR register can also be made in the OFS0 register. For details, see section 25.3.8. Association between Option Function Select Register 0 (OFS0) and WDT Registers.

TOPS[1:0] bits (Timeout Period Select)

The TOPS[1:0] bits select the timeout period, the period until the down-counter underflows, from 1024, 4096, 8192, and 16384 cycles, taking the divided clock specified in the CKS[3:0] bits as 1 cycle. After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the number of PCLKB cycles until the counter underflows.

Table 25.2 lists the relationship between the CKS[3:0] and TOPS[1:0] bit settings, the timeout period, and the number of PCLKB cycles.

Table 25.2 Timeout period settings

CKS[3:0] bits	TOPS[1:0] bits	Clock division ratio	Timeout period (number of cycles)	PCLKB clock cycles
0x1	00b	PCLKB/4	1024	4096
	01b		4096	16384
	10b		8192	32768
	11b		16384	65536
0x4	00b	PCLKB/64	1024	65536
	01b		4096	262144
	10b		8192	524288
	11b		16384	1048576
0xF	00b	PCLKB/128	1024	131072
	01b		4096	524288
	10b		8192	1048576
	11b		16384	2097152
0x6	00b	PCLKB/512	1024	524288
	01b		4096	2097152
	10b		8192	4194304
	11b		16384	8388608
0x7	00b	PCLKB/2048	1024	2097152
	01b		4096	8388608
	10b		8192	16777216
	11b		16384	33554432
0x8	00b	PCLKB/8192	1024	8388608
	01b		4096	33554432
	10b		8192	67108864
	11b		16384	134217728

CKS[3:0] bits (Clock Division Ratio Select)

The CKS[3:0] bits specify the division ratio of the clock used for the down-counter. The division ratio can be selected from the PCLKB divided by 4, 64, 128, 512, 2048, and 8192. Combined with the TOPS[1:0] bit setting, this allows the WDT to be configured to a count period between 4096 and 134217728 PCLKB clock cycles.

RPES[1:0] bits (Window End Position Select)

The RPES[1:0] bits specify the window end position that indicates the refresh-permitted period. 75%, 50%, 25%, or 0% of the timeout period can be selected for the window end position. Set the window end position to a value less than the value for the window start position (window start position > window end position). If the window start position is set to a value less than or equal to the window end position, the window start position setting is enabled and the window end position is set to 0%.

RPSS[1:0] bits (Window Start Position Select)

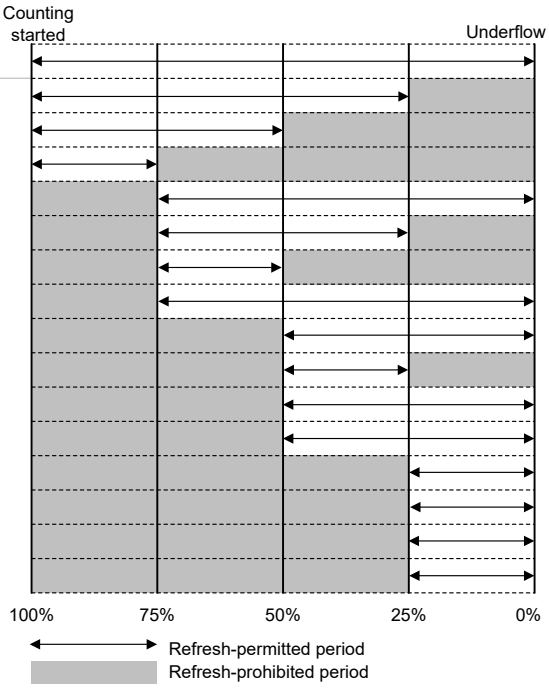
The RPSS[1:0] bits specify the window start position that indicates the refresh-permitted period. 100%, 75%, 50%, or 25% of the timeout period can be selected for the window start position. Set the window start position to a value greater than the value for the window end position. If the window start position is set to a value less than or equal to the window end position, the window start position setting is enabled and the window end position is set to 0%.

Table 25.3 lists the counter values for the window start and end positions, and Figure 25.2 shows the refresh-permitted period set in the RPSS[1:0], RPES[1:0], and TOPS[1:0] bits.

Table 25.3 Relationship between the timeout period and window start and end counter values

TOPS[1:0]	Timeout period		Window start and end counter value			
	Cycles	Counter value	100%	75%	50%	25%
00b	1024	0x03FF	0x03FF	0x02FF	0x01FF	0x00FF
01b	4096	0x0FFF	0x0FFF	0x0BFF	0x07FF	0x03FF
10b	8192	0x1FFF	0x1FFF	0x17FF	0x0FFF	0x07FF
11b	16384	0x3FFF	0x3FFF	0x2FFF	0x1FFF	0x0FFF

RPSS[1:0] bits		RPES[1:0] bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75



Note: If window end position setting ≥ window start position setting, the window end position setting is set to 0%.

Figure 25.2 RPSS[1:0] and RPES[1:0] bits setting and refresh-permitted period

25.2.3 WDTSR : WDT Status Register

Base address: WDT0 = 0x4020_2600
WDT0_NS = 0x5020_2600

Offset address: 0x04

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	REFEF	UNDF	CNTVAL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	CNTVAL[13:0]	Down-Counter Value Value counted by the down-counter	R
14	UNDFF	Underflow Flag 0: No underflow occurred 1: Underflow occurred	R/W*1

Bit	Symbol	Function	R/W
15	REFEF	Refresh Error Flag 0: No refresh error occurred 1: Refresh error occurred	R/W*1

Note: S-TYPE-3, P-TYPE-3
Note 1. Only 0 can be written to clear the flag.

The WDTSR register indicates the counter value of the down-counter and the status of whether an underflow or refresh error occurred in the down-counter.

CNTVAL[13:0] bits (Down-Counter Value)

Read the CNTVAL[13:0] bits to confirm the value of the down-counter. The read value might differ from the actual count by 1.

UNDFF flag (Underflow Flag)

Read the UNDFF flag to confirm whether an underflow occurred in the counter. A value of 1 indicates that the down counter underflowed. Write 0 to the flag to set the value to 0. Writing 1 has no effect.

Clearing of the UNDFF flag takes (N + 1) PCLKB cycles. In addition, clearing of the flag is ignored for (N + 1) PCLKB cycles after an underflow. N is specified in the WDTCR.CKS[3:0] bits as follows:

- When WDTCR.CKS[3:0] = 0x1, N = 4
- When WDTCR.CKS[3:0] = 0x4, N = 64
- When WDTCR.CKS[3:0] = 0xF, N = 128
- When WDTCR.CKS[3:0] = 0x6, N = 512
- When WDTCR.CKS[3:0] = 0x7, N = 2048
- When WDTCR.CKS[3:0] = 0x8, N = 8192

REFEF flag (Refresh Error Flag)

Read the REFEF flag to confirm whether a refresh error occurred, indicating that a refresh operation was performed during a prohibited period. A value of 1 indicates that a refresh error occurred. Write 0 to the flag to set the value to 0. Writing 1 has no effect.

Clearing of the REFEF flag takes (N + 1) PCLKB cycles. In addition, clearing of the flag is ignored for (N + 1) PCLKB cycles after a refresh error. N is specified in the WDTCR.CKS[3:0] bits as follows:

- When WDTCR.CKS[3:0] = 0x1, N = 4
- When WDTCR.CKS[3:0] = 0x4, N = 64
- When WDTCR.CKS[3:0] = 0xF, N = 128
- When WDTCR.CKS[3:0] = 0x6, N = 512
- When WDTCR.CKS[3:0] = 0x7, N = 2048
- When WDTCR.CKS[3:0] = 0x8, N = 8192

25.2.4 WDTRCR : WDT Reset Control Register

Base address: WDT0 = 0x4020_2600
WDT0_NS = 0x5020_2600

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RSTIRQS	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	RSTIRQS	WDT Behavior Selection 0: Interrupt 1: Reset	R/W

Note: S-TYPE-3, P-TYPE-3

The WDTRCR register controls reset output by a WDT down-counter underflow or interrupt request output.

Some constraints apply to writes to the WDTRCR register. For details, see section 25.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the WDTRCR register settings are disabled, and the settings in the Option Function Select register 0 (OFS0) are enabled. The settings for the WDTRCR register can also be made for the OFS0 register. For details, see section 25.3.8. Association between Option Function Select Register 0 (OFS0) and WDT Registers.

25.2.5 WDTCSTPR : WDT Count Stop Control Register

Base address: WDT0 = 0x4020_2600
WDT0_NS = 0x5020_2600

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SLCSTP	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	SLCSTP	CPU Sleep-Mode Count Stop Control Register 0: Disable count stop 1: Stop count on transition to CPU Sleep mode or CPU Deep Sleep mode	R/W

Note: S-TYPE-3, P-TYPE-3

The WDTCSTPR register controls whether to stop the WDT counter in CPU Sleep mode or CPU Deep Sleep mode. Some constraints apply to writes to the WDTCSTPR register. For details, see section 25.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the WDTCSTPR register settings are disabled, and the settings in the Option Function Select register 0 (OFS0) are enabled. The settings for the WDTCSTPR register can also be made for the OFS0 register. For details, see section 25.3.8. Association between Option Function Select Register 0 (OFS0) and WDT Registers.

SLCSTP bit (CPU Sleep-Mode Count Stop Control Register)

The SLCSTP bit selects whether to stop counting on transition to CPU Sleep mode or CPU Deep Sleep mode.

25.2.6 Option Function Select Register 0 (OFS0)

For information on the OFS0 register, see section 25.3.8. Association between Option Function Select Register 0 (OFS0) and WDT Registers.

25.3 Operation

25.3.1 Count Operation in each Start Mode

The WDT has two start modes:

- Auto start mode, in which counting automatically starts after a release from the reset state
- Register start mode, in which counting starts with a refresh by writing to the register.

In auto start mode, counting automatically starts after a release from the reset state according to the settings in the Option Function Select register 0 (OFS0) in the flash.

In register start mode, counting starts with a refresh by writing to the WDTRR register after the respective registers are set after a release from the reset state.

Select auto start mode or register start mode by setting the WDT Start Mode Select bit (OFS0.WDTSTRT) in the OFS0 register.

When the auto start mode is selected, the settings in the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are disabled while the settings in the OFS0 register are enabled.

When the register start mode is selected, the setting for the OFS0 register is disabled while the settings for the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are enabled.

25.3.1.1 Register start mode

When the WDT Start Mode Select bit (OFS0.WDTSTRT) is 1, register start mode is selected, the OFS0 register setting is invalid, and the WDT control register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are enabled.

After the reset state is released, set the following:

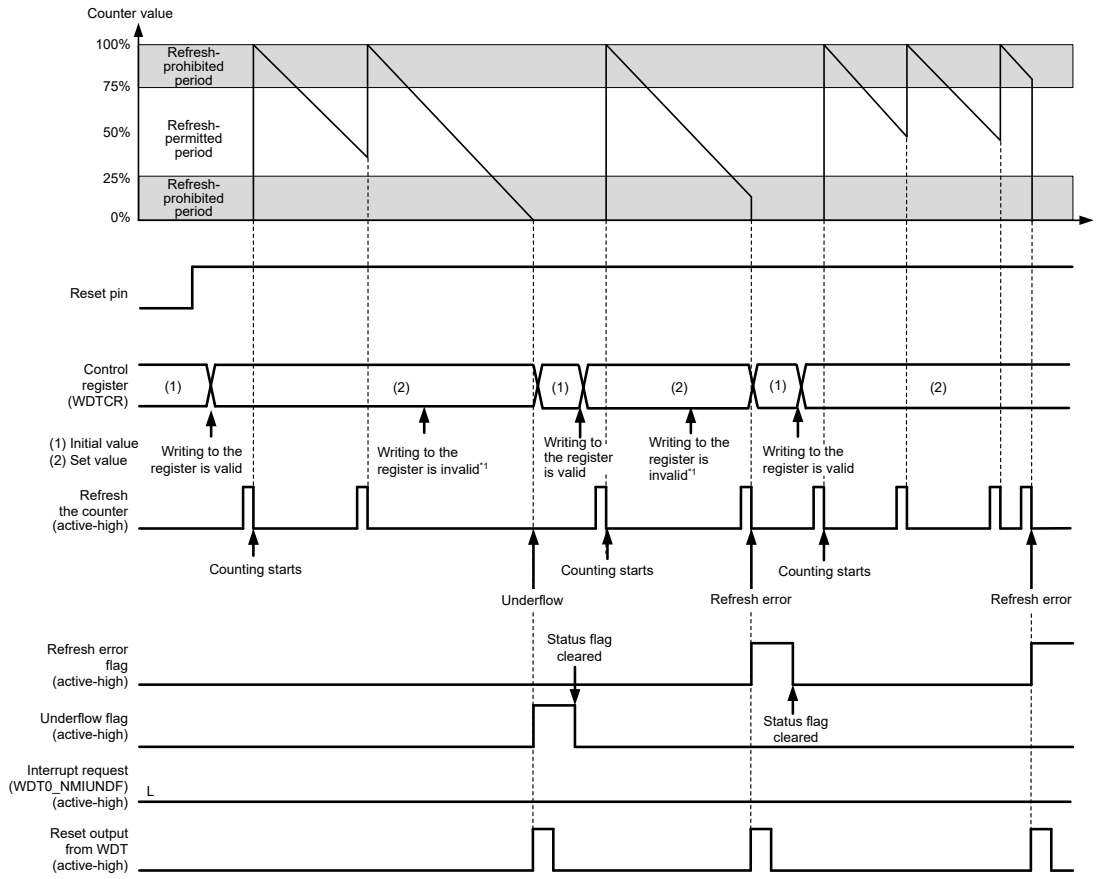
- Clock division ratio in the WDTCR register
- Window start and end positions in the WDTCR register
- Timeout period in the WDTCR register
- Reset output or interrupt request output in the WDTRCR register
- Counter stop control during transitions to CPU Sleep mode or CPU Deep Sleep mode in the WDTCSTPR register

The WDT refresh register (WDTRR) refreshes the down counter. As a result, the downcount starts at the value set by the timeout period selection bit (WDTCR.TOPS[1:0]).

Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and counting down continues. The WDT does not output the reset signal or non-maskable interrupt request/interrupt request as long as counting continues. However, if the down-counter underflows because the down-counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the WDT outputs the reset signal or a non-maskable interrupt request/interrupt request (WDT0_NMIUNDF). Reset output or interrupt request output can be selected in the WDT Reset Interrupt Request Select bit (WDTRCR.RSTIRQS). The interrupt enabled for operating the NMI can be selected in the WDT Underflow/Refresh Error Interrupt Enable bit (NMIER.WDTEN).

Figure 25.3 shows an example of operation under the following conditions:

- Register start mode (OFS0.WDTSTRT = 1)
- WDT reset interrupt request selection (WDTRCR.RSTIRQS = 1)
- The window start position is 75% (WDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (WDTCR.RPES[1:0] = 10b)



Note: Refreshing the counter requires up to 4 count signal cycles from refresh request to execution.
Note 1. See section 25.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.
Reset pin = RES

Figure 25.3 Operation example in register start mode

25.3.1.2 Auto start mode

When the WDT Start Mode Select bit (OFS0.WDTSTRT) in the Option Function Select Register 0 (OFS0) is 0, auto start mode is selected, the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are disabled, and the settings in the OFS0 register are enabled.

Within the reset state, the setting values for the following in the Option Function Select Register 0 (OFS0) are set in the WDT registers:

- Clock division ratio
- Window start and end positions
- Timeout period
- Reset output or interrupt request
- Counter stop control during transition to CPU Sleep mode or CPU Deep Sleep mode

When the reset state is released, the down-counter automatically starts counting down from the value set in the WDT Timeout Period Select bits (OFS0.WDTTOPS[1:0]).

Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and counting down continues. The WDT does not output the reset signal or non-maskable interrupt

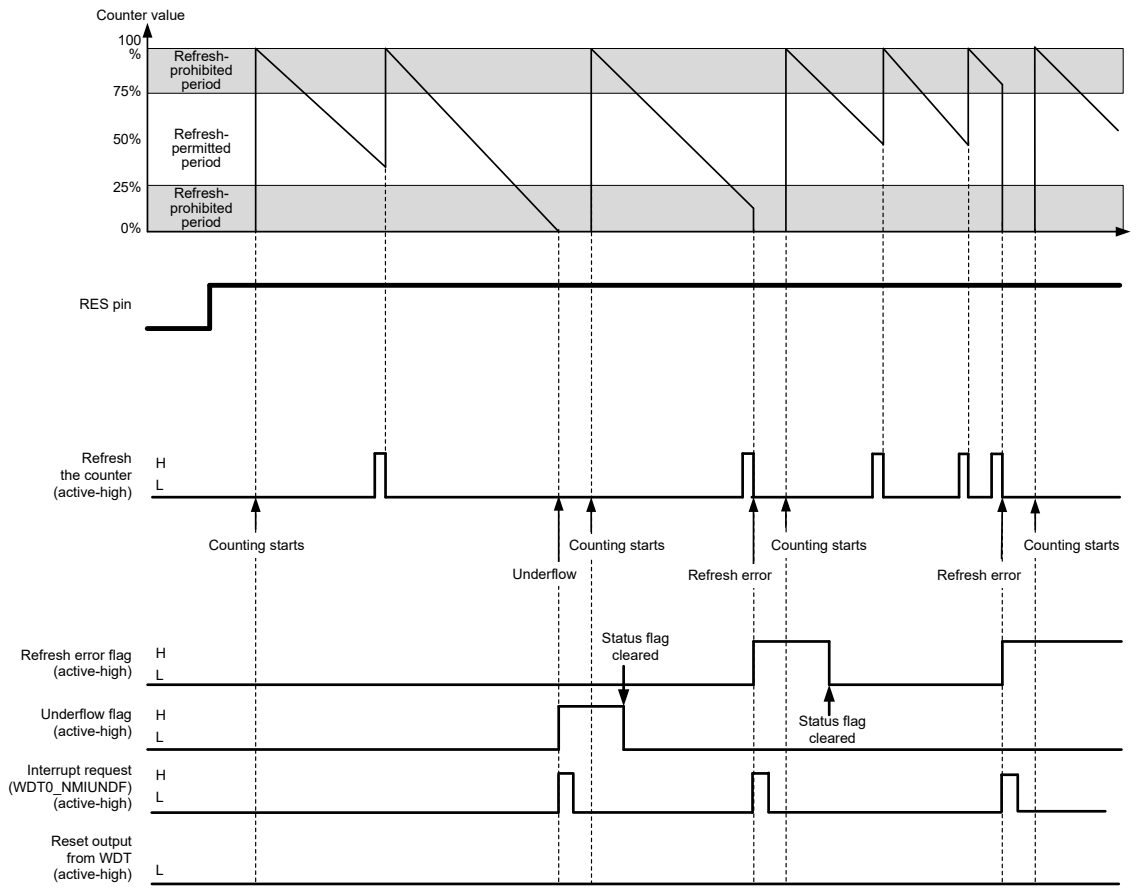
request/interrupt request (WDT0_NMIUNDF) as long as the counting continues. However, if the down-counter underflows because refreshing of the down-counter is not possible due to a runaway program or if a refresh error occurs due to refreshing outside the refresh-permitted period, the WDT outputs the reset signal or non-maskable interrupt request/interrupt request (WDT0_NMIUNDF).

After the reset signal or non-maskable interrupt request/interrupt request is generated, the counter reloads the timeout period after counting for 1 cycle. The value of the timeout period is set in the down-counter and counting restarts.

Reset output or interrupt request output can be selected by setting the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS). Non-maskable interrupt request or interrupt request can be selected in the WDT Underflow/ Refresh Error Interrupt Enable bit (NMIER.WDTEN).

Figure 25.4 shows an example of operation (non-maskable interrupt) under the following conditions:

- Auto start mode (OFS0.WDTSTRT = 0)
- WDT behavior selection: interrupt (OFS0.WDTRSTIRQS = 0)
- Non-maskable Interrupt: WDT Underflow/Refresh Error Interrupt Enabled (NMIER.WDTEN = 1)
- The window start position is 75% (OFS0.WDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.WDTRPES[1:0] = 10b)



Note: Refreshing the counter requires up to 4 count signal cycles from refresh request to execution.
Note: Reset pin = RES

Figure 25.4 Operation example in auto start mode

25.3.2 Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers

Writing to the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), or WDT Count Stop Control Register (WDTCSTPR) is possible once each between the release from the reset state and the first refresh operation.

After a refresh (counting starts) or a write to WDTCR, WDTRCR or WDTCSTPR register, the protection signal in the WDT becomes 1 to protect WDTCR, WDTRCR and WDTCSTPR register against subsequent write attempts. This protection is released by the reset source of the WDT. With other reset sources, the protection is not released.

Figure 25.5 shows control waveforms produced in response to writing to the WDTCR.

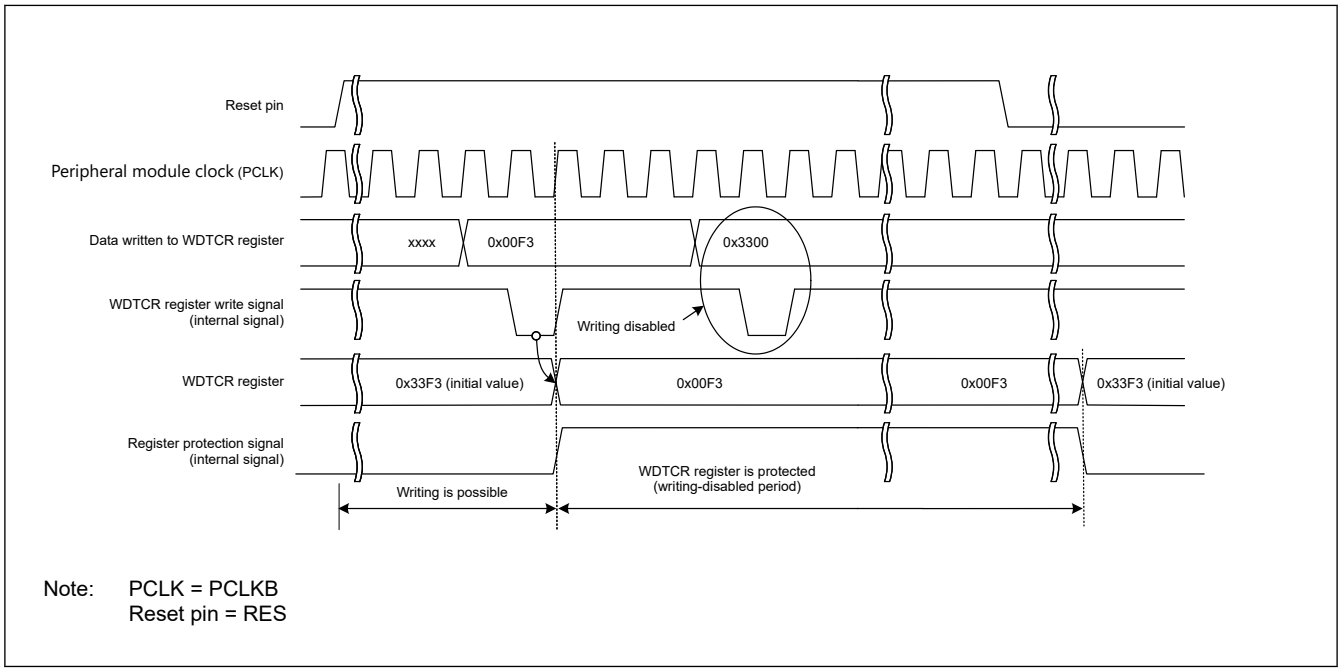


Figure 25.5 Control waveforms produced in response to writes to the WDTCR register

25.3.3 Refresh Operation

The down-counter is refreshed and starts counting operation on a write of the values 0x00 and then 0xFF to the WDT Refresh Register (WDTRR).

If a value other than 0xFF is written after 0x00, the down-counter is not refreshed.

If an invalid value is written, correct refreshing resumes on a write of 0x00 and then 0xFF to the WDTRR register.

Correct refreshing is also performed when a register other than WDTRR is accessed or WDTRR is read between writing 0x00 and writing 0xFF to WDTRR. Writes to refresh the counter must be made within the refresh-permitted period, and this is determined by the 0xFF write. For this reason, correct refreshing is performed even when 0x00 is written outside the refresh-permitted period.

[Example write sequences that are valid for refreshing the counter]

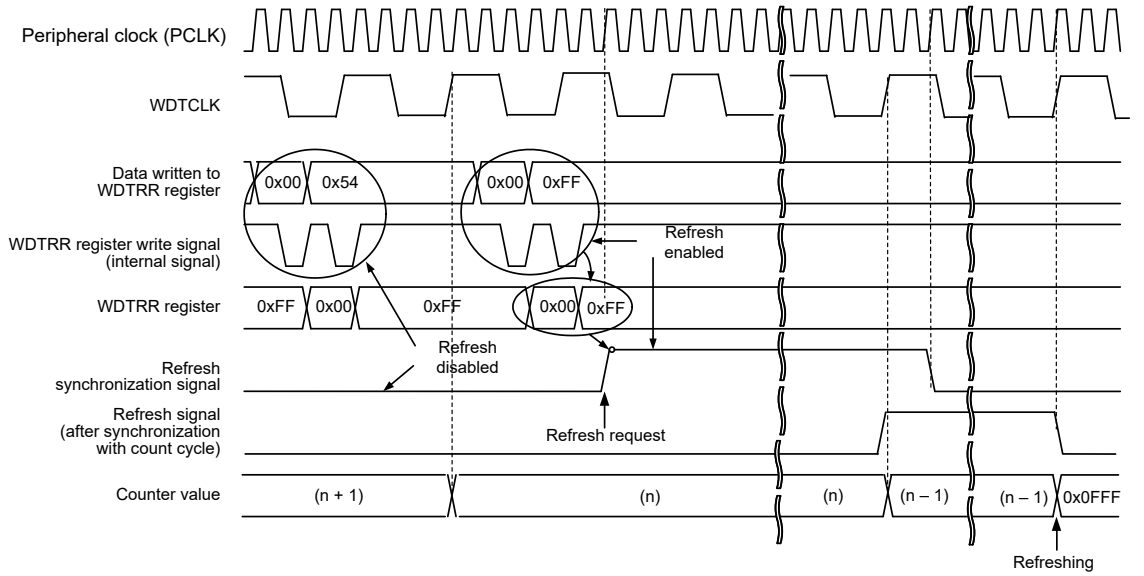
- 0x00 → 0xFF
- 0x00 ((n–1)th time) → 0x00 (nth time) → 0xFF
- 0x00 → access to another register or read from WDTRR → 0xFF

[Example write sequences that are invalid for refreshing the counter]

- 0x23 (a value other than 0x00) → 0xFF
- 0x00 → 0x54 (a value other than 0xFF)
- 0x00 → 0xAA (0x00 and a value other than 0xFF) → 0xFF

After 0xFF is written to the WDT Refresh Register (WDTRR), refreshing the down-counter requires up to 4 cycles of the signal for counting. To meet this requirement, complete writing 0xFF to WDTRR 4 count cycles before the down-counter underflows.

Figure 25.6 shows the WDT refresh-operation waveforms when the clock division ratio is PCLKB/64.



Note: PCLK = PCLKB

Figure 25.6 WDT refresh operation waveforms when WDTCR.CKS[3:0] = 0x4 and WDTCR.TOPS[1:0] = 01b

Note: When setting the refresh time, consider the oscillation accuracy of the clock sources of the PCLKB and WDTCLK. Set values which ensure that refreshing is possible even when the frequency varies in the range of error of the oscillation accuracy.

25.3.4 Status Flags

The refresh error (WDTSR.REFEF) and underflow (WDTSR.UNDFF) flags retain the source of the interrupt request from the WDT. After a release from the interrupt request generation, read the WDTSR.REFEF and WDTSR.UNDFF flags to check for the interrupt source. For each flag, writing 0 clears the bit. Writing 1 has no effect. Leaving the status flags unchanged does not affect operation. If the flags are not cleared at the next interrupt request from the WDT, the earlier interrupt source is cleared and the new interrupt source is written. For the time period between when 0 is written in each flag and when its value is reflected, see section 25.2.3. WDTSR : WDT Status Register.

25.3.5 Reset Output

When the Reset Interrupt Select bit (WDTRCR.RSTIRQS) is set to 1 in register start mode, or when the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS) in the Option Function Select Register 0 (OFS0) is set to 1 in auto start mode, a reset signal is output for 1 cycle count when an underflow in the down-counter or a refresh error occurs.

In register start mode, the down-counter is initialized (all bits set to 0) and stopped in that state after output of a reset signal. After the reset state is released and the program is restarted, the counter is set up again and counting down starts again with a refresh. In auto start mode, counting down starts automatically after the reset state is released.

25.3.6 Interrupt Sources

When the Reset Interrupt Select bit (WDTRCR.RSTIRQS) is set to 0 in register start mode or when the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS) in the Option Function Select Register 0 (OFS0) is set to 0 in auto start

mode, an interrupt (WDT0_NMIUNDF) signal is generated when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt or an interrupt. For details, see section 13, Interrupt Controller Unit (ICU).

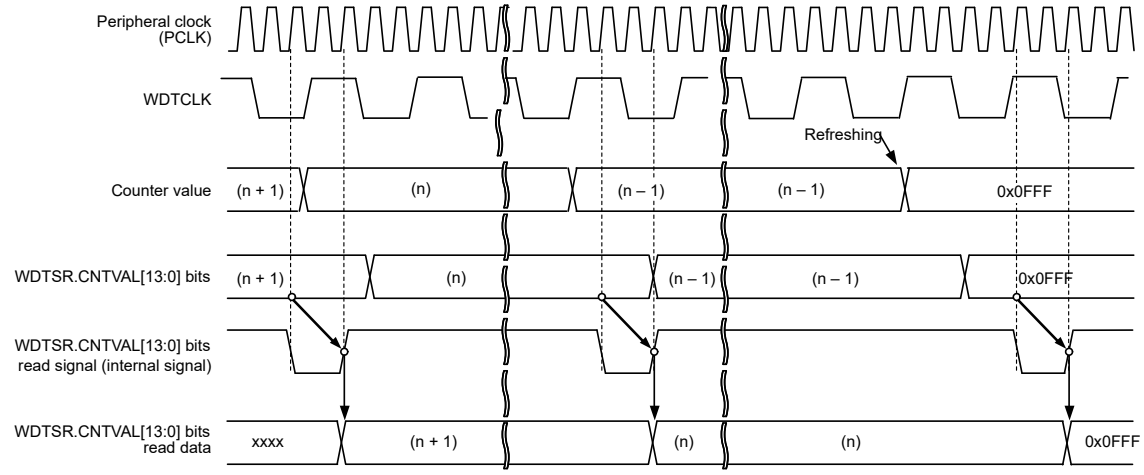
Table 25.4 WDT interrupt source

Name	Interrupt source	Interrupt to CPU	Start DMAC or DTC
WDT0_NMIUNDF	• Down-counter underflow • Refresh error	Possible	Not possible

25.3.7 Reading the Down-Counter Value

The WDT stores the counter value in the down-counter value bits (WDTSR.CNTVAL[13:0]) of the WDT Status Register. Check these bits to obtain the counter value. The read value of the down-counter might differ from the actual count by one.

Figure 25.7 shows the processing for reading the WDT down-counter value when the clock division ratio is PCLKB/64.



Note: PCLK = PCLKB

Figure 25.7 Processing for reading WDT down-counter value when WDTCR.CKS[3:0] = 0x4 and WDTCR.TOPS[1:0] = 01b

25.3.8 Association between Option Function Select Register 0 (OFS0) and WDT Registers

Table 25.5 lists the association between the Option Function Select Register 0 (OFS0) used in auto start mode, and the registers used in register start mode. For details on the Option Function Select Register 0 (OFS0), see section 6.2.1. OFS0 : Option Function Select Register 0.

Table 25.5 Association between Option Function Select Register 0 (OFS0) and the WDT registers (1 of 2)

Control target	Function	OFS0 register (enabled in auto start mode) OFS0.WDTSTRT = 0	WDT registers (enabled in register start mode) OFS0.WDTSTRT = 1
Down-counter	Timeout period selection	OFS0.WDTTOPS[1:0]	WDTCR.TOPS[1:0]
	Clock division ratio selection	OFS0.WDTCKS[3:0]	WDTCR.CKS[3:0]
	Window start position selection	OFS0.WDTRPSS[1:0]	WDTCR.RPSS[1:0]
	Window end position selection	OFS0.WDTRPES[1:0]	WDTCR.RPES[1:0]

Table 25.5 Association between Option Function Select Register 0 (OFS0) and the WDT registers (2 of 2)

Control target	Function	OFS0 register (enabled in auto start mode) OFS0.WDTSTRT = 0	WDT registers (enabled in register start mode) OFS0.WDTSTRT = 1
Reset output or interrupt request output	Select a reset interrupt request	OFS0.WDTRSTIRQS	WDTRCR.RSTIRQS
Count stop	CPU Sleep mode or CPU Deep Sleep mode count stop control	OFS0.WDTSTPCTL	WDTCSTPR.SLCSTP

25.4 Output to the Event Link Controller (ELC)

The WDT is capable of a link operation for the previously specified module when interrupt request signal is used as an event signal by the ELC. The event signal is output by the counter underflow and refresh error. An event signal is output regardless of the setting of the Reset Interrupt Request Select bit (WDTRCR.RSTIRQS) in register start mode or auto start mode. An event signal can also be output when the next interrupt source is generated while the Refresh Error flag (WDTSR.REFEF) or Underflow flag (WDTSR.UNDFF) is 1. For details, see section 18, Event Link Controller (ELC).

25.5 Usage Notes

25.5.1 ICU Event Link Setting Register n (IELSRn) Setting

Setting 0x53 to ICU Event Link Setting Register n (ICU.IELSRn) is prohibited when WDT reset interrupt request selection resets (OFS0.WDTRSTIRQS = 0 or WDTRCR.RSTIRQS = 0) or when enabling event link operation (ELSRn.ELS[8:0] = 0x53).

26. Independent Watchdog Timer (IWDT)

26.1 Overview

The Independent Watchdog Timer (IWDT) consists of a 14-bit down counter that must be serviced periodically to prevent counter underflow. The IWDT provides functionality to reset the MCU or to generate a non-maskable interrupt or an underflow interrupt. Because the timer operates with an independent clock from CPU clock (CPUCLK) and System clock (ICLK), it is particularly useful in returning the MCU to a known state as a fail-safe mechanism when the system runs out of control. The IWDT can be triggered automatically by a reset, underflow, refresh error, or a refresh of the count value in the registers.

The IWDT functions differ from those of the WDT in the following respects:

The divided IWDT clock (IWDTCLK) is used as the count source (not affected by PCLKB)

Table 26.1 lists the IWDT specifications.

Table 26.1 IWDT specifications

Parameter	Specifications
Count source*1	IWDT clock (IWDTCLK)
Clock division ratio	Division by 1, 16, 32, 64, 128, or 256
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	• Auto start mode: Counting automatically starts after a reset or after an underflow or refresh error occurs. • Register start mode: Counting is started with a refresh by writing to the IWDTRR register. • Only secure developer can select Auto-start mode or Register-start mode.
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values). • A counter underflows or a refresh error is generated.
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods).
Independent watchdog timer reset sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt/interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading of the counter value	The down-counter value can be read by the IWDTSR register.
Event link function (output)	• Down-counter underflow event output • Refresh error event output
Output signal (internal signal)	• Reset output • Interrupt request output • CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode and Deep Software Standby mode count stop control output.
Trust Zone Filter	Security and Privilege attribution can be set.

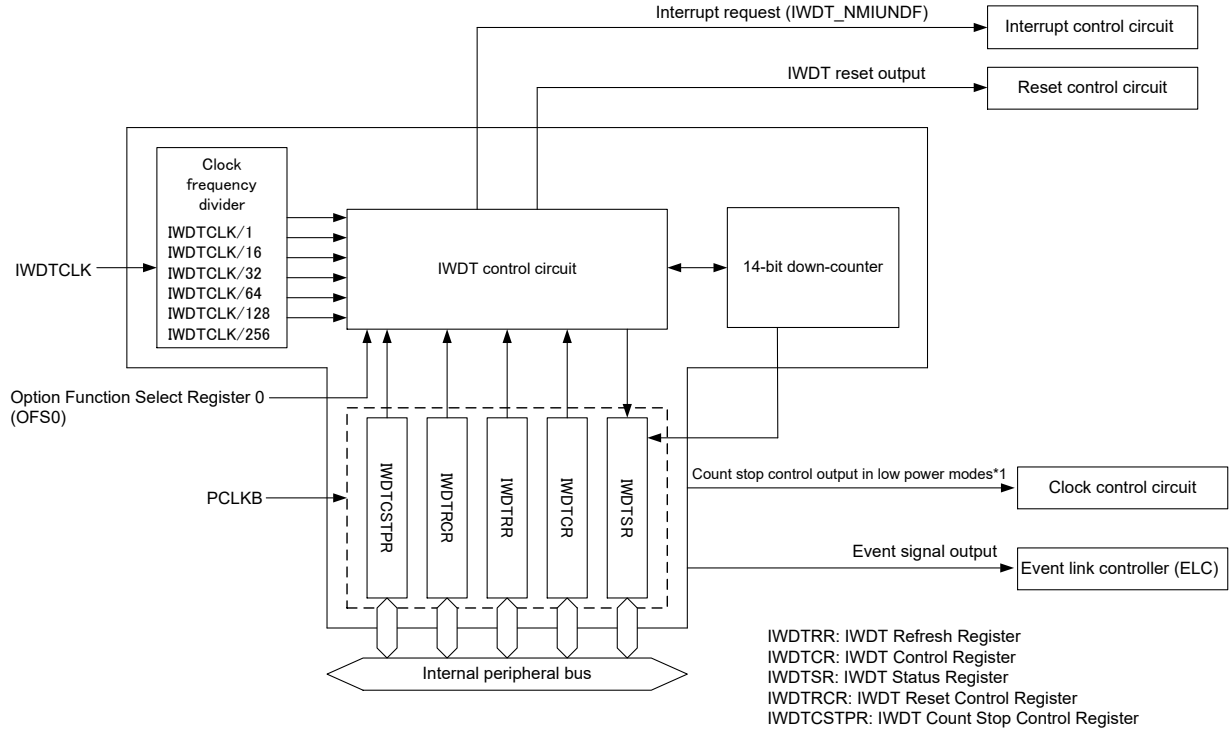
Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≧ 4 × (the frequency of the count clock source after division). The bus interface and registers operate with PCLKB, and the 14-bit counter and control circuits operate with IWDTCLK.

26.1.1 Block Diagram

In addition to the peripheral clock (PCLKB), an MCU requires the IWDT clock (IWDTCLK) that does not stop even in low power modes so that the MCU runs even in low power modes in which the peripheral clock (PCLKB) stops. The bus interface and registers operate with the peripheral clock (PCLKB), and the 14-bit down-counter and control circuits operate with the IWDT clock (IWDTCLK).

Signals are connected between the block operating with the peripheral clock (PCLKB) and the block operating with the IWDT clock (IWDTCLK) via a synchronization circuit.

Figure 26.1 shows a block diagram.



Note 1. Low power modes indicate CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode, Deep Software Standby mode 1

Figure 26.1 IWDT block diagram

26.2 Register specifications

26.2.1 IWDTRR : IWDT Refresh Register

Base address: IWDT = 0x4020_2200
IWDT_NS = 0x5020_2200

Offset address: 0x0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	REFRESH[7:0]							
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	REFRESH[7:0]	Refresh Register The counter is refreshed by writing 0x00 and then writing 0xFF to this register.	R/W

Note: S-TYPE-3, P-TYPE-3

The IWDTRR register refreshes the down-counter of the IWDT.

REFRESH[7:0] bits (Refresh Register)

The down-counter of the IWDT is refreshed by writing 0x00 and then writing 0xFF to IWDTRR(refresh operation) within the refresh-permitted period. After the down-counter is refreshed, it starts counting down from the value selected by the IWDT time-out period selection bits (OFS0.IWDTTOPS[1:0]) in the auto start mode.

In the register start mode, counting down starts from the value set by the TOPS[1:0] bits in the IWDT Control Register (IWDTCR).

Also in the register start mode, counting down starts from the value set by the IWDTCR.TOPS[1:0] bits by the first refresh operation after the release from the reset state.

When 0x00 is written, the read value is always 0x00. When a value other than 0x00 is written, the read value is always 0xFF.

For details of the refresh operation, see section 26.3.3. Refresh operation.

26.2.2 IWDTCR : IWDT Control Register

Base address: IWDT = 0x4020_2200
IWDT_NS = 0x5020_2200

Offset address: 0x2

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	RPSS[1:0]		—	—	RPES[1:0]		CKS[3:0]				—	—	TOPS[1:0]	
Value after reset:	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

Bit	Symbol	Function	R/W
1:0	TOPS[1:0]	Timeout Period Select 0 0: 128 cycles (0x007F) 0 1: 512 cycles (0x01FF) 1 0: 1024 cycles (0x03FF) 1 1: 2048 cycles (0x07FF) Each value in parentheses () denotes a start value of down-counting.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	CKS[3:0]	Clock Division Ratio Select 0 0 0 0: IWDTCLK 0 0 1 0: IWDTCLK/16 0 0 1 1: IWDTCLK/32 0 1 0 0: IWDTCLK/64 1 1 1 1: IWDTCLK/128 0 1 0 1: IWDTCLK/256 Other settings are prohibited.	R/W
9:8	RPES[1:0]	Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (No window end position setting)	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	RPSS[1:0]	Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (No window start position setting)	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The IWDTCR register is used to specify the timeout period (time until the down-counter underflows in the register start mode), clock division ratio, and the window start and end positions of refresh-permitted period. Some restrictions apply to writes to this register. For details, see section 26.3.2. Controlling writes to the IWDTCR, IWDTRCR, and IWDTCSTPR registers.

In auto start mode, the settings of this register are disabled and the settings for the option function select register 0 (OFS0) are enabled. The same settings as in the bits in this register can be performed at the option function select register 0 (OFS0). For details, see section 26.3.8. Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

TOPS[1:0] bits (Timeout Period Select)

The TOPS[1:0] bits select the timeout period, the period until the down-counter underflows, from 128, 512, 1024, and 2048 cycles, taking the divided clock specified in the CKS[3:0] bits as one cycle.

After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of IWDT clock (IWDTCLK) cycles) until the counter underflows.

section 26.2.2. IWDTCR : IWDT Control Register shows the relationship among the CKS[3:0] and TOPS[1:0] bits settings, the timeout period, and the number of IWDT clock (IWDTCLK) cycles.

Table 26.2 IWDT timeout period settings

CKS[3:0]				TOPS[1:0]		Clock division ratio	Timeout period (Number of cycles)	Number of IWDT clock (IWDTCLK) cycles
0	0	0	0	0	0	IWDTCLK	128	128
				0	1		512	512
				1	0		1024	1024
				1	1		2048	2048
0	0	1	0	0	0	IWDTCLK/16	128	2048
				0	1		512	8192
				1	0		1024	16384
				1	1		2048	32768
0	0	1	1	0	0	IWDTCLK/32	128	4096
				0	1		512	16384
				1	0		1024	32768
				1	1		2048	65536
0	1	0	0	0	0	IWDTCLK/64	128	8192
				0	1		512	32768
				1	0		1024	65536
				1	1		2048	131072
1	1	1	1	0	0	IWDTCLK/128	128	16384
				0	1		512	65536
				1	0		1024	131072
				1	1		2048	262144
0	1	0	1	0	0	IWDTCLK/256	128	32768
				0	1		512	131072
				1	0		1024	262144
				1	1		2048	524288

CKS[3:0] bits (Clock Division Ratio Select)

These bits select IWDT clock (IWDTCLK) division ratio from among division by 1, 16, 32, 64, 128, and 256.

Combined with the TOPS[1:0] bits setting, a count period between 128 and 524288 cycles of the IWDT clock (IWDTCLK) can be selected for the IWDT.

Note: In order to read the down-counter value correctly, the relationship between the peripheral clock (PCLKB) frequency and the count clock (IWDTCLK) frequency must be set appropriately. For the setting, see Note1 in Table 26.1

RPES[1:0] bits (Window End Position Select)

These bits select 75%, 50%, 25% or 0% of the count period for the window end position of the down-counter. The selected window end position should be a value smaller than the value for the window start position (that is, window start position > window end position). If the window start position is set to a value less than or equal to the window end position, the window start position setting is enabled and the window end position is set to 0%.

The counter values for the window start and end positions selected by setting the RPES[1:0] and RPSS[1:0] bits change depending on the TOPS[1:0] bits setting.

Table 26.3 lists the counter values for the window start and end positions corresponding to TOPS[1:0] bits values.

Table 26.3 Relationship between timeout periods and window start and end counter values

TOPS[1:0]		Timeout period		Window start and end counter value			
		Number of cycles	Counter value	100%	75%	50%	25%
0	0	128	0x007F	0x007F	0x005F	0x003F	0x001F
0	1	512	0x01FF	0x01FF	0x017F	0x00FF	0x007F
1	0	1024	0x03FF	0x03FF	0x02FF	0x01FF	0x00FF
1	1	2048	0x07FF	0x07FF	0x05FF	0x03FF	0x01FF

RPSS[1:0] bits (Window Start Position Select)

These bits select a counter window start position from 100%, 75%, 50%, or 25% of the count period (100% when the count starts and 0% when the counter underflows). The interval between the window start position and window end position is the refresh-permitted period and the other periods are refresh-prohibited periods.

Figure 26.2 shows the relationship between the RPSS[1:0] and RPES[1:0] bits settings and the refresh-permitted and refresh-prohibited periods.

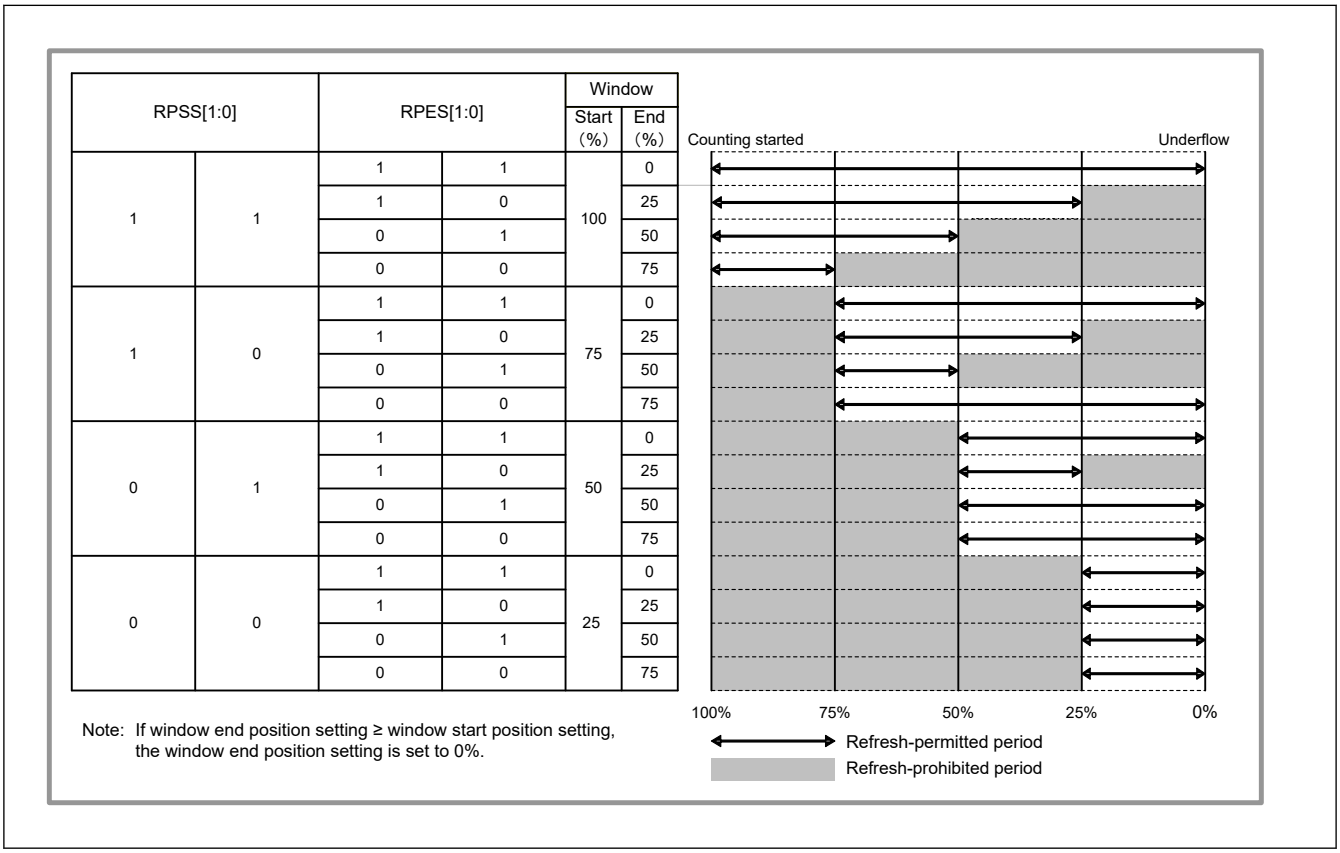


Figure 26.2 RPSS[1:0] and RPES[1:0] bits settings and refresh-permitted and refresh-prohibited periods

26.2.3 IWDTSR : IWDT Status Register

Base address: IWDT = 0x4020_2200
IWDT_NS = 0x5020_2200

Offset address: 0x4

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	REFEF	UNDF	CNTVAL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	CNTVAL[13:0]	Down-Counter Value Counter value of the down-counter	R
14	UNDFF	Underflow Flag 0: No underflow occurred. 1: Underflow occurred.	R/W
15	REFEF	Refresh Error Flag 0: No refresh error occurred. 1: Refresh error occurred.	R/W

Note: S-TYPE-3, P-TYPE-3

The IWDTSR register indicates the counter value of the down-counter and whether an underflow or refresh error occurred in the down-counter.

It is initialized by the reset source of the IWDT. IWDTSR is not initialized by other reset sources.

CNTVAL[13:0] bits (Down-Counter Value)

Read these bits to confirm the counter value of the down-counter.

Note: The read value might differ from the actual count by 1.

UNDFF flag (Underflow Flag)

Read the UNDFF flag to confirm whether an underflow occurred in the down-counter.

A value of 1 indicates that the down-counter underflowed. A value of 0 indicates that the down-counter has not underflowed.

Write 0 to the UNDFF flag to clear the value. Writing 1 has no effect.

REFEF flag (Refresh Error Flag)

Read the REFEF flag to confirm whether a refresh error (a refresh operation was performed during the prohibited period) occurred.

A value of 1 indicates that a refresh error occurred. A value of 0 indicates that no refresh error has occurred.

Write 0 to the REFEF flag to clear the value. Writing 1 has no effect.

26.2.4 IWDTRCR : IWDT Reset Control Register

Base address: IWDT = 0x4020_2200
IWDT_NS = 0x5020_2200

Offset address: 0x6

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RSTIRQS	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	RSTIRQS	Reset Interrupt Request Select 0: Enable non-maskable interrupt requests or interrupt request output. 1: Enable reset output.	R/W

Note: S-TYPE-3, P-TYPE-3

IWDTRCR register controls the reset output or interrupt request output when the down-counter of the IWDT underflows. Some restrictions apply to writes to this register. For details, see section 26.3.2. Controlling writes to the IWDTCR, IWDTRCR, and IWDTCSTPR registers.

In auto start mode, the settings of this register are disabled and the settings for the input ports (OFS0) are enabled. The same settings as in the bits in this register can be performed at the input ports (OFS0). For details, see section 26.3.8. Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

RSTIRQS bit (Reset Interrupt Request Select)

The RSTIRQS bit selects the reset output or interrupt request output when the down-counter underflows or a refresh error occurs.

26.2.5 IWDTCSTPR : IWDT Count Stop Control Register

Base address: IWDT = 0x4020_2200
IWDT_NS = 0x5020_2200

Offset address: 0x8

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SLCSTP	—	—	—	—	—	—	—
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	—	This bit is read as 1. The write value should be 1.	R/W
7	SLCSTP	CPU Sleep-Mode Count Stop Select 0: Disable count stop. 1: Stop the counter when the CPU enters CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode, or Deep Software Standby mode 1.	R/W

Note: S-TYPE-3, P-TYPE-3

The IWDTCSTPR register is used to stop the down-counter of the IWDT in case the CPU enters low-power modes. Some restrictions apply to writes to this register. For details, see section 26.3.2. Controlling writes to the IWDTCR, IWDTRCR, and IWDTCSTPR registers.

In auto start mode, the settings of this register are disabled and the settings for the option function select register 0 (OFS0) are enabled. The same settings as in the bits in this register can be performed at the option function select register 0 (OFS0) . For details, see section 26.3.8. Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

SLCSTP bit (CPU Sleep-Mode Count Stop Select)

The SLCSTP bit selects whether to stop the counter in case the CPU enters CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode, or Deep Software Standby mode 1.

26.3 Operation

26.3.1 Count operation in each start mode

The start mode of the IWDT is selected by setting the IWDT start mode selection (OFS0.IWDTSTRT) bit during a reset.

When the IWDT start mode selection (OFS0.IWDTSTRT) bit is 1 (register start mode), the IWDT Control Register (IWDTCR), IWDT Reset Control Register (IWDTRCR), and IWDT Count Stop Control Register (IWDTCSTPR) are enabled, and counting is started by writing to the IWDT Refresh Register (IWDTRR) to refresh the counter. When the IWDT start mode selection (OFS0.IWDTSTRT) bit is 0 (auto start mode), the settings of the option function select register 0 (OFS0) are enabled, and counting automatically starts after a reset.

(1) Register start mode

When the register start mode is selected by setting the IWDT start mode selection (OFS0.IWDTSTRT) bit to 1 during a reset, the IWDT Control Register (IWDTCR), IWDT Reset Control Register (IWDTRCR), and IWDT Count Stop Control Register (IWDTCSTPR) are enabled.

After the reset state is released, set the clock division ratio, window start and end positions, and timeout period in the IWDTCR register, the reset output or interrupt request output in the IWDTRCR register, and the IWDT down-counter stop control at transitions to low power modes in the IWDTCSTPR register. Then, refresh the down-counter to set the value specified by the IWDTCR.TOPS[1:0] bits in the counter and start counting down.

After that, as long as the program continues normal operation and the counter is refreshed in the refresh permitted period, the value in the counter is reset each time the counter is refreshed and counting down continues. The IWDT does not output a reset signal as long as this continues.

However, if the down counter underflows because the down-counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the IWDT outputs a reset signal or a non-maskable interrupt request (IWDT_NMIUNDF).

The reset output or interrupt request output is selected by setting the IWDTRCR.RSTIRQS bit.

Figure 26.3 shows an example of operation under the following conditions:

- Register start mode (OFS0.IWDTSTRT = 1)
- Reset output is enabled (IWDTRCR.RSTIRQS = 1)
- The window start position is 75% (IWDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (IWDTCR.RPES[1:0] = 10b)

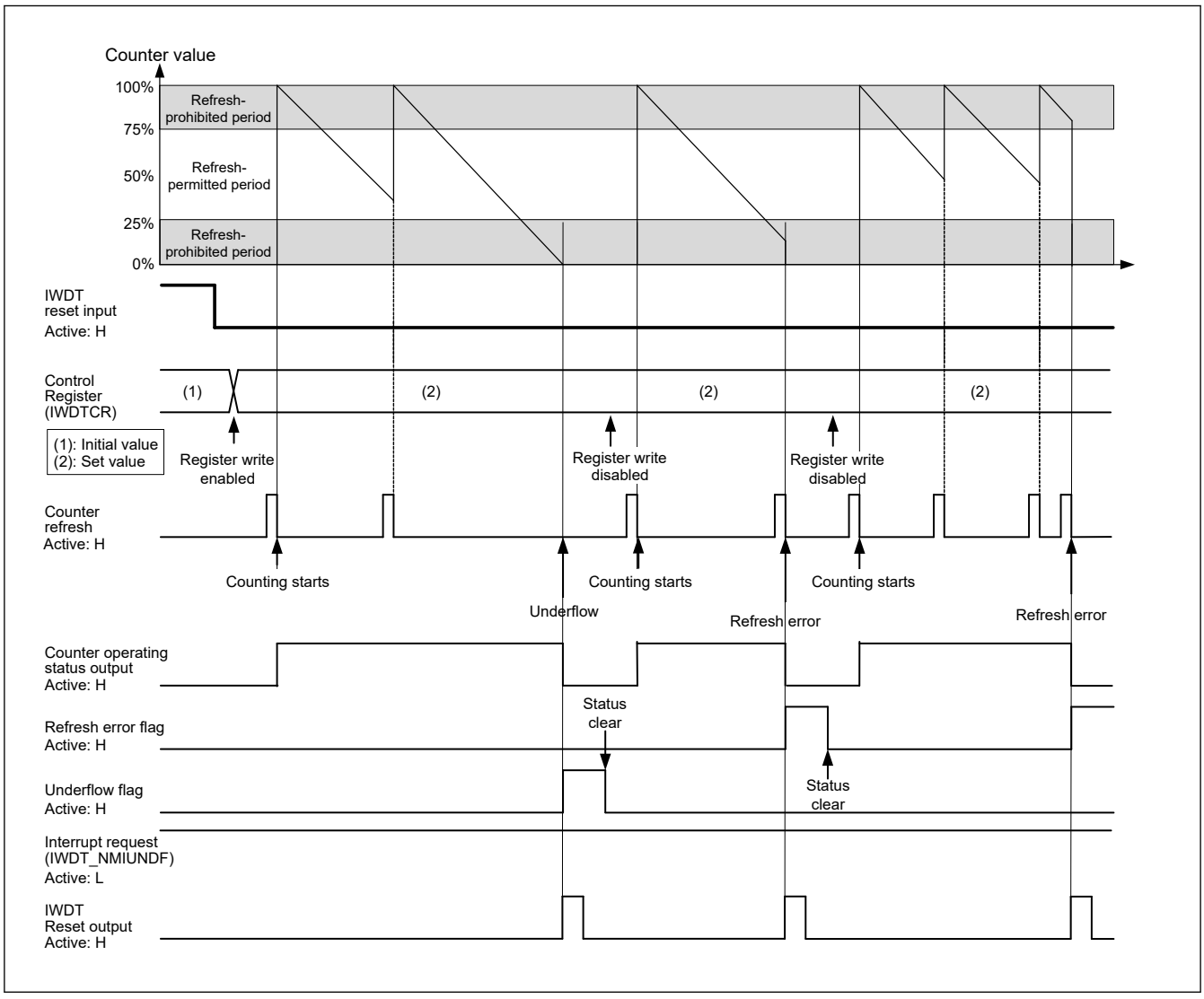


Figure 26.3 Operation example in register start mode

(2) Auto start mode

When the auto start mode is selected by setting the IWDT start mode selection (OFS0.IWDTSTRT) bit to 0 during a reset, the IWDT Control Register (IWDTCR), IWDT Reset Control Register (IWDTRCR), and IWDT Count Stop Control Register (IWDTCSTPR) are disabled.

During a reset, the clock division ratio, window start and end positions, timeout period, reset output or interrupt request output, and counter stop control at transitions to low power modes should be set through the the option function select register 0 (OFS0). Then, when the reset state is released, the timeout period set through the option function select register 0 (OFS0) is set in the down-counter, and then the down-counter automatically starts counting down.

After that, as long as the program continues normal operation and the counter is refreshed in the refresh permitted period, the value in the counter is reset each time the counter is refreshed and counting down continues. The IWDT does not output a reset signal as long as this continues.

However, if the down-counter underflows because the down counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the IWDT outputs a reset signal or a non-maskable interrupt request (IWDT_NMIUNDF).

The reset output or interrupt request output is selected by setting the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS).

Figure 26.4 shows an example of operation under the following conditions:

- Auto start mode (OFS0.IWDTSTRT = 0)
- Non-maskable interrupt request output is enabled (OFS0.IWDTRSTIRQS = 0)
- The window start position is 75% (OFS0.IWDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.IWDTRPES[1:0] = 10b)

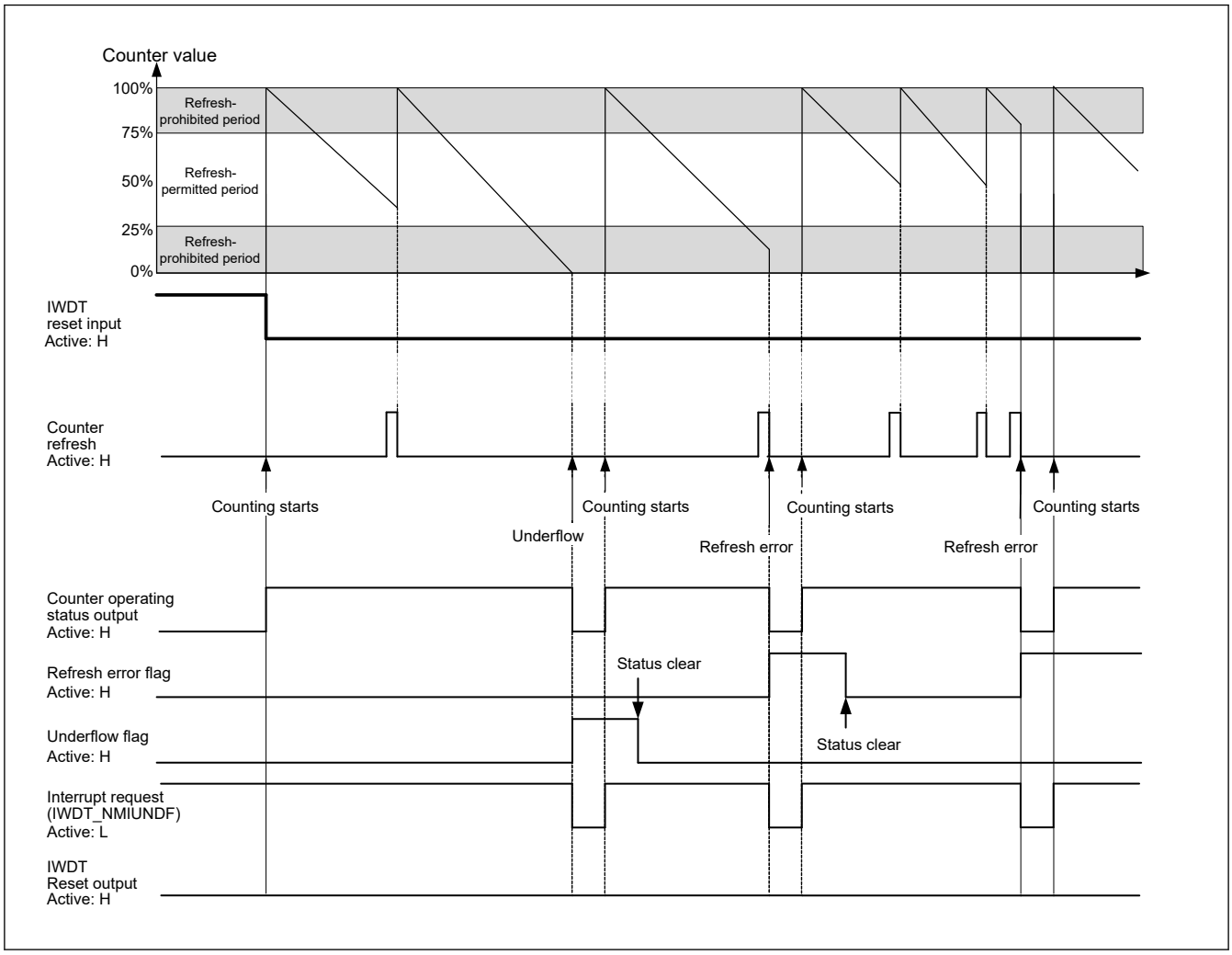


Figure 26.4 Operation example in auto start mode

26.3.2 Controlling writes to the IWDTCR, IWDTRCR, and IWDTCSTPR registers

Writing to the IWDT Control Register (IWDTCR) is possible only once between the release from the reset state and the first refresh operation.

After a refresh operation (counting starts) or by writing to the IWDTCR register, the protection signal in the IWDT becomes 1 to protect the IWDTCR register against subsequent attempts of writing.

The IWDT Reset Control Register (IWDTRCR) and the IWDT Count Stop Control Register (IWDTCSTPR) are also controlled in the same way.

This protection is released by a reset source to the IWDT. With other reset sources, the protection is not released.

Figure 26.5 shows control waveforms produced in response to writing to the IWDTCR register.

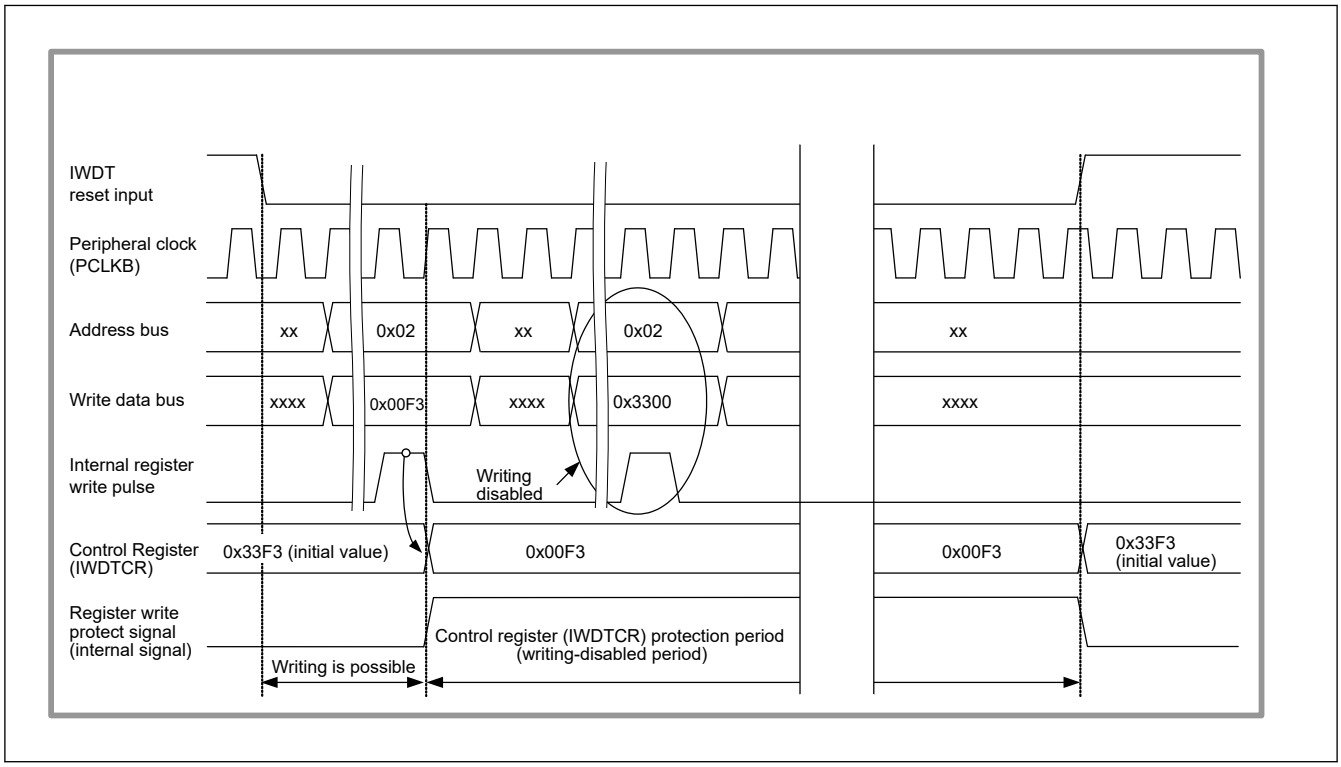


Figure 26.5 Control waveforms produced in response to writes to the IWDTCR register

26.3.3 Refresh operation

The down-counter is refreshed and starts operation (counting is started by refreshing) by writing the values 0x00 and then 0xFF to the IWDT Refresh Register (IWDTRR). If a value other than 0xFF is written after 0x00, the counter is not refreshed. After such invalid writing, correct refreshing is performed by writing 0x00 and then 0xFF again to the IWDT Refresh Register (IWDTRR).

When writing is done in the order of 0x00 (first time) → 0x00 (second time), and if 0xFF is written after that, the writing order 0x00 → 0xFF is satisfied; writing 0x00 ((n–1) th time) → 0x00 (nth time) → 0xFF is valid and correct refreshing will be done. Even when the first value written before 0x00 is not 0x00, correct refreshing will be done if the operation contains the sequence of writing 0x00 → 0xFF. Correct refreshing is also performed when a register other than IWDTRR is accessed or IWDTRR is read between writing 0x00 and writing 0xFF to IWDTRR.

[Example write sequences that are valid for refreshing the counter]

- 0x00 → 0xFF
- 0x00 ((n-1) th time) → 0x00 (nth time) → 0xFF
- 0x00 → access to another register or read from IWDTRR → 0xFF

[Example write sequences that are invalid for refreshing the counter]

- 0x23 (a value other than 0x00) → 0xFF

- 0x00 → 0x54 (a value other than 0xFF)
- 0x00 → 0xAA (0x00 and a value other than 0xFF) → 0xFF

Even when 0x00 is written to IWDTRR outside the refresh-permitted period, if 0xFF is written to IWDTRR in the refresh-permitted period, the writing sequence is valid and refreshing will be done.

After 0xFF is written to the IWDTRR register, refreshing the counter requires up to four cycles of the signal for counting (the clock division ratio select bits (IWDTCR.CKS[3:0]) determine how many cycles of the IWDT clock (IWDTCLK) make up one cycle for counting).

Therefore, writing 0xFF to the IWDTRR register should be completed four count cycles before the end position of the refresh-permitted period or the down-counter underflow. The value of the down-counter can be checked using the down-counter value bits (IWDTSR.CNTVAL[13:0]).

[Example refreshing timings]

- When the window start position is set to 0x1FFF, even if 0x00 is written to IWDTRR before 0x1FFF is reached (at 0x2002, for example), refreshing occurs if 0xFF is written to the IWDTRR register after the value of the IWDTSR.CNTVAL[13:0] bits reaches 0x1FFF.
- When the window end position is set to 0x1FFF, refreshing is done if 0x2003 (four count cycles before 0x1FFF) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 0x00 → 0xFF to the IWDTRR register.
- When the refresh-permitted period continues until count 0x0000, refreshing can be done immediately before an underflow. In this case, if 0x0003 (four count cycles before an underflow) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 0x00 → 0xFF to the IWDTRR register, no underflow occurs and refreshing is done.

Figure 26.6 shows the IWDT refresh operation waveforms when the peripheral clock (PCLKB) > IWDT clock (IWDTCLK) and the clock division ratio = IWDTCLK/1. Figure 26.7 shows the IWDT refresh operation waveforms when the peripheral clock (PCLKB) < IWDT clock (IWDTCLK) and the clock division ratio = IWDTCLK/16.

When transitioning to Software Standby mode or Deep Software Standby mode 1, the refresh procedure must be completed before the transitioning (0xFF is written after 0x00 is written to the IWDTRR register).

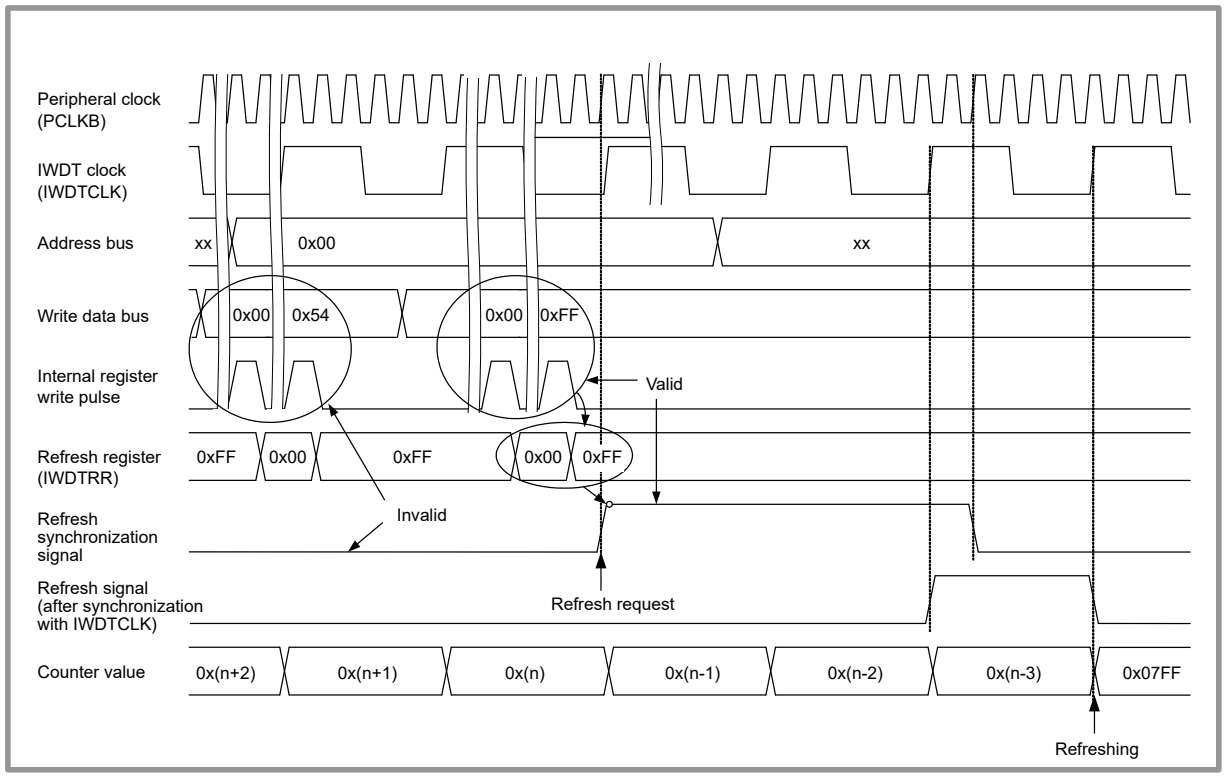


Figure 26.6 IWDT refresh operation waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

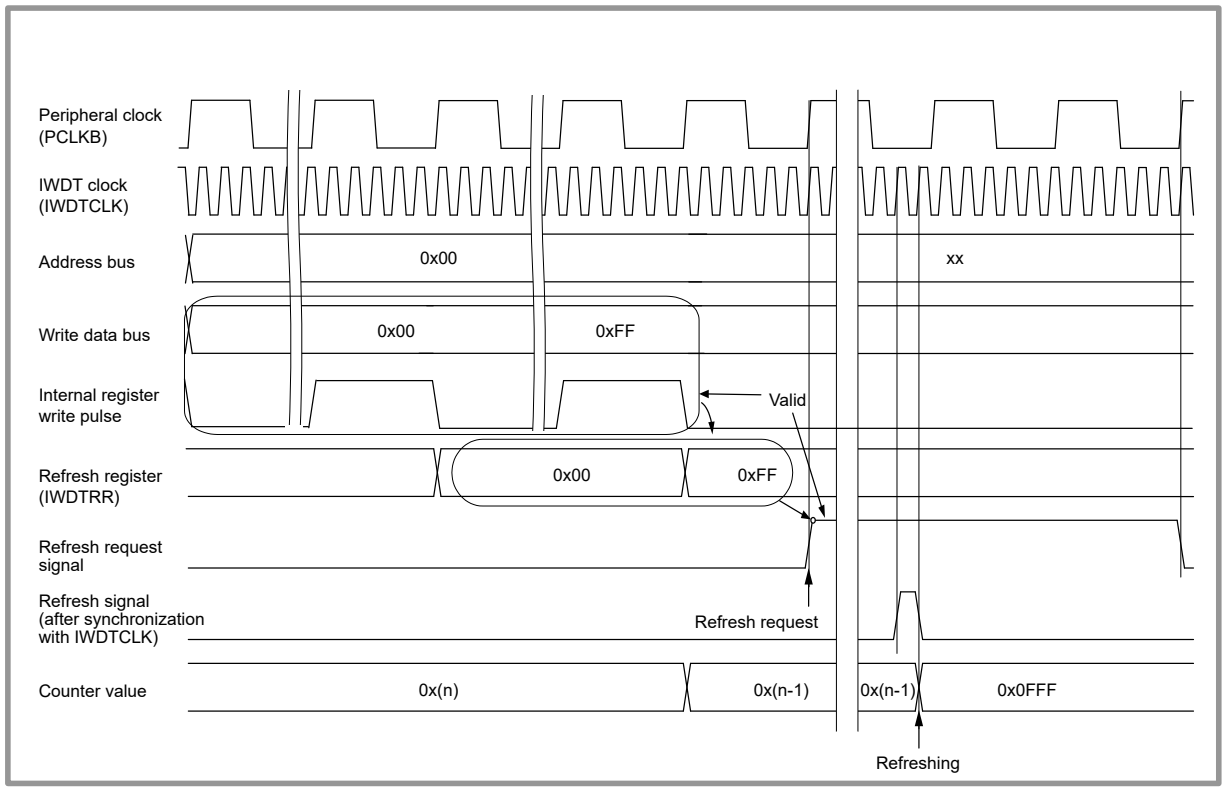


Figure 26.7 IWDT refresh operation waveforms (IWDTCR.CKS[3:0] = 0010b, IWDTCR.TOPS[1:0] = 01b)

26.3.4 Status flags

The refresh error flag (IWDTSR.REFEF) and underflow flag (IWDTSR.UNDFF) retain the source of the reset output from the IWDT or the source of the interrupt request from the IWDT.

Thus, after release from the reset state or interrupt request generation, read the IWDTSR.REFEF and IWDTSR.UNDFF flags to check for the reset or interrupt source.

For each flag, writing 0 clears the bit and writing 1 has no effect.

Leaving each flag unchanged does not affect operation. If the flags are not cleared, the earlier reset or interrupt source is cleared and the new reset or interrupt source is written when the IWDT outputs a reset signal next.

26.3.5 Reset output

When the reset interrupt select bit (IWDTRCR.RSTIRQS) is set to 1 in register start mode or when the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) is set to 1 in auto start mode, a reset signal is output for one count cycle when the down counter overflows or a refresh error occurs. In register start mode, the down counter is initialized (all bits set to 0) and kept in that state after assertion of the reset signal. After the reset is released and the program is restarted, the counter is set up again and counting down is started by refreshing. In auto start mode, counting down automatically starts after the reset output.

26.3.6 Interrupt sources

When the reset interrupt select bit (IWDTRCR.RSTIRQS) is set to 0 in register start mode or when the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) is set to 0 in auto start mode, a non-maskable interrupt (IWDT_NMIUNDF) is generated for one-count cycle when the down-counter overflows or a refresh error occurs.

26.3.7 Reading the down-counter value

Because the down-counter in the IWDT operates on the IWDT clock (IWDTCLK), the counter value cannot be read directly.

Therefore, the IWDT synchronizes the counter value with the peripheral clock (PCLKB) and stores it in the down-counter value bits (CNTVAL[13:0]) of the IWDT Status Register (IWDTSR). The counter value can be checked indirectly by reading the value stored in the IWDTSR.CNTVAL[13:0] bits.

Reading the counter value requires multiple peripheral clock (PCLKB) cycles (up to four clock cycles), and the read counter value may differ from the actual counter value by a value of one count.

Figure 26.8 shows the processing for reading the IWDT counter value when the peripheral clock (PCLKB) > IWDT clock (IWDTCLK) and the clock division ratio = IWDTCLK/1. Figure 26.9 shows the processing for reading the IWDT counter value when the peripheral clock (PCLKB) < IWDT clock (IWDTCLK) and the clock division ratio = IWDTCLK/16.

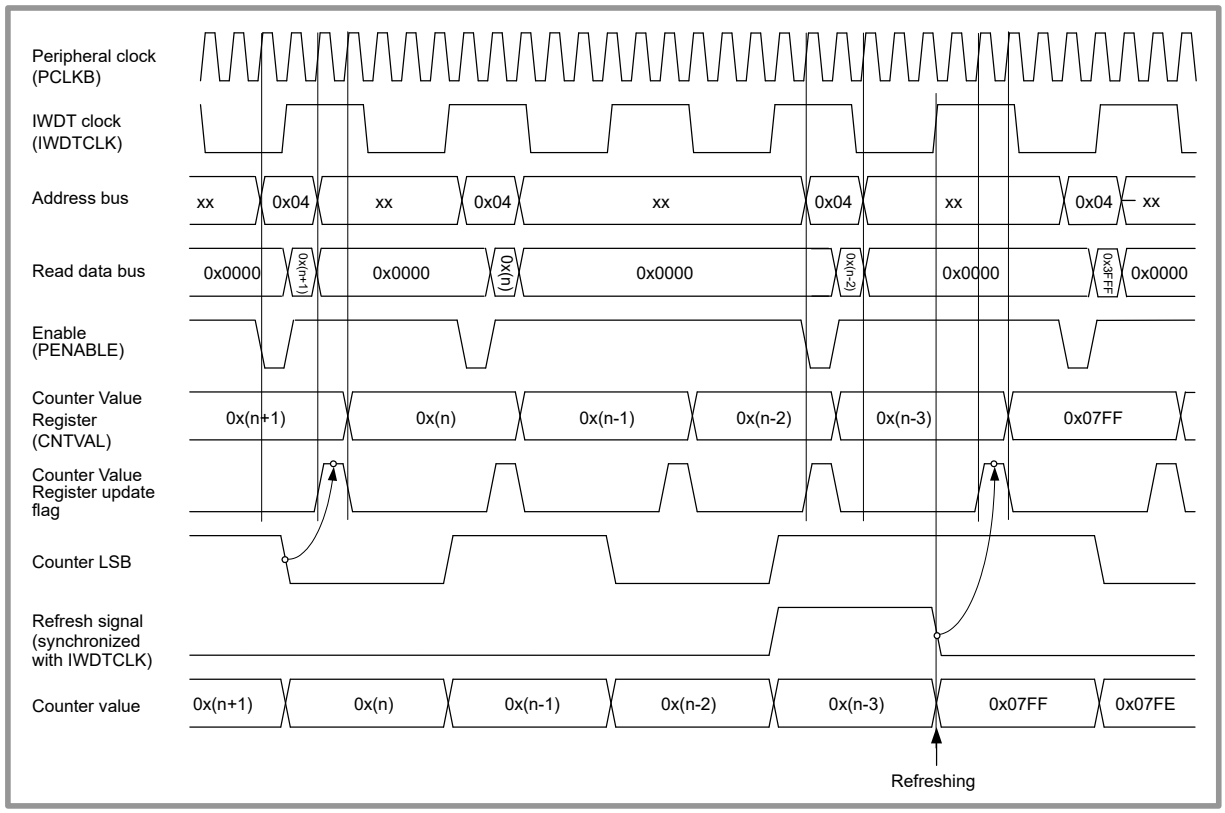


Figure 26.8 Processing for reading IWDT counter value when IWDTCR.CKS[3:0] = 0000b and IWDTCR.TOPS[1:0] = 11b

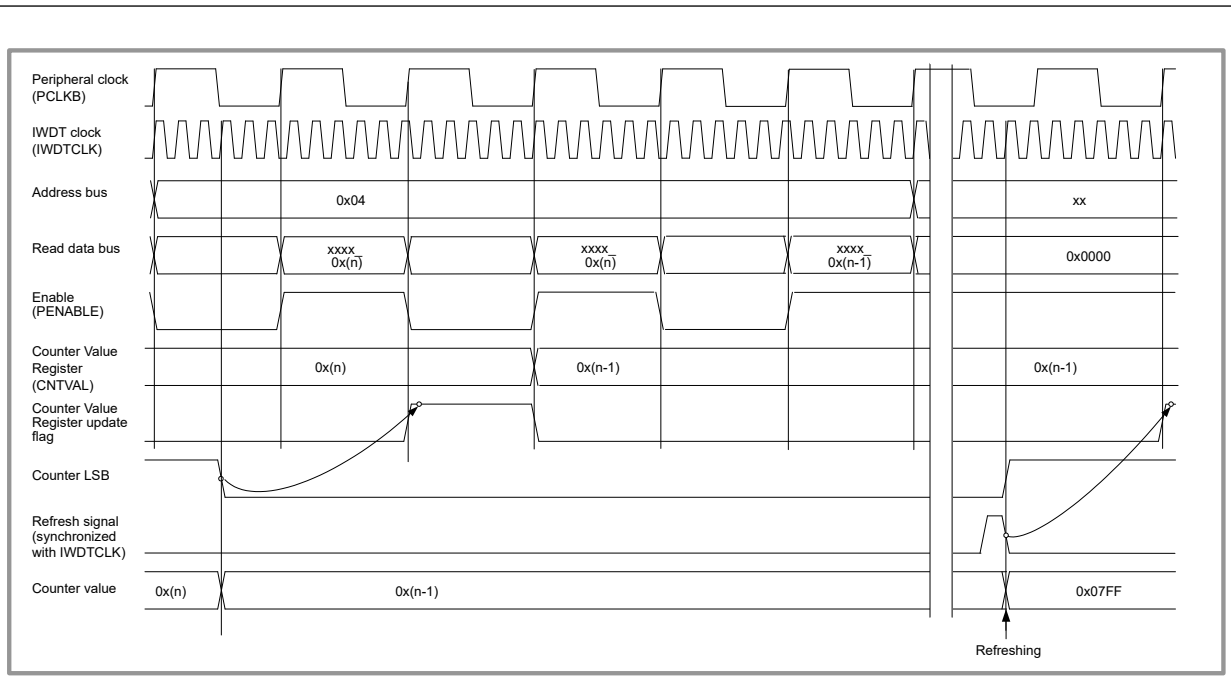


Figure 26.9 Processing for reading IWDT counter value when IWDTCR.CKS[3:0] = 0010b and IWDTCR.TOPS[1:0] = 11b

26.3.8 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Table 26.4 lists the correspondence between the option function select register 0 (OFS0) and the IWDT registers (IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR)) regarding control of the down-counter, reset or interrupt request output, and count stop function.

Control can be switched between the option function select register 0 (OFS0) and the IWDT registers (IWDTCR, IWDTRCR, and IWDTCSTPR) through the setting of the IWDT start mode selection (OFS0.IWDTSTRT) bit.

Note that the option function select register 0 (OFS0) setting should be kept unchanged during IWDT operation.

Table 26.4 Association between Option Function Select Register 0 (OFS0) and the IWDT register

Control	Function	OFS0 register (enabled in auto start mode) OFS0.IWDTSTRT = 0	IWDT register (enabled in register start mode) OFS0.IWDTSTRT = 1
Down-counter	Timeout period selection	OFS0.IWDTTOPS[1:0]	IWDTCR.TOPS[1:0]
	Clock division ratio selection	OFS0.IWDTCKS[3:0]	IWDTCR.CKS[3:0]
	Window start position selection	OFS0.IWDTRPSS[1:0]	IWDTCR.RPSS[1:0]
	Window end position selection	OFS0.IWDTRPES[1:0]	IWDTCR.RPES[1:0]
Reset output/ interrupt request output	Reset output/ interrupt request output select	OFS0.IWDTRSTIRQS	IWDTRCR.RSTIRQS
Stop the counter	CPU Sleep-mode count stop selection	OFS0.IWDTSLCSTP	IWDTCSTPR.SLCSTP

26.4 Link operation by the event link function

The IWDT is capable of link operation for a specified module when an interrupt request signal is used as an event signal. The IWDT outputs an event signal on an underflow of the down-counter or generation of a refresh error.

It outputs an event signal regardless of the setting of the reset interrupt request select bit (IWDTRCR.RSTIRQS) in register start mode or the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) in auto start mode. It can output an event signal at generation of the next interrupt source while retaining the refresh error flag (IWDTSR.REFEF) or underflow flag (IWDTSR.UNDFF).

27. Ethernet MAC Controller (ETHERC)

27.1 Overview

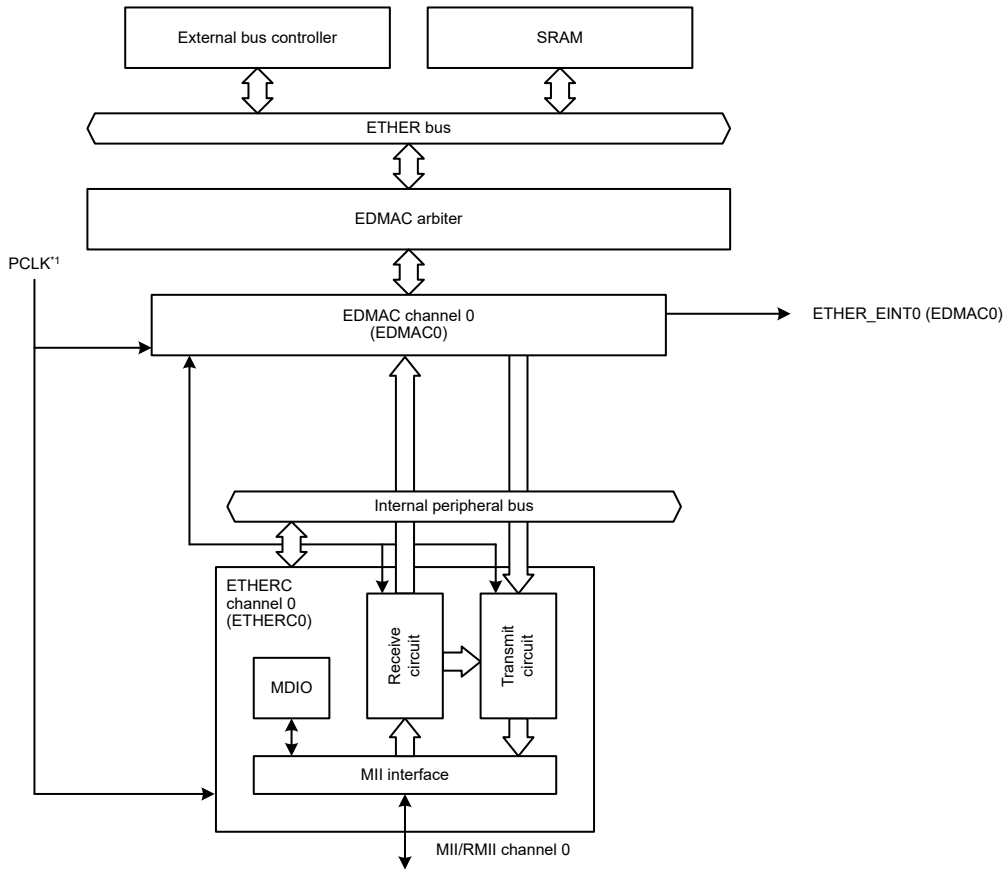
The MCU provides a one-channel Ethernet Controller (ETHERC) compliant with the Ethernet or IEEE802.3 Media Access Control (MAC) layer protocol. Each ETHERC channel has one channel of the MAC layer interface. Connecting the MCU to the physical layer LSI (PHY-LSI) allows transmission and reception of frames compliant with the Ethernet/IEEE802.3 standard. The ETHERC is connected to the Ethernet DMA Controller (EDMAC), so data can be transferred without using the CPU.

Table 27.1 lists the ETHERC specifications, Figure 27.1 shows the configuration, and Table 27.2 lists the I/O pins.

Figure 27.1 and Figure 27.3 show examples connections of the MCU to an external PHY-LSI.

Table 27.1 ETHERC specifications

Parameter	Specifications
Number of channels	One channel
Protocol	Flow control compliant with IEEE802.3x
Data transmission/reception	Frames compliant with the Ethernet/IEEE802.3 standard can be transmitted and received
Bit rate	Supports 10 Mbps and 100 Mbps
Operation modes	Supports full-duplex and half-duplex modes
Interfaces	Media Independent Interface (MII), Reduced Media Independent Interface (RMII), compliant with the IEEE802.3u standard
Functions	• Magic Packet™ detection • Wake-on-LAN (WOL) signal output
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security and Privilege attribution can be set.



Note 1. PCLK: PCLKA
When using the ETHERC, 12.5 MHz ≤ PCLK ≤ 120 MHz.

Figure 27.1 ETHERC configuration

Table 27.2 ETHERC I/O pins (1 of 2)

Operating mode	Pin name	I/O	Description
MII	ET0_TX_CLK *1	Input	Transmit clock Timing reference signal for outputting the ET0_TX_EN, ET0_ETXD3 to ET0_ETXD0, and ET0_TX_ER signals.
	ET0_RX_CLK *1	Input	Receive clock Timing reference signal for inputting the ET0_RX_DV, ET0_ERXD3 to ET0_ERXD0, and ET0_RX_ER signals.
	ET0_TX_EN *1	Output	Transmit data valid This signal indicates that valid transmit data was output on pins ET0_ETXD3 to ET0_ETXD0.
	ET0_ETXD3 to ET0_ETXD0 *1	Output	4-bit transmit data
	ET0_TX_ER *1	Output	Transmit error This signal notifies the PHY-LSI that an error occurred during transmission.
	ET0_RX_DV *1	Input	Receive data valid This signal indicates that valid receive data is on pins ET0_ERXD3 to ET0_ERXD0.
	ET0_ERXD3 to ET0_ERXD0 *1	Input	4-bit receive data
	ET0_RX_ER *1	Input	Receive error This signal indicates that there is an error in a frame that is being transferred from the PHY-LSI to the ETHERC.
	ET0_CRS *1	Input	Carrier sense
	ET0_COL *1	Input	Collision detection signal
	ET0_MDC *1	Output	Management data clock Reference clock signal for transfer of information on the ET0_MDIO pin.
	ET0_MDIO *1	I/O	Management data Input/Output Bidirectional data signal for exchanging management data with the PHY-LSI.
	ET0_LINKSTA	Input	Link status input from the PHY-LSI
	ET0_EXOUT	Output	General output pin
	ET0_WOL	Output	Wake-on-LAN. This signal indicates that a Magic Packet was received

Table 27.2 ETHERC I/O pins (2 of 2)

Operating mode	Pin name	I/O	Description
RMII	REF50CK0 *2	Input	Reference clock Timing reference signal for the RMII0_TXD_EN, RMII0_TXD1 to RMII0_TXD0, RMII0_CRS_DV, RMII0_RXD1 to RMII0_RXD0, and RMII0_RX_ER pins.
	RMII0_TXD_EN *2	Output	Transmit data valid This signal indicates that valid transmit data was output on the RMII0_TXD1 and RMII0_TXD0 pins.
	RMII0_TXD1 to RMII0_TXD0 *2	Output	2-bit transmit data
	RMII0_CRS_DV *2	Input	Carrier sense/receive data valid This signal indicates that valid receive data is on the RMII0_RXD1 and RMII0_RXD0 pins.
	RMII0_RXD1 to RMII0_RXD0 *2	Input	2-bit receive data
	RMII0_RX_ER *2	Input	Receive error This signal indicates that there is an error in a frame that is being transferred from the PHY-LSI to the ETHERC. See the note in section 27.5.2. Input to RMII0_RX_ER Pin While RMII Is Selected.
	ET0_MDC *2	Output	Management data clock Reference clock signal for transfer of information on the ET0_MDIO pin
	ET0_MDIO *2	I/O	Management data Input/Output Bidirectional data signal for exchanging management data with the PHY-LSI.
	ET0_LINKSTA	Input	Link status input from the PHY-LSI.
	ET0_EXOUT	Output	General output pin
	ET0_WOL	Output	Wake-on-LAN. This signal indicates that a Magic Packet was received.

Note 1. MII signal compliant with IEEE802.3u.
Note 2. RMII signal compliant with IEEE802.3u.

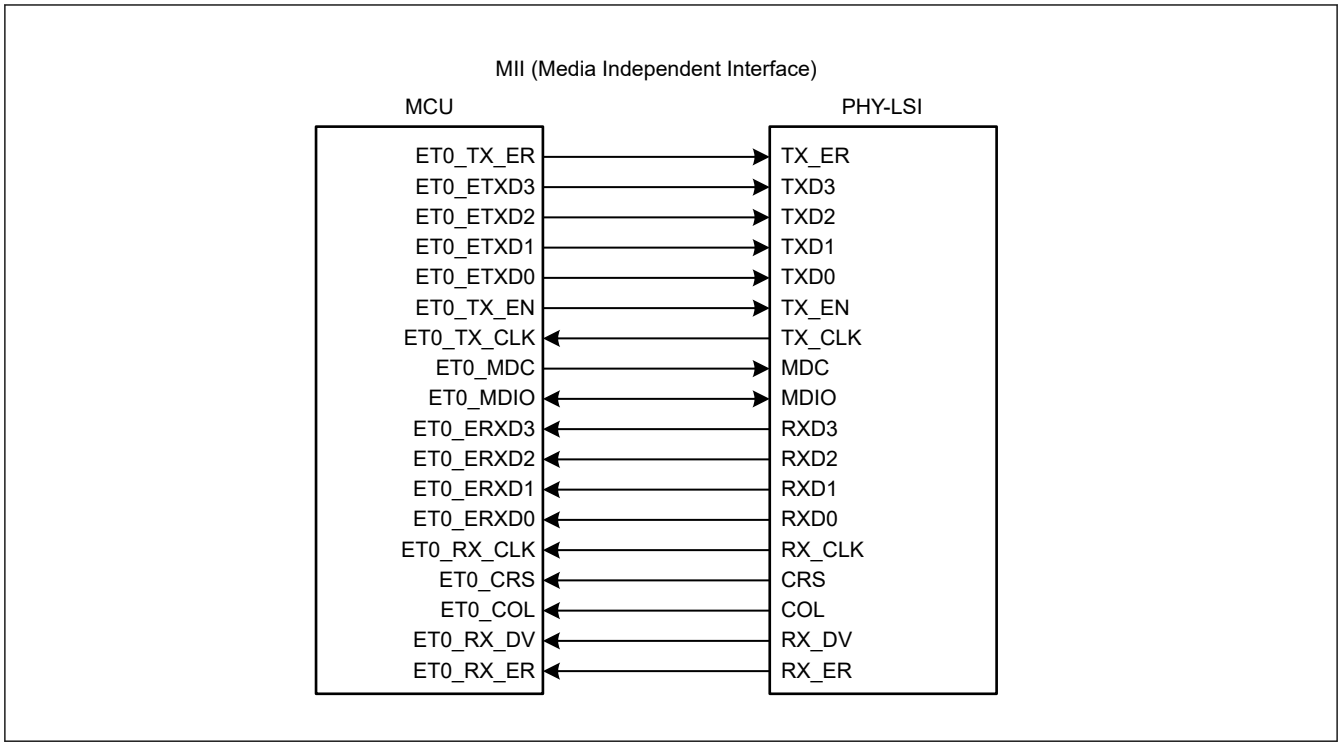


Figure 27.2 Example of connection with PHY-LSI for MII

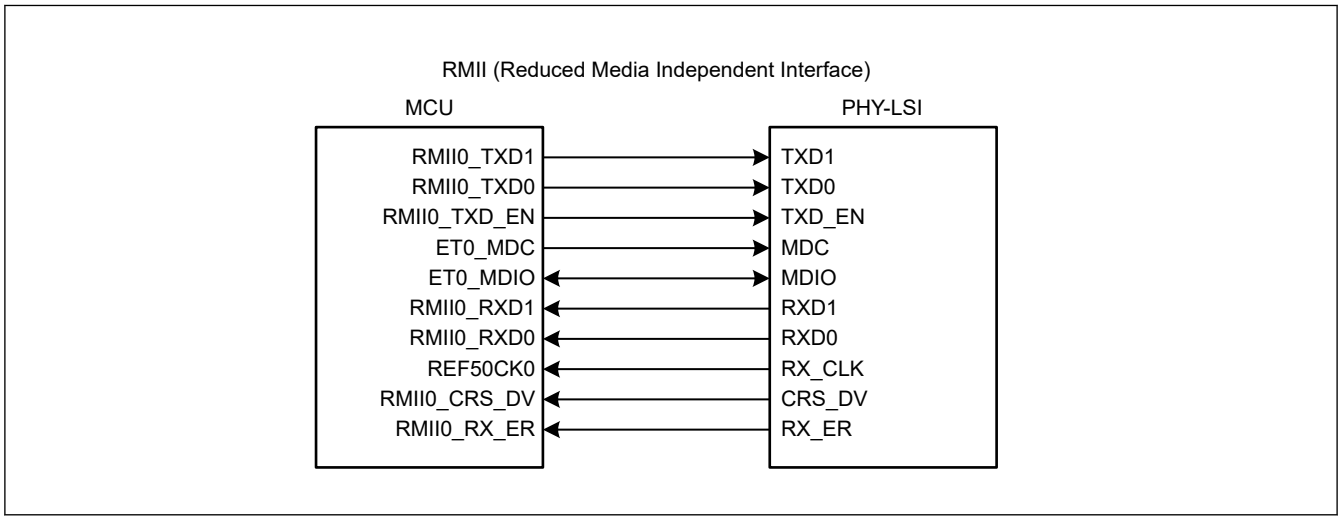


Figure 27.3 Example of connection with PHY-LSI for RMII

27.2 Register Descriptions

27.2.1 ECMR : ETHERC Mode Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	TPC	ZPF	PFR	RXF	TXF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PRCEF	—	—	MPDE	—	—	RE	TE	—	ILB	RTM	DM	PRM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRM	Promiscuous Mode 0: Disable promiscuous mode 1: Enable promiscuous mode.	R/W
1	DM	Duplex Mode 0: Half-duplex mode 1: Full-duplex mode.	R/W
2	RTM	Bit Rate 0: 10 Mbps 1: 100 Mbps.	R/W
3	ILB	Internal Loopback Mode 0: Perform normal data transmission or reception 1: Loop data back in the ETHERC when full-duplex mode is selected.	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	TE	Transmission Enable 0: Disable transmit function 1: Enable transmit function.	R/W
6	RE	Reception Enable 0: Disable receive function 1: Enable receive function.	R/W
8:7	—	These bits are read as 0. The write value should be 0.	R/W
9	MPDE	Magic Packet Detection Enable 0: Disable Magic Packet detection 1: Enable Magic Packet detection.	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	PRCEF	CRC Error Frame Receive Mode 0: Notify EDMAC of a CRC error 1: Do not notify EDMAC of a CRC error.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	TXF	Transmit Flow Control Operating Mode 0: Disable automatic PAUSE frame transmission (PAUSE frame is not automatically transmitted) 1: Enable automatic PAUSE frame transmission (PAUSE frame is automatically transmitted as required).	R/W
17	RXF	Receive Flow Control Operating Mode 0: Disable PAUSE frame detection 1: Enable PAUSE frame detection.	R/W

Bit	Symbol	Function	R/W
18	PFR	PAUSE Frame Receive Mode 0: Do not transfer PAUSE frame to the EDMAC 1: Transfer PAUSE frame to the EDMAC.	R/W
19	ZPF	0 Time PAUSE Frame Enable 0: Do not use PAUSE frames that containing a pause_time parameter of 0 1: Use PAUSE frames that containing a pause_time parameter of 0.	R/W
20	TPC	PAUSE Frame Transmit 0: Transmit PAUSE frame even during a PAUSE period 1: Do not transmit PAUSE frame during a PAUSE period.	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The ECMR register controls ETHERC operation. Except for the TE and RE bits, set the bits in this register during initialization after a reset. When rewriting this register outside the initialization process, set the EDMAC0.EDMR.SWR bit to 1 to reset the EDMAC and ETHERC, then set this register again.

PRM bit (Promiscuous Mode)

When the PRM bit is set to 1, the ETHERC operates in promiscuous mode, where all Ethernet frames are received. In promiscuous mode, the ETHERC receives all valid frames regardless of whether the address matches the destination or broadcast address, and regardless of the multicast bit setting.

RTM bit (Bit Rate)

The RTM bit sets the bit rate when the RMII is selected.

ILB bit (Internal Loopback Mode)

When the ILB bit is set to 1, transmit frames can be looped back in the MCU. Set the DM bit to 1 (full-duplex mode) to perform a loopback test.

TE bit (Transmission Enable)

When the TE bit is set to 1, the ETHERC transmit function is enabled. When the TE bit is set to 0, the transmit function is disabled after the frame being processed is completely transmitted.

RE bit (Reception Enable)

When the RE bit is set to 1, the ETHERC receive function is enabled. When the RE bit is set to 0, the receive function is disabled after the frame being processed is completely received.

PRCEF bit (CRC Error Frame Receive Mode)

When the PRCEF bit is set to 1, the EDMAC is not notified that a CRC error has occurred even when the error is detected in a receive frame. Accordingly, the EDMAC0.EESR.CERF flag and RFS0 bit in receive descriptor 0 (RD0) do not set to 1.

ZPF bit (0 Time PAUSE Frame Enable)

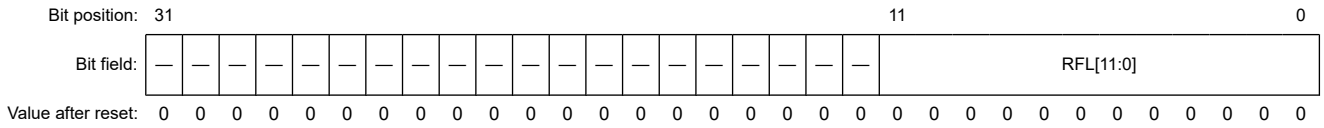
When the ZPF bit is 1, a PAUSE frame with a pause_time parameter of 0 is transmitted when a PAUSE frame transmit request is canceled before the PAUSE time of the previously transmitted PAUSE frame has elapsed. After the PAUSE frame containing the pause_time parameter of 0 is received, the ETHERC is ready for transmission.

When the ZPF bit is 0, even if the PAUSE frame transmit request from the receive FIFO is canceled, the next PAUSE frame is not transmitted until the PAUSE time of the previously transmitted PAUSE frame has elapsed. When a PAUSE frame containing a pause_time parameter of 0 is received, it is discarded.

27.2.2 RFLR : Receive Frame Maximum Length Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0x08



Bit	Symbol	Function	R/W
11:0	RFL[11:0]	Receive Frame Maximum Length The set value becomes the maximum frame length. The minimum value that can be set is 1,518 bytes, and the maximum value that can be set is 2,048 bytes. Values less than 1,518 bytes are regarded as 1,518 bytes, and values larger than 2,048 bytes are regarded as 2,048 bytes.	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RFLR register specifies the maximum frame length that can be received by the MCU. Set the length in bytes. Do not rewrite this register while the ECMR.RE bit is 1 (receive function enabled).

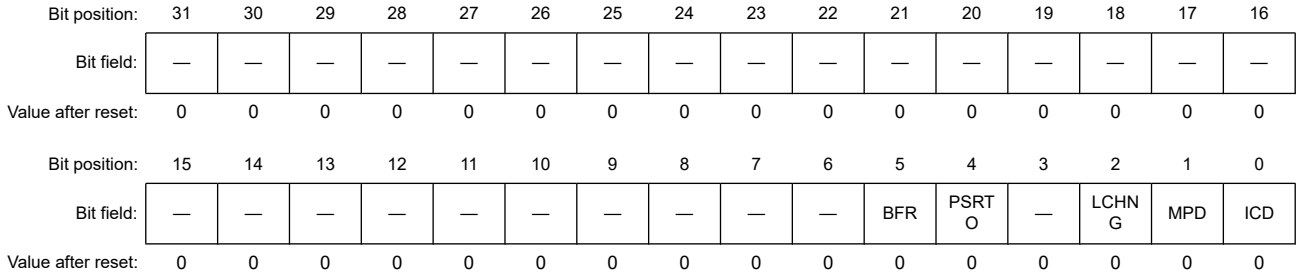
RFL[11:0] bits (Receive Frame Maximum Length)

The RFL[11:0] bits set the frame length to be checked. The frame length is the number of bytes in a field, extending from the destination address to the frame check sequence [FCS] of the received frame. When this length exceeds the RFL[11:0] bit value, the EDMAC is notified of a frame-too-long error, and the excess data is discarded.

27.2.3 ECSR : ETHERC Status Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0x10



Bit	Symbol	Function	R/W
0	ICD	False Carrier Detect Flag 0: PHY-LSI has not detected a false carrier on the line 1: PHY-LSI detected a false carrier on the line.	R/W*1
1	MPD	Magic Packet Detect Flag 0: Magic Packet not detected 1: Magic Packet detected.	R/W*1
2	LCHNG	Link Signal Change Flag 0: Change in the ET0_LINKSTA signal not detected 1: Change in the ET0_LINKSTA signal detected (high to low, or low to high).	R/W*1
3	—	This bit is read as 0. The write value should be 0.	R/W
4	PSRTO	PAUSE Frame Retransmit Over Flag 0: PAUSE frame retransmit count has not reached the upper limit 1: PAUSE frame retransmit count reached the upper limit.	R/W*1

Bit	Symbol	Function	R/W
5	BFR	Continuous Broadcast Frame Reception Flag 0: Continuous reception of broadcast frames not detected 1: Continuous reception of broadcast frames detected.	R/W*1
31:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. Write 1 to clear the flag.

The ECSR register indicates the status of the ETHERC. When any flag in the ECSR register is set to 1 while the associated bit in the ECSIPR register is 1 (interrupt enabled), the EDMAC0.EESR.ECI flag is set to 1.

ICD flag (False Carrier Detect Flag)

The ICD flag indicates that the PHY-LSI has detected a false carrier on the line. The flag is set to 1 when a receive error signal shown in Figure 27.11 is received from the PHY-LSI. The information might not be correct when signals input from the PHY-LSI change faster than software recognizes the change. Check the timing of the PHY-LSI.

LCHNG flag (Link Signal Change Flag)

The LCHNG flag indicates that the ET0_LINKSTA signal input from the PHY-LSI has changed from high to low, or from low to high. Check the PSR.LMON flag for the current link status. See section 27.5.1. Preventing the LCHNG Flag from Erroneously Setting to 1 for more information.

PSRTO flag (PAUSE Frame Retransmit Over Flag)

The PSRTO flag indicates that the number of retransmissions reached the value set in the TPAUSER register when retransmitting a PAUSE frame while automatic PAUSE frame transmission is enabled.

27.2.4 ECSIPR : ETHERC Interrupt Enable Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	BFSIPR	PSRTOIP	—	LCHNGIP	MPDIP	ICDIP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ICDIP	False Carrier Detect Interrupt Enable 0: Disable interrupt notification 1: Enable interrupt notification.	R/W
1	MPDIP	Magic Packet Detect Interrupt Enable 0: Disable interrupt notification 1: Enable interrupt notification.	R/W
2	LCHNGIP	LINK Signal Change Interrupt Enable 0: Disable interrupt notification 1: Enable interrupt notification.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	PSRTOIP	PAUSE Frame Retransmit Over Interrupt Enable 0: Disable interrupt notification 1: Enable interrupt notification.	R/W

Bit	Symbol	Function	R/W
5	BFSIPR	Continuous Broadcast Frame Reception Interrupt Enable 0: Disable interrupt notification 1: Enable interrupt notification.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The ECSIPR register selects whether to notify the EDMAC of the status indicated in the ECSR register. Each bit is associated with the flag with the same bit number in the ECSR register.

27.2.5 PIR : PHY Interface Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MDI	MDO	MMD	MDC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0

Bit	Symbol	Function	R/W
0	MDC	MII/RMII Management Data Clock This value is output from the ET0_MDC pin to supply the management data clock to the MII or RMII.	R/W
1	MMD	MII/RMII Management Mode 0: Read 1: Write.	R/W
2	MDO	MII/RMII Management Data-Out This value is output from the ET0_MDIO pin when the MMD bit is 1 (write), and not when MMD is 0 (read).	R/W
3	MDI	MII/RMII Management Data-In This bit indicates the level of the ET0_MDIO pin. The write value should be 0.	R
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The PIR register accesses registers in the PHY-LSI through the MII or RMII. The management clock and management data are controlled by software. See section 27.3.4. Accessing the MII and RMII Registers for details on accessing the MII and RMII registers.

27.2.6 PSR : PHY Status Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0x28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LMON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

Bit	Symbol	Function	R/W
0	LMON	ET0_LINKSTA Pin Status Flag The link status can be read by connecting the link signal output from the PHY-LSI to the ET0_LINKSTA pin. For details on the polarity, see the specifications of the connected PHY-LSI.	R
31:1	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3

The PSR register monitors interface signals from the PHY-LSI.

27.2.7 RDMLR : Random Number Generation Counter Upper Limit Setting Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0x40

Bit position:	31												19																	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	RMD[19:0]																	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Note: S-TYPE3, P-TYPE3

Bit	Symbol	Function	R/W
19:0	RMD[19:0]	Random Number Generation Counter 0x00000: Normal operation 0x00001 to 0xFFFFF: Setting prohibited.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

The RDMLR register specifies the maximum value for the counter used in the random number generator. Do not rewrite this register while the ECMR.TE bit is 1 (transmit function enabled) or while the ECMR.RE bit is 1 (receive function enabled).

27.2.8 IPGR : Interpacket Gap Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0x50

Bit position:	31																											4		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPG[4:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0

Bit	Symbol	Function	R/W
4:0	IPG[4:0]	Interpacket Gap 0x00: 16 bit times 0x01: 20 bit times ⋮ 0x14: 96 bit times (initial value) ⋮ 0x1F: 140 bit times.	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The IPGR register specifies the interpacket gap (IPG) value. Do not rewrite this register while the ECMR.TE bit is 1 (transmit function enabled) or while the ECMR.RE bit is 1 (receive function enabled). See section 27.3.6. Adjusting Transmission Efficiency by Changing the IPG for details on the IPG.

27.2.9 APR : Automatic PAUSE Frame Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0x54

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AP[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	AP[15:0]	Automatic PAUSE Time Setting These bits set the value of the pause_time parameter for PAUSE frames that are automatically transmitted. Transmission is not performed until a period equal to the set value multiplied by 512 bit times has elapsed.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The APR register specifies the PAUSE time for PAUSE frames that are automatically transmitted. The value set in the APR register is used for the pause_time parameter of the PAUSE frame. Do not rewrite this register while the ECMR.TE bit is 1 (transmit function enabled) or while the ECMR.RE bit is 1 (receive function enabled).

27.2.10 MPR : Manual PAUSE Frame Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0x58

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MP[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
15:0	MP[15:0]	Manual PAUSE Time Setting These bits set the value of the pause_time parameter for PAUSE frames that are manually transmitted. Transmission is not performed until the set value multiplied by 512 bit times has elapsed. The read value is undefined.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

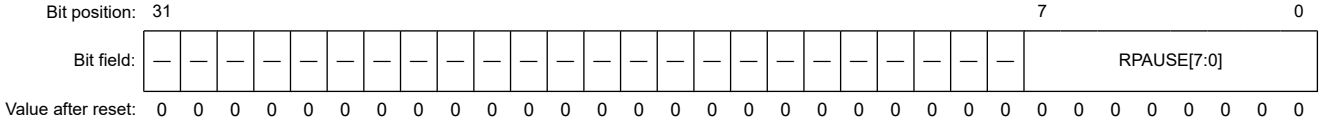
Note: S-TYPE3, P-TYPE3

The MPR register specifies the PAUSE time for PAUSE frames that are manually transmitted. The value set in the MPR register is used for the pause_time parameter of the PAUSE frame. When a value is set to this register, a PAUSE frame is transmitted. Rewrite this register while the ECMR.TE bit is 1 (transmit function enabled).

27.2.11 RFCF : Received PAUSE Frame Counter

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0x60



Bit	Symbol	Function	R/W
7:0	RPAUSE[7:0]	Received PAUSE Frame Count Number of received PAUSE frames.	R
31:8	—	These bits are read as 0.	R

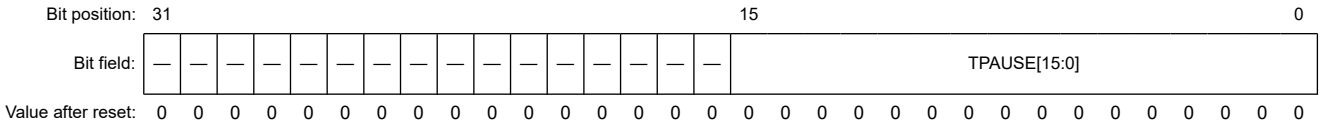
Note: S-TYPE3, P-TYPE3

The RFCF register is a counter that indicates the number of received PAUSE frames. The counter is reset after this register is read.

27.2.12 TPAUSER : PAUSE Frame Retransmit Count Setting Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0x64



Bit	Symbol	Function	R/W
15:0	TPAUSE[15:0]	Automatic PAUSE Frame Retransmit Setting 0x0000 0: Number of retransmissions is unlimited 0x0000 1: Maximum number of retransmissions is 1 ⋮ ⋮ 0xFFFF F: Maximum number of retransmissions is 65,535.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

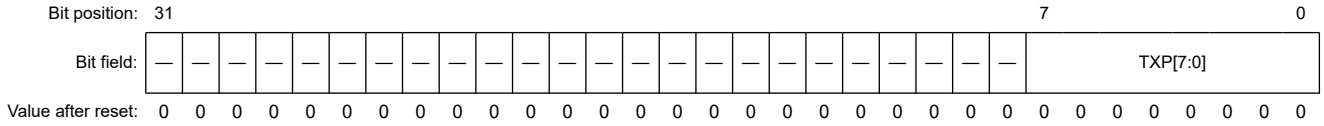
Note: S-TYPE3, P-TYPE3

The TPAUSECR register selects the maximum number of times a PAUSE frame is automatically transmitted. Do not rewrite this register while the ECMR.TE bit is 1 (transmit function enabled).

27.2.13 TPAUSECR : PAUSE Frame Retransmit Counter

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0x68



Bit	Symbol	Function	R/W
7:0	TXP[7:0]	PAUSE Frame Retransmit Count Number of times a PAUSE frame was retransmitted.	R
31:8	—	These bits are read as 0.	R

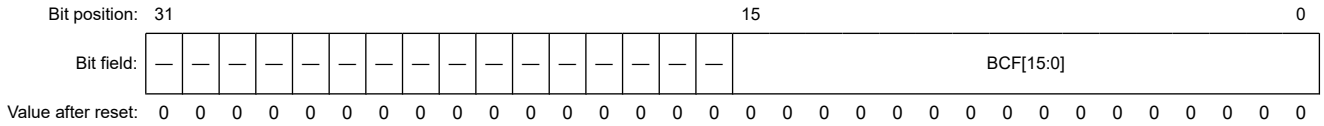
Note: S-TYPE3, P-TYPE3

The TPAUSECR register is a counter that indicates the number of times a PAUSE frame was automatically retransmitted. The counter is reset after this register is read.

27.2.14 BCFRR : Broadcast Frame Receive Count Setting Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0x6C



Bit	Symbol	Function	R/W
15:0	BCF[15:0]	Broadcast Frame Continuous Receive Count Setting 0x0000 Number of reception is unlimited : 0x0001 Receive 1 frame. : ⋮ 0xFFFF Receive 65,535 frames. :	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

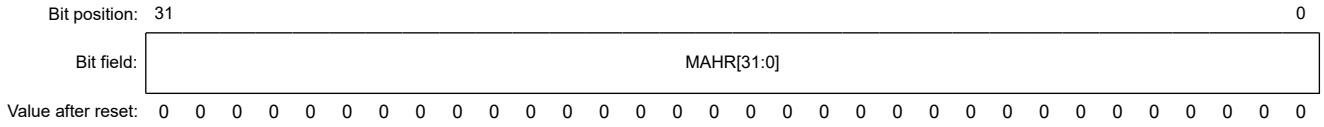
Note: S-TYPE3, P-TYPE3

The BCFRR register specifies the number of times broadcast frames can be received continuously. When the number of received frames exceeds the BCF[15:0] bit value, the excess broadcast frames are discarded. Do not rewrite this register while the EMCR.RE bit is 1 (receive function enabled).

27.2.15 MAHR : MAC Address Upper Bit Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0xC0



Bit	Symbol	Function	R/W
31:0	MAHR[31:0]	MAC Address Upper Bit See the following.	R/W

Note: S-TYPE3, P-TYPE3

The MAHR register specifies the upper 32 bits ([47:16]) of the 48-bit MAC address. For example, if the MAC address is 01-23-45-67-89-AB, set the register to 0x01234567.

Set the MAHR register during initialization after a reset. Do not rewrite this register while the ECMR.TE bit is 1 (transmit function enabled) or while the ECMR.RE bit is 1 (receive function enabled). When rewriting this register, set the EDMAC0.EDMR.SWR bit to 1 to reset the EDMAC and ETHERC, then set this register again.

27.2.16 MALR : MAC Address Lower Bit Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0xC8

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MALR[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	MALR[15:0]	MAC Address Lower Bit These bits set the lower 16 bits of the MAC address.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The MALR register specifies the lower 16 bits of the 48-bit MAC address. For example, if the MAC address is 01-23-45-67-89-AB, set the register to 0x000089AB.

Set the MALR register during initialization after a reset. Do not rewrite this register while the ECMR.TE bit is 1 (transmit function enabled) or while the ECMR.RE bit is 1 (receive function enabled). When rewriting this register, set the EDMAC0.EDMR.SWR bit to 1 to reset the EDMAC and ETHERC, then set this register again.

27.2.17 TROCR : Transmit Retry Over Counter Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0xD0

Bit position:	31																													0
Bit field:	TROCR[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TROCR[31:0]	Transmit Retry Over Counter See the following.	R/W

Note: S-TYPE3, P-TYPE3

The TROCR register is a counter that indicates the number of frames that failed to be retransmitted. The register value is incremented by 1 when a frame fails to be retransmitted 15 times. The counter stops when the register value becomes 0xFFFF_FFFF. Writing any value to the TROCR register clears the counter value to 0.

27.2.18 CDCR : Late Collision Detect Counter Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0xD4

Bit position:	31																												0
Bit field:	CDCR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CDCR[31:0]	Late Collision Detect Counter See the following.	R/W

Note: S-TYPE3, P-TYPE3

The CDCR register is a counter that indicates the number of late collisions that are detected after transmission starts. When the register value becomes 0xFFFF_FFFF, the counter stops. Writing any value to the CDCR register clears the counter value to 0.

27.2.19 LCCR : Lost Carrier Counter Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0xD8

Bit position:	31																												0
Bit field:	LCCR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	LCCR[31:0]	Lost Carrier Counter See the following.	R/W

Note: S-TYPE3, P-TYPE3

The LCCR register is a counter that indicates the number of times a loss of carrier is detected during frame transmission.

When the register value becomes 0xFFFF FFFF, the counter stops. Writing any value to the LCCR register clears the counter value to 0.

27.2.20 CNDCR : Carrier Not Detect Counter Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0xDC

Bit position:	31																												0
Bit field:	CNDCR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CNDCR[31:0]	Carrier Not Detect Counter See the following.	R/W

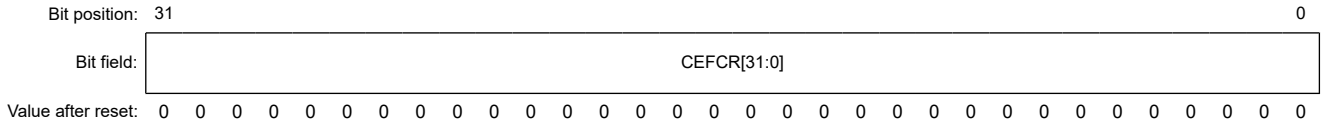
Note: S-TYPE3, P-TYPE3

The CNDCR register is a counter that indicates the number of times a carrier is not detected during preamble transmission. When the register value becomes 0xFFFF_FFFF, the counter stops. Writing any value to the CNDCR register clears the counter value to 0.

27.2.21 CEFCR : CRC Error Frame Receive Counter Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0xE4



Bit	Symbol	Function	R/W
31:0	CEFCR[31:0]	CRC Error Frame Receive Counter See the following.	R/W

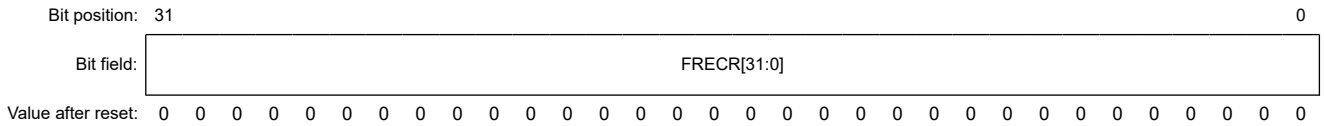
Note: S-TYPE3, P-TYPE3

The CEFCR register is a counter that indicates the number of received frames in which a CRC error was detected. When the register value becomes 0xFFFF_FFFF, the counter stops. Writing any value to the CEFCR register clears the counter value to 0.

27.2.22 FRECR : Frame Receive Error Counter Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0xE8



Bit	Symbol	Function	R/W
31:0	FRECR[31:0]	Frame Receive Error Counter See the following.	R/W

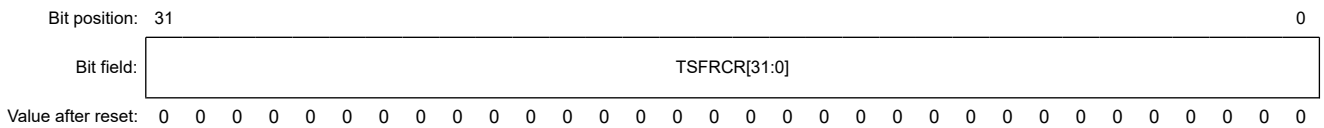
Note: S-TYPE3, P-TYPE3

The FRECR register is a counter that indicates the number of times a frame receive error has occurred. The PHY-LSI notifies the ETHERC of the frame receive error using the ET0_RX_ER pin. The FRECR register increments each time the ET0_RX_ER pin goes high. When the register value becomes 0xFFFF_FFFF, the counter stops. Writing any value to the FRECR register clears the counter value to 0.

27.2.23 TSFRCR : Too-Short Frame Receive Counter Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0xEC



Bit	Symbol	Function	R/W
31:0	TSFRCR[31:0]	Too-Short Frame Receive Counter See the following.	R/W

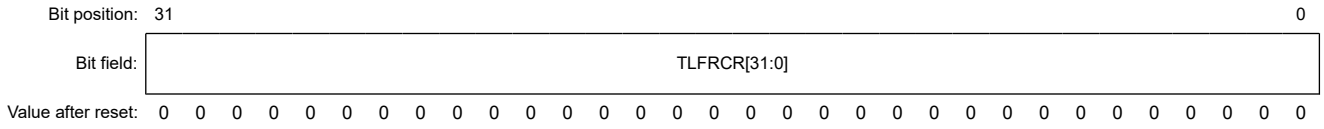
Note: S-TYPE3, P-TYPE3

The TSFRCR register is a counter that indicates the number of times a short frame that is shorter than 64 bytes was received. When the register value becomes 0xFFFF_FFFF, the counter stops. Writing any value to the TSFRCR register clears the counter value to 0.

27.2.24 TLFRCR : Too-Long Frame Receive Counter Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0xF0



Bit	Symbol	Function	R/W
31:0	TLFRCR[31:0]	Too-Long Frame Receive Counter See the following.	R/W

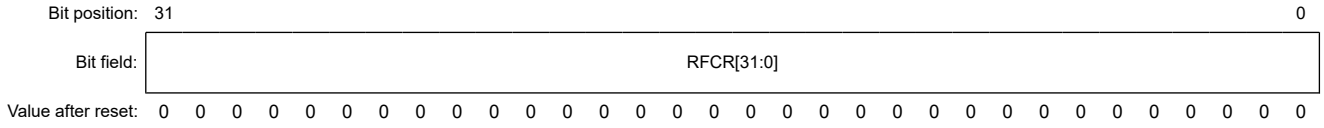
Note: S-TYPE3, P-TYPE3
Note: The TLFRCR register does not increment when a frame is received with an alignment error. In this case, the RFCR register increments.

The TLFRCR register is a counter that indicates the number of times a long frame that is longer than the RFLR register value was received. When the register value becomes 0xFFFF_FFFF, the counter stops. Writing any value to the TLFRCR register clears the counter value to 0.

27.2.25 RFCR : Received Alignment Error Frame Counter Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0xF4



Bit	Symbol	Function	R/W
31:0	RFCR[31:0]	Received Alignment Error Frame Counter See the following.	R/W

Note: S-TYPE3, P-TYPE3

The RFCR register is a counter that indicates the number of times a frame was received with an alignment error, meaning that it is not an integral number of octets. When the register value becomes 0xFFFF_FFFF, the counter stops. Writing any value to the RFCR register clears the counter value to 0.

27.2.26 MAFCR : Multicast Address Frame Receive Counter Register

Base address: ETHERC0 = 0x4035_4100
ETHERC0_NS = 0x5035_4100

Offset address: 0xF8

Bit position:	31																												0
Bit field:	MAFCR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	MAFCR[31:0]	Multicast Address Frame Receive Counter See the following.	R/W

Note: S-TYPE3, P-TYPE3

The MAFCR register is a counter that indicates the number of times a frame with the multicast address set was received. When the register value becomes 0xFFFF_FFFF, the counter stops. Writing any value to the MAFCR register clears the counter value to 0.

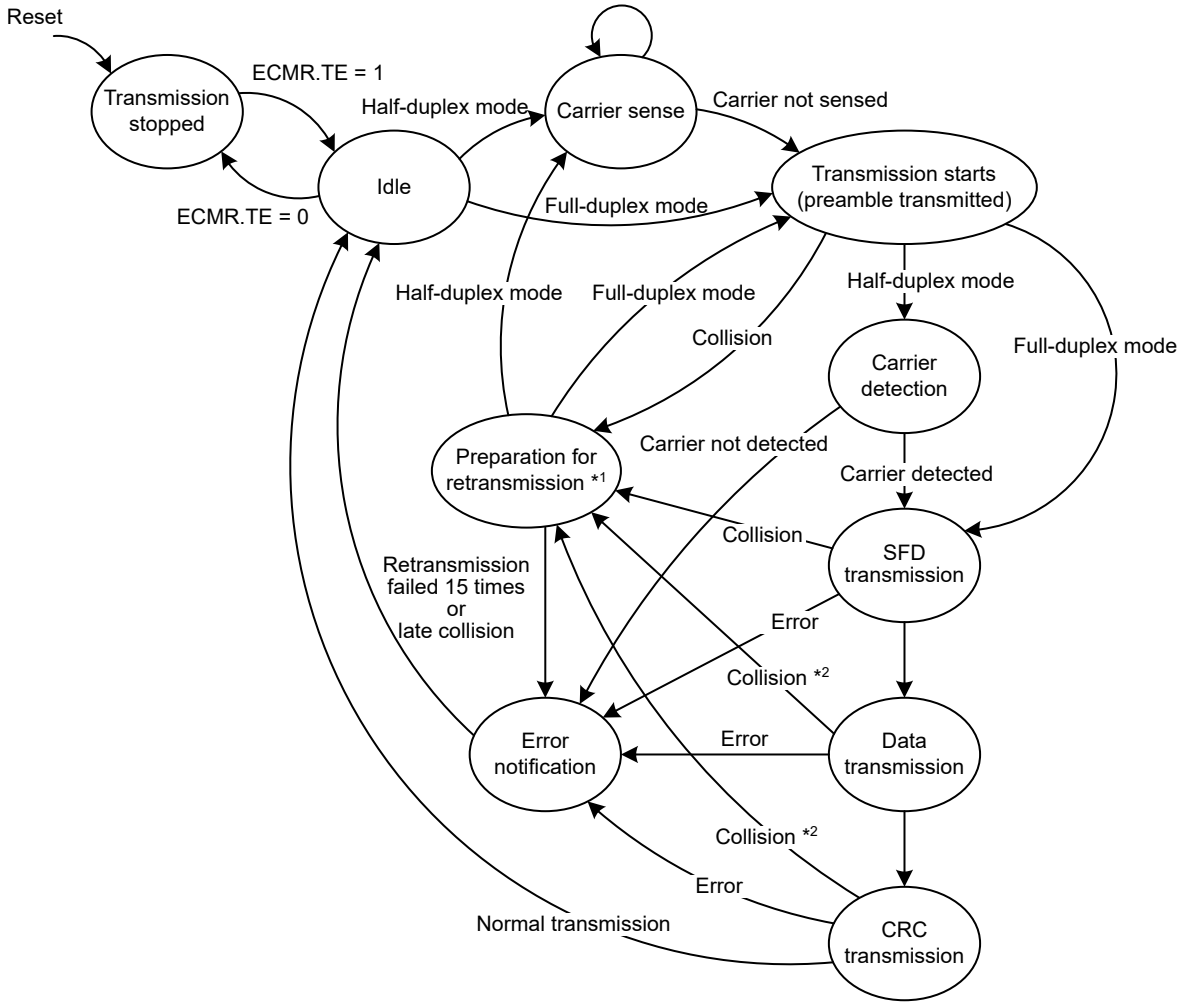
27.3 Operation

This section provides an overview of the ETHERC operations. ETHERC supports the flow control compliant with IEEE802.3x, and can transmit and receive PAUSE frames.

27.3.1 Transmission

The ETHERC transmitter assembles the transmit data into a frame and outputs it to the MII or RMII when a transmit request is received from the EDMAC. The frame transmitted through the MII or RMII is transmitted on the line by the PHY-LSI.

Figure 27.4 shows the state transitions of the ETHERC transmitter.



Note 1. Preparation for retransmission after a collision is detected includes transmitting a jam signal and adjusting transmission time intervals using the backoff algorithm.

Note 2. Preparation for retransmission is performed only when a collision is detected within 512 bit times after preamble transmission starts. When a collision is detected after 512 bit times, only a jam signal is transmitted and processing using the backoff algorithm is not performed.

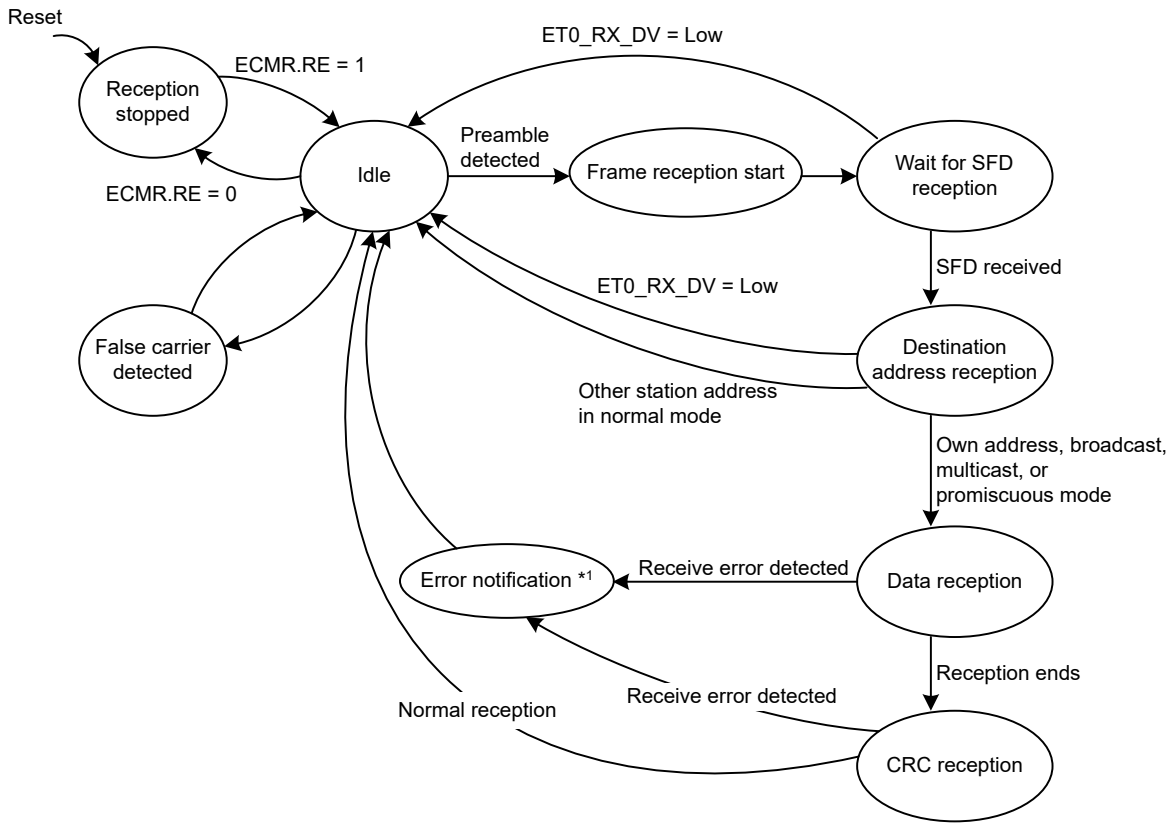
Figure 27.4 ETHERC transmitter state transitions

The ETHERC transmitter state transitions are as follows:

1. When the ECMR.TE bit is set to 1, the ETHERC enters the transmit idle state.
2. When a transmit request is received from the EDMAC, the ETHERC enters the carrier sense state. The ETHERC waits for the interpacket gap and then transmits a preamble to the MII or RMII. When full-duplex mode is selected, carrier sensing is not required, so the ETHERC transmits a preamble immediately after receiving a transmit request from the EDMAC.
3. The ETHERC transmits the Start Frame Delimiter (SFD), transmit data, and CRC sequentially. When the transmission completes successfully, the ETHERC notifies the EDMAC of successful completion, and the EDMAC sets the EDMAC0.EESR.TC flag to 1. When a late collision or loss of carrier is detected during data transmission, the ETHERC stops the transmission and notifies the EDMAC of the error.
4. After the time specified as the interpacket gap has elapsed, the ETHERC enters the idle state and continues the transmission when transmit data remains.

27.3.2 Reception

The ETHERC receiver separates the frame input from the MII or RMII into the preamble, SFD, receive data, and CRC, and transmits only the receive data (destination address, source address, type/length, data/LLC). Figure 27.5 shows the state transitions of the ETHERC receiver.



Note 1. Data in error frames is also transferred to the receive buffer.

Figure 27.5 ETHERC receiver state transitions

The ETHERC receiver state transitions are as follows:

1. When the ECMR.RE bit is set to 1, the ETHERC enters the receive idle state.
2. When the SFD following the preamble of the receive packet is detected, the ETHERC starts reception. If the received SFD is invalid, the ETHERC discards the frame.
3. In normal mode, the ETHERC starts data reception when the destination address of the receive frame is the address of the MCU or the receive frame is a broadcast or multicast frame. In promiscuous mode, the ETHERC starts data reception regardless of the receive frame type.
4. After receiving data from the MII or RMII, the ETHERC performs a CRC check. The ETHERC notifies the EDMAC of the CRC check result. After the received data is transferred to the receive buffer, the CRC check result is written back to the receive descriptor as status. The result is also reflected in the EDMAC0.EESR.CERF flag.
5. When the ECMR.RE bit is 1 after one frame is received, the ETHERC prepares to receive the next frame.

27.3.3 Frame Timing

27.3.3.1 MII frame timing

Figure 27.6 to Figure 27.11 show the MII frame timing.

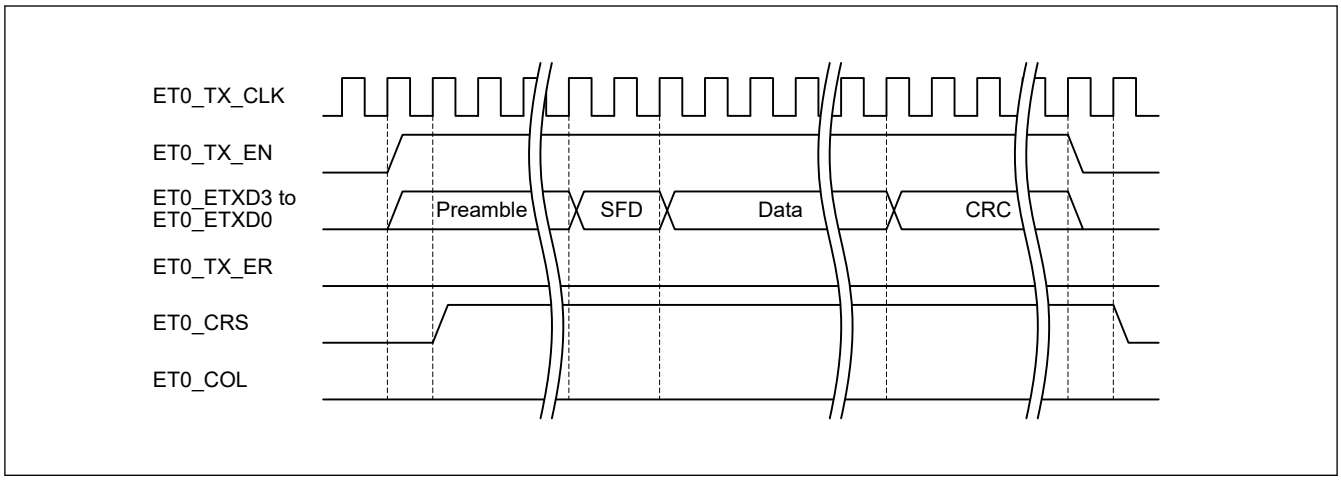


Figure 27.6 MII frame transmit timing during normal transmission

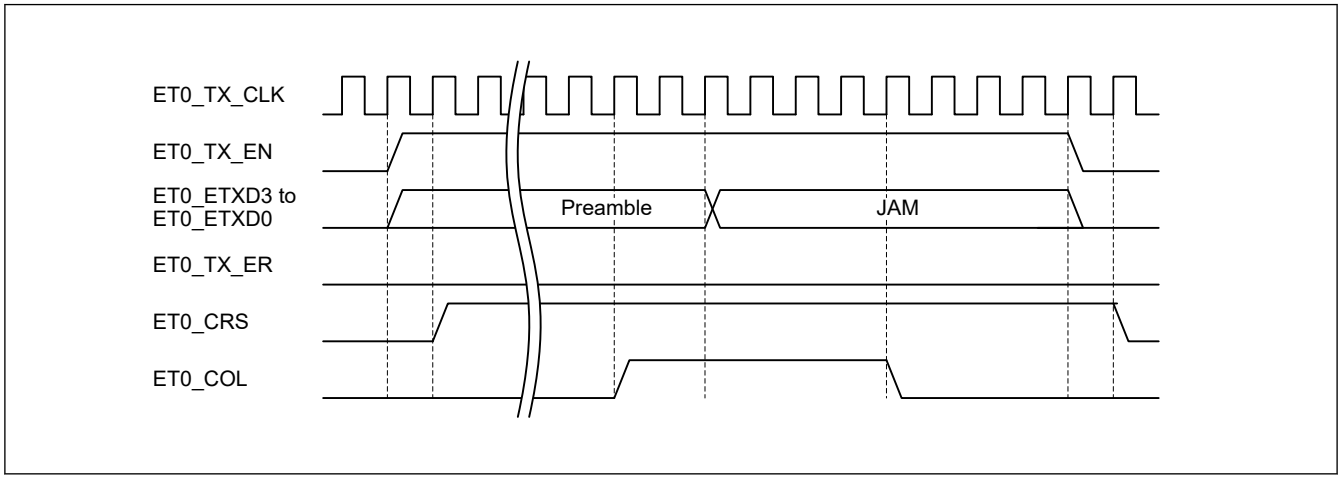


Figure 27.7 MII frame transmit timing when a collision occurs

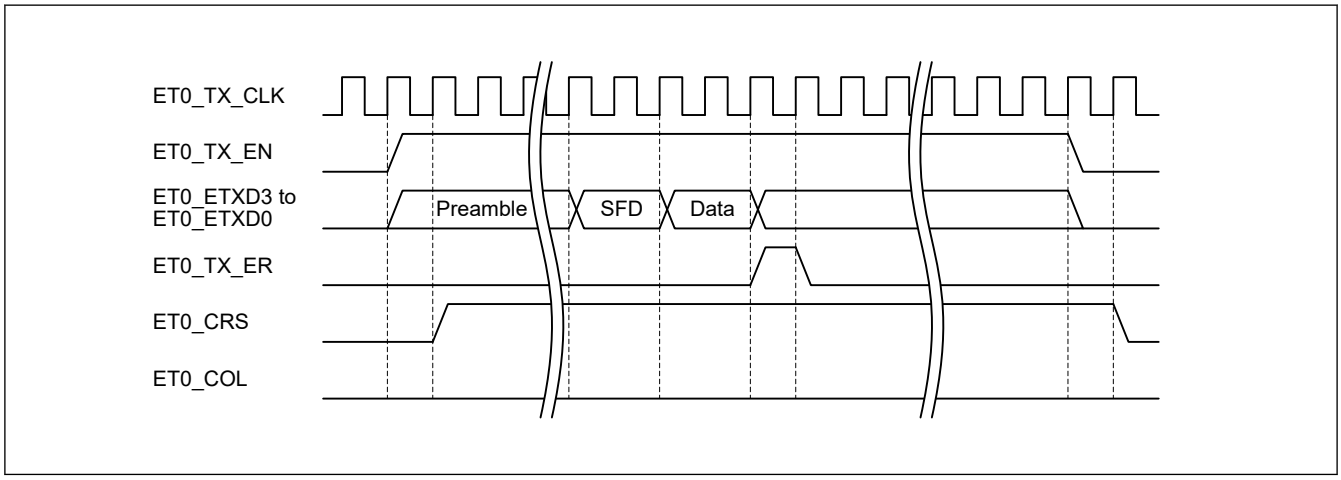


Figure 27.8 MII frame transmit timing when a transmit error occurs

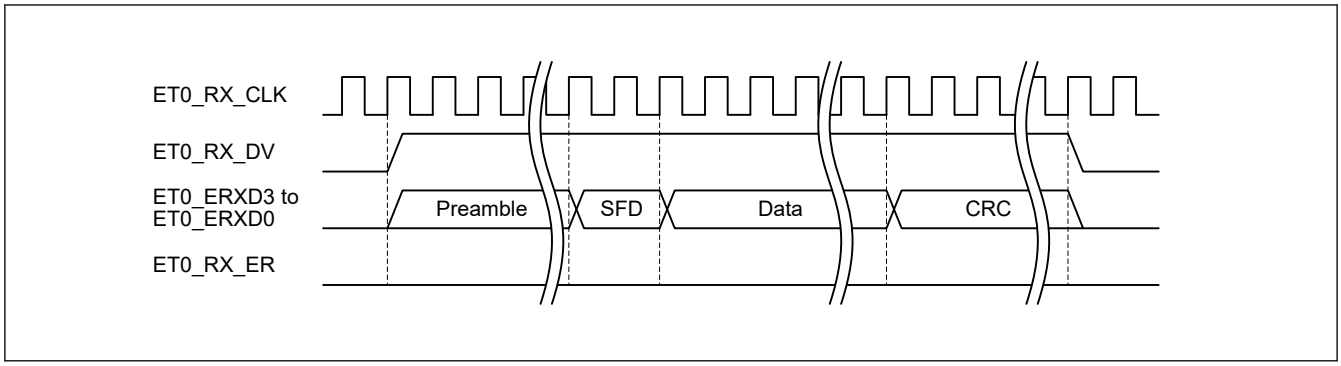


Figure 27.9 MII frame receive timing during normal reception

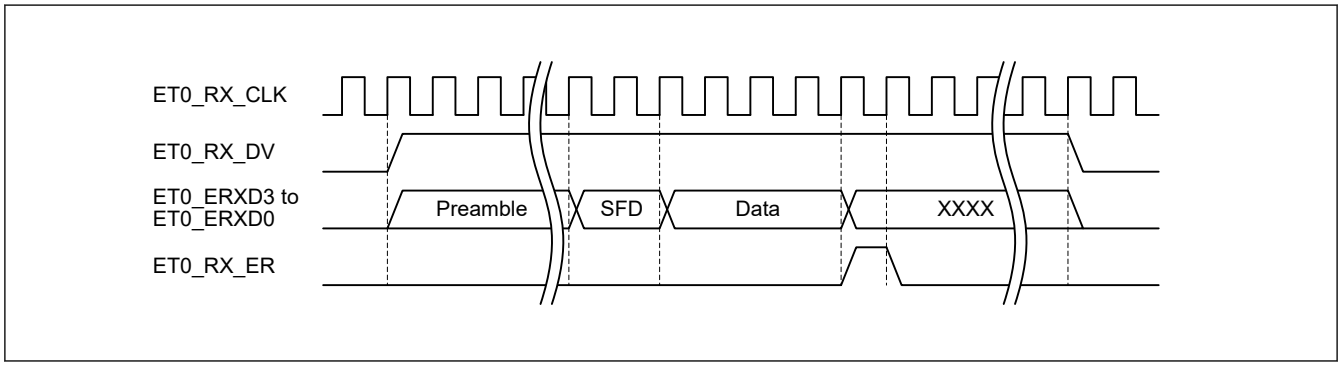


Figure 27.10 MII frame receive timing for receive error notification

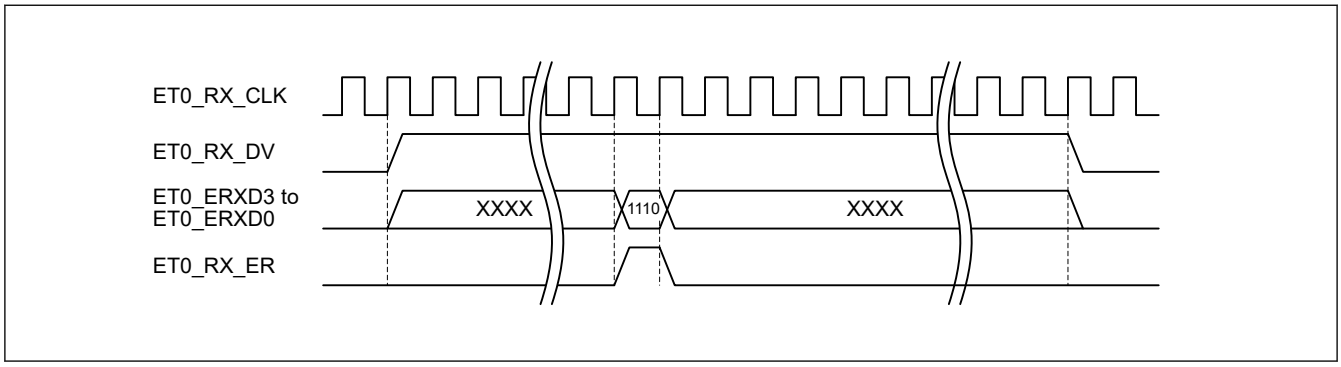


Figure 27.11 MII frame receive timing for false carrier notification

27.3.3.2 RMII frame timing

Figure 27.12 to Figure 27.14 show the RMII frame timing.

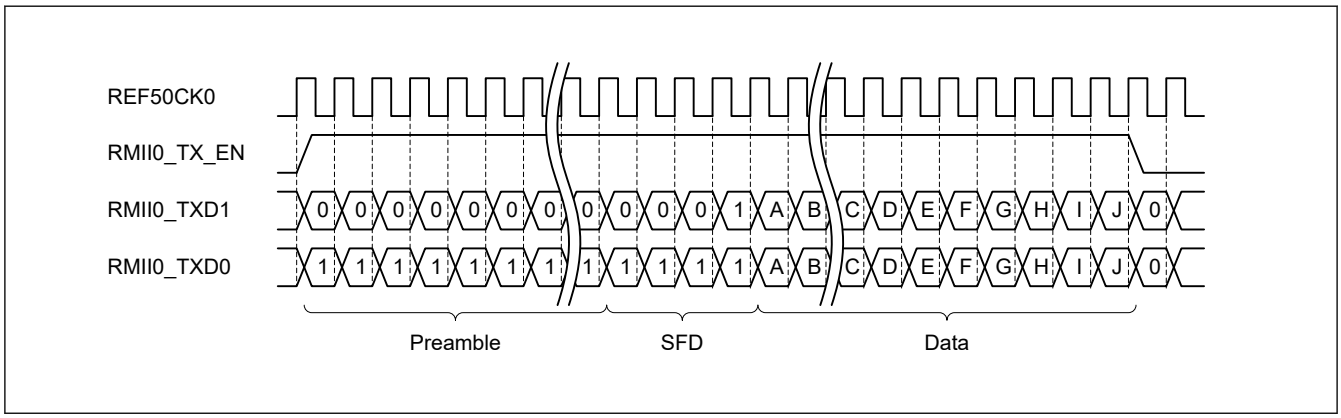


Figure 27.12 RMII frame transmit timing during normal transmission

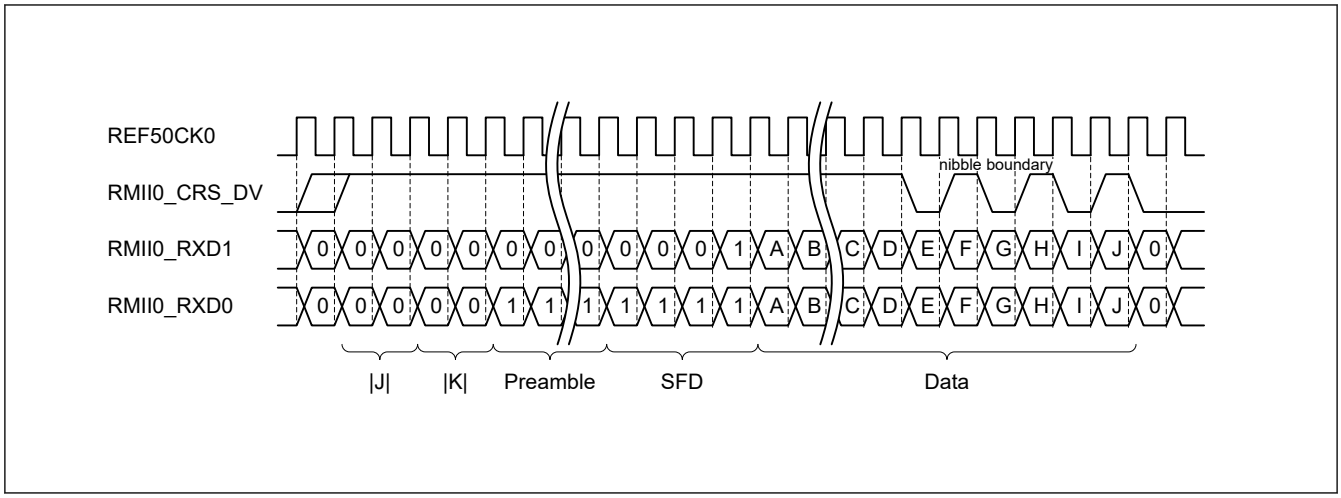


Figure 27.13 RMII frame receive timing during normal reception

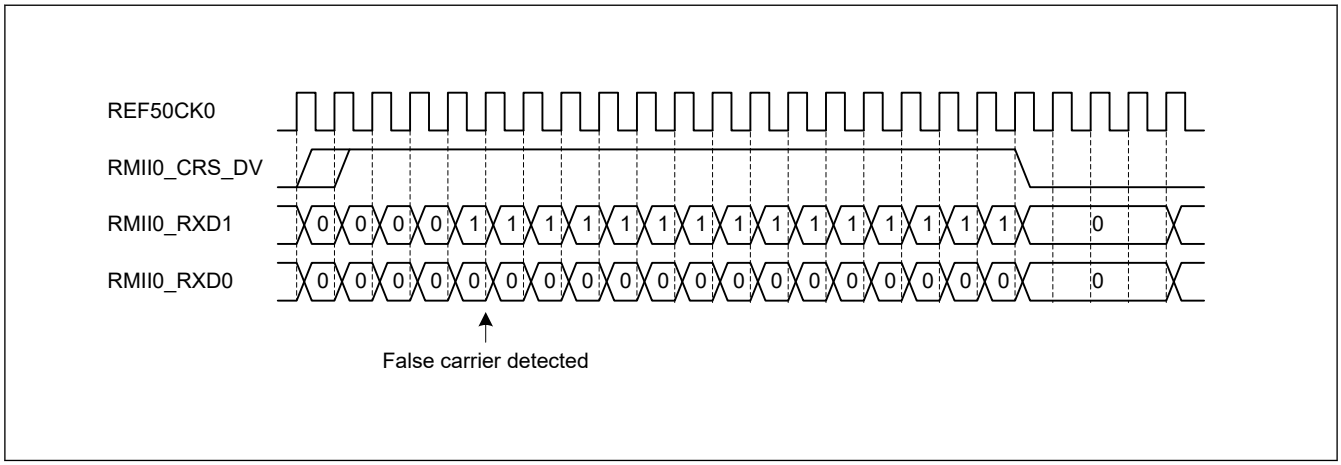


Figure 27.14 RMII frame receive timing when a false carrier is detected

27.3.4 Accessing the MII and RMII Registers

Use the PIR register to access the MII and RMII registers in the PHY-LSI. Serial data in the MII and RMII management frame format is transmitted and received through the ET0_MDC and ET0_MDIO pins controlled by software.

27.3.4.1 MII and RMII management frame format

Table 27.3 lists the MII and RMII management frame formats.

Table 27.3 MII and RMII management frame formats

Access type	MII and RMII management frame							
Parameter	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
Number of bits	32	2	2	5	5	2	16	1
Read	1…1	01	10	00001	RRRRR	Z0	DDDDDDDD DDDDDDDD	Z
Write	1…1	01	01	00001	RRRRR	10	DDDDDDDD DDDDDDDD	Z

Note: PRE (preamble): Send 32 consecutive 1s.
ST (start of frame): Send 01b.
OP (operation code): Send 10b for read or 01b for write.
PHYAD (PHY address): Up to 32 PHY-LSIs can be connected to one MAC. PHY-LSIs are selected with these 5 bits. When the PHY-LSI address is 1, send 0x01.

REGAD (register address): One register is selected from up to 32 registers in the PHY-LSI. When the register address is 1, send 0x01.
TA (turnaround): Use 2-bit turnaround time to avoid contention between the register address and data during a read operation. Send 10b during a write operation. Release the bus for 1 bit during a read operation (Z is output).
(This is indicated as Z0 because 0 is output from the PHY-LSI on the next clock cycle.)
DATA (data): 16-bit data. Sequentially send or receive starting from the MSB.
IDLE (IDLE condition): Wait time before inputting the next MII or RMII management format. Release the bus during a write operation (Z is output). No control is required, because a bus was already released during a read operation.

27.3.4.2 MII and RMII register access procedure

Access to the MII and RMII registers includes writing data in 1-bit units, reading data in 1-bit units, and releasing the bus. Figure 27.15 to Figure 27.18 show examples of the MII and RMII register access timing. The access timing differs with the PHY-LSI type.

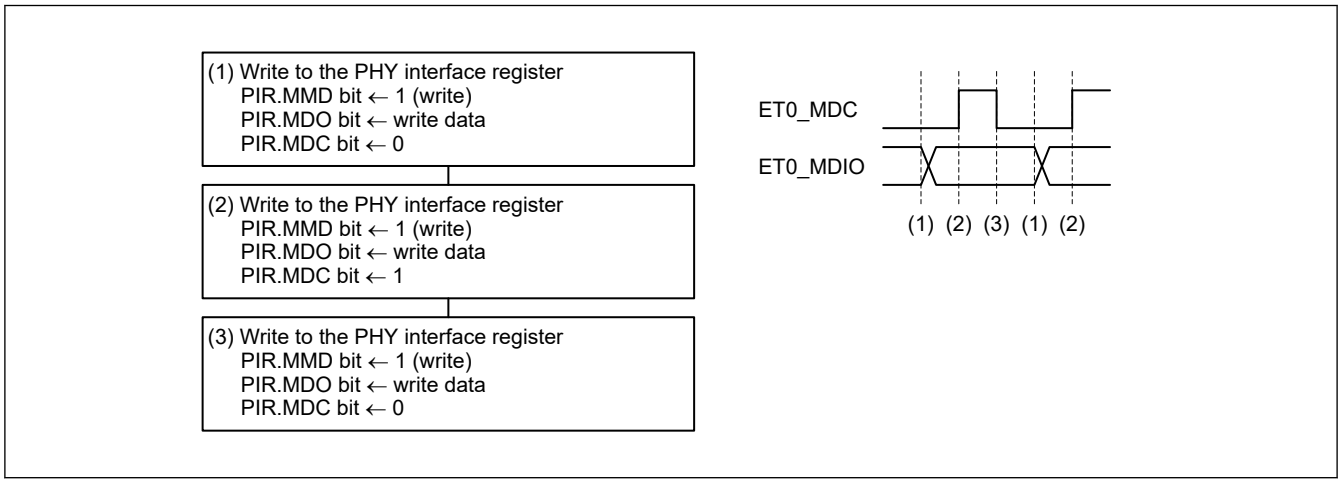


Figure 27.15 1-bit data write flow

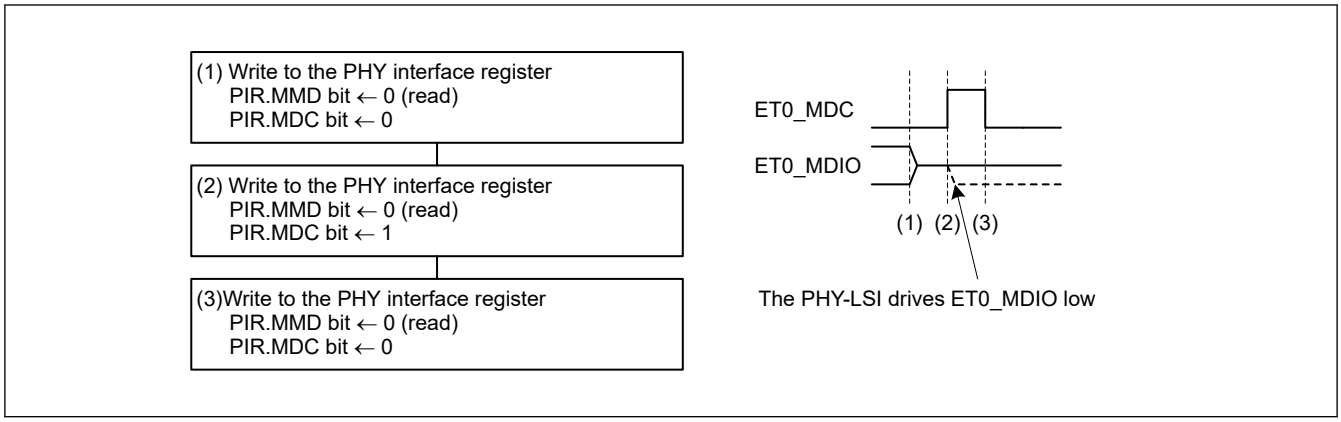


Figure 27.16 Bus release flow, with TA in read operation in Table 27.3

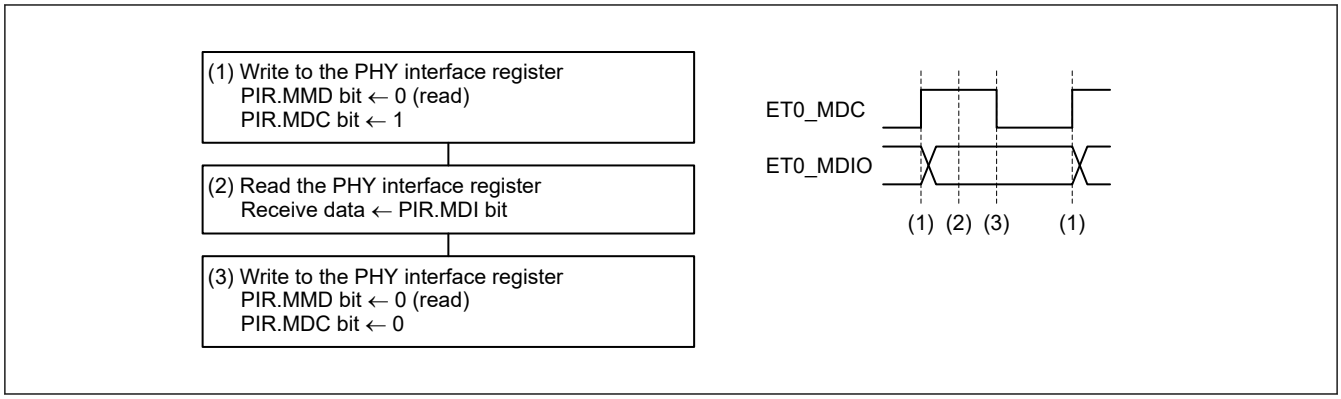


Figure 27.17 1-bit data read flow

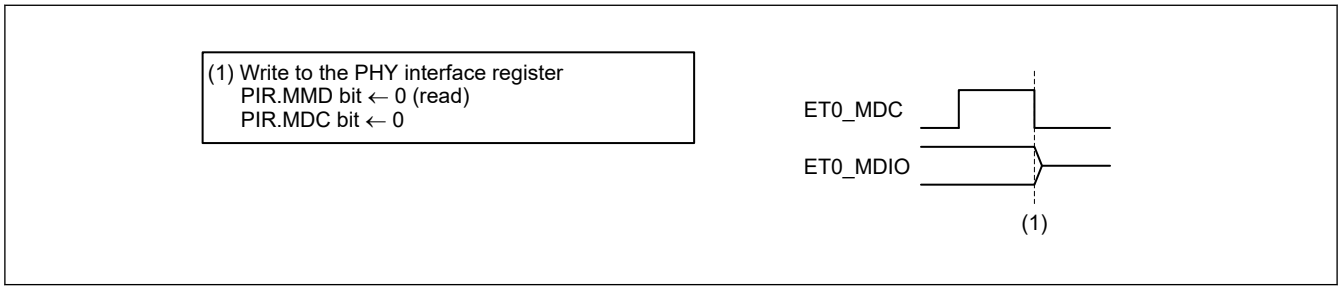


Figure 27.18 Bus release flow, with IDLE in write operation in Table 27.3

27.3.5 Magic Packet Detection

The ETHERC supports Wake-on-LAN (WOL). WOL is a function to detect a Magic Packet transmitted from a host device or other device, and to wake the MCU from a low power mode such as Sleep. When the ETHERC detects a Magic Packet, it outputs high on the ET0_WOL pin. Write 1 to the EDMAC0.EDMR.SWR bit to drive the ET0_WOL pin low.

Because a Magic Packet is transmitted in broadcast mode, it is received regardless of the destination MAC address selected in the format. The ETHERC outputs high on the ET0_WOL pin only when the destination MAC address matches its own MAC address. See the technical documentation provided by Advanced Micro Devices, Inc., for details on the Magic Packet.

To use WOL in the MCU, use the procedure in the following example:

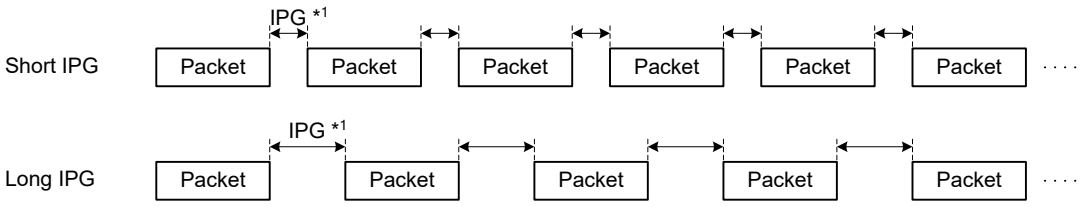
1. Configure the ICU to disable ETHER_EINT0 interrupt requests.
2. Set the ECMR.MPDE bit to 1 to enable Magic Packet detection, and set the ECMR.RE bit to 1 to enable reception.
3. Set the ECSIPR.MPDIP bit to 1 to enable notification of Magic Packet detection interrupts.
4. Set the EDMAC0.EESIPR.ECIIP bit to 1 to enable ETHERC status register source interrupts.
5. Configure the ICU to enable ETHER_EINT0 interrupt requests.
6. Change the CPU operating mode to CPU Sleep mode or place unused peripherals in the module-stop state, as required.
7. When a Magic Packet is detected, an interrupt request is sent to the CPU. High is output on the ET0_WOL pin to notify peripheral devices that the Magic Packet was detected.

27.3.5.1 Constraints on Magic Packet detection

The ETHERC receives packets, including broadcast packets, even when waiting to receive a Magic Packet. This means that receive data might already be stored in the receive FIFO of the EDMAC when a Magic Packet is detected. Also, flags in the ECSR and EDMAC0.EESR registers might have changed. When returning to normal operation after detecting a Magic Packet, set the EDMAC0.EDMR.SWR bit to 1 to reset the ETHERC and EDMAC.

27.3.6 Adjusting Transmission Efficiency by Changing the IPG

The IPG is a non-transmit period between transmit frames. The ETHERC can change the value of the IPG to increase or decrease transmission efficiency based on the value set in the IPGR register. Typical values are specified in the IEEE802.3 standard. When changing the setting, confirm that all devices in the same network operate normally.



Note 1. The IPG may be longer than the set value depending on the condition of the line or system bus.

Figure 27.19 Differences in transmission efficiency based on changes in the IPG

27.3.7 Flow Control

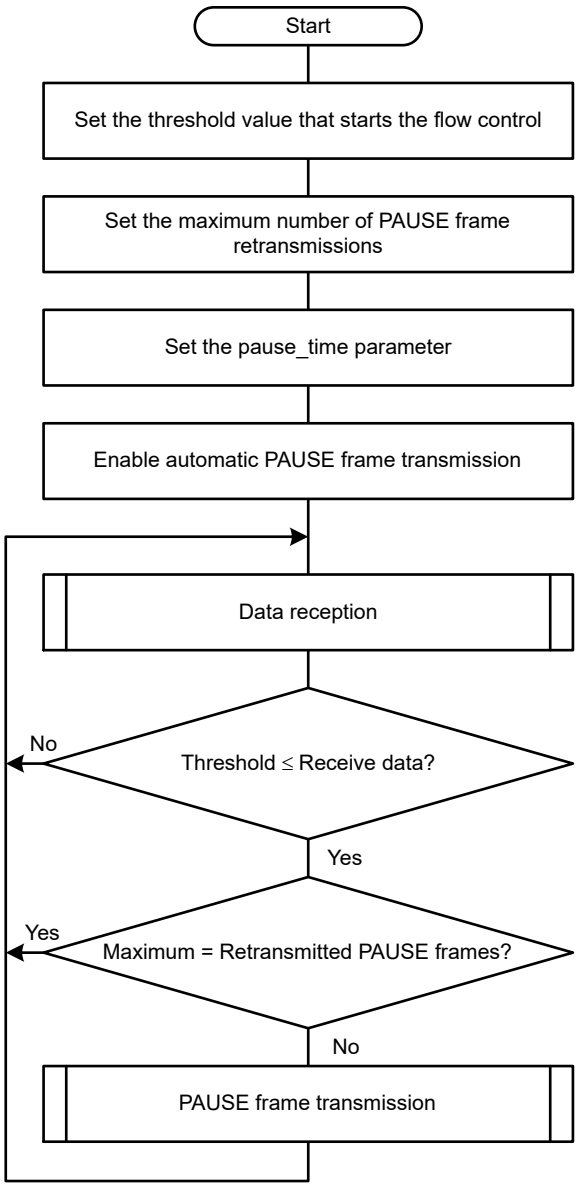
The ETHERC can perform flow control compliant with IEEE802.3x in full-duplex mode, and the receiver and transmitter can be set independently. PAUSE frames can be transmitted automatically or manually.

27.3.7.1 Automatic PAUSE frame transmission

When the ECMR.TXF bit is set to 1, automatic PAUSE frame transmission is enabled. A PAUSE frame is automatically transmitted by a PAUSE frame transmit request from the EDMAC. The APR.AP[15:0] bit value is used for the pause_time parameter of the PAUSE frame.

When a PAUSE frame is transmitted, if the EDMAC is still requesting PAUSE frame transmission after the PAUSE time elapses, a PAUSE frame is transmitted again. The maximum number of PAUSE frame retransmissions can be set in the TPAUSER.TPAUSE[15:0] bits. If the maximum number of retransmissions is reached, subsequent PAUSE frames are not transmitted.

Figure 27.20 shows the procedure for setting up automatic PAUSE frame transmission.



1. Set the threshold value in the EDMAC0.FCFTR.RFDO[2:0] bits or EDMAC0.FCFTR.RFFO[2:0] bits.
2. Set the maximum number of automatic PAUSE frame retransmissions in the TPAUSER.TPAUSE[15:0] bits.
3. Set the pause_time parameter of the automatic PAUSE frame in the APR.AP[15:0] bits.
4. Set the ECMR.TXF bit to 1.
5. Set the ECMR.RE bit to 1.
6. Check whether the amount of receive FIFO data or the number of received frames reached the threshold.
7. Check whether the number of automatic PAUSE frame retransmissions reached the threshold.

Figure 27.20 Example procedure for setting up automatic PAUSE frame transmission

27.3.7.2 Manual PAUSE frame transmission

A PAUSE frame can be manually transmitted at any time. When the software writes the pause_time parameter of the PAUSE frame to the MPR.MP[15:0] bits, the ETHERC transmits a PAUSE frame once. To transmit a PAUSE frame more than once, write to the MPR.MP[15:0] bits for each transmission.

27.3.7.3 PAUSE frame reception

When the ECMR.RXF bit is set to 1, PAUSE frame detection is enabled. After a PAUSE frame is received, the ETHERC completes transmission of the current frame and waits for the PAUSE time of the received PAUSE frame to elapse before it can transmit the next frame. The ETHERC also increments the RFCF.RPAUSE[7:0] bit value.

However, while waiting for the PAUSE time to elapse, if a PAUSE frame that contains a pause_time parameter of 0 is received and the ECMR.ZPF bit is 1, the ETHERC becomes ready to transmit immediately.

27.4 Interrupts

When a flag in the ECSR register sets to 1 and the associated bit in the ECSIPR register is 1, the ETHERC notifies the EDMAC of the interrupt source status. After receiving the notification, the EDMAC sets the EDMAC0.EESR.ECI flag to 1. When the EDMAC0.EESIPR.ECIIP bit is 1, the EDMAC sends an ETHER_EINT0 interrupt request to the CPU.

For details, see section 28, Ethernet DMA Controller (EDMAC).

27.5 Usage Notes

27.5.1 Preventing the LCHNG Flag from Erroneously Setting to 1

The ECSR.LCHNG flag might set to 1 even when the input level of the ET0_LINKSTA pin remains the same. In this case, high level is input to the ET0_LINKSTA pin when setting the PFS.PmnPFS register to assign the ET0_LINKSTA signal to a port, or when releasing the ETHERC and EDMAC software reset using the EDMAC0.EDMR.SWR bit. The ECSR.LCHNG flag is sets to 1 because the ET0_LINKSTA signal in the ETHERC is fixed low level regardless of the input level to the external pin if the MPC does not assign the ET0_LINKSTA signal or during an ETHERC and EDMAC software reset.

To avoid erroneously generating a link signal change interrupt, clear the ECSR.LCHNG flag, and then set the ECSIPR.LCHNGIP bit to 1.

27.5.2 Input to RMII0_RX_ER Pin While RMII Is Selected

When the width of a reception error signal received from the PHY-LSI is only 1 cycle of the REF50CK0 clock (50 MHz) while the RMII is selected, the signal is not recognized as an error signal.

27.5.3 Collision Occurrence in Half-Duplex Mode

Transmission might start and communication might collide within 21 clock cycles (50 MHz) from reception in halfduplex mode.

28. Ethernet DMA Controller (EDMAC)

28.1 Overview

The MCU provides one channel for the Ethernet DMA Controller (EDMAC) for the Ethernet Controller (ETHERC) .

The EDMAC controls most of the transmit and receive buffer management for communications. This reduces the load on the CPU and allows efficient data transmission and reception. The data transfers are controlled according to the information referred to as descriptors, in memory. EDMAC0 controls data transmission and reception for ETHERC0.

Table 28.1 lists the EDMAC specifications and Figure 28.1 shows the configuration. Figure 28.2 shows the configuration of descriptors and the transmit and receive buffers in memory.

Table 28.1 EDMAC specifications

Parameter	Specifications
Data transmission and reception	• Controls data transmission and reception according to descriptors • Supports single buffer frame transmission and reception (1 buffer per frame) and multi-buffer frame transmission and reception (multiple buffers per frame)
Functions	• Minimizes system bus occupancy time using block transfer (32-byte units) • Writes back the transmit or receive frame state to descriptors • Inserts padding in receive data
Module-stop function	Module-stop state can be set to reduce power consumption
Trust Zone Filter	Security and Privilege attribution can be set

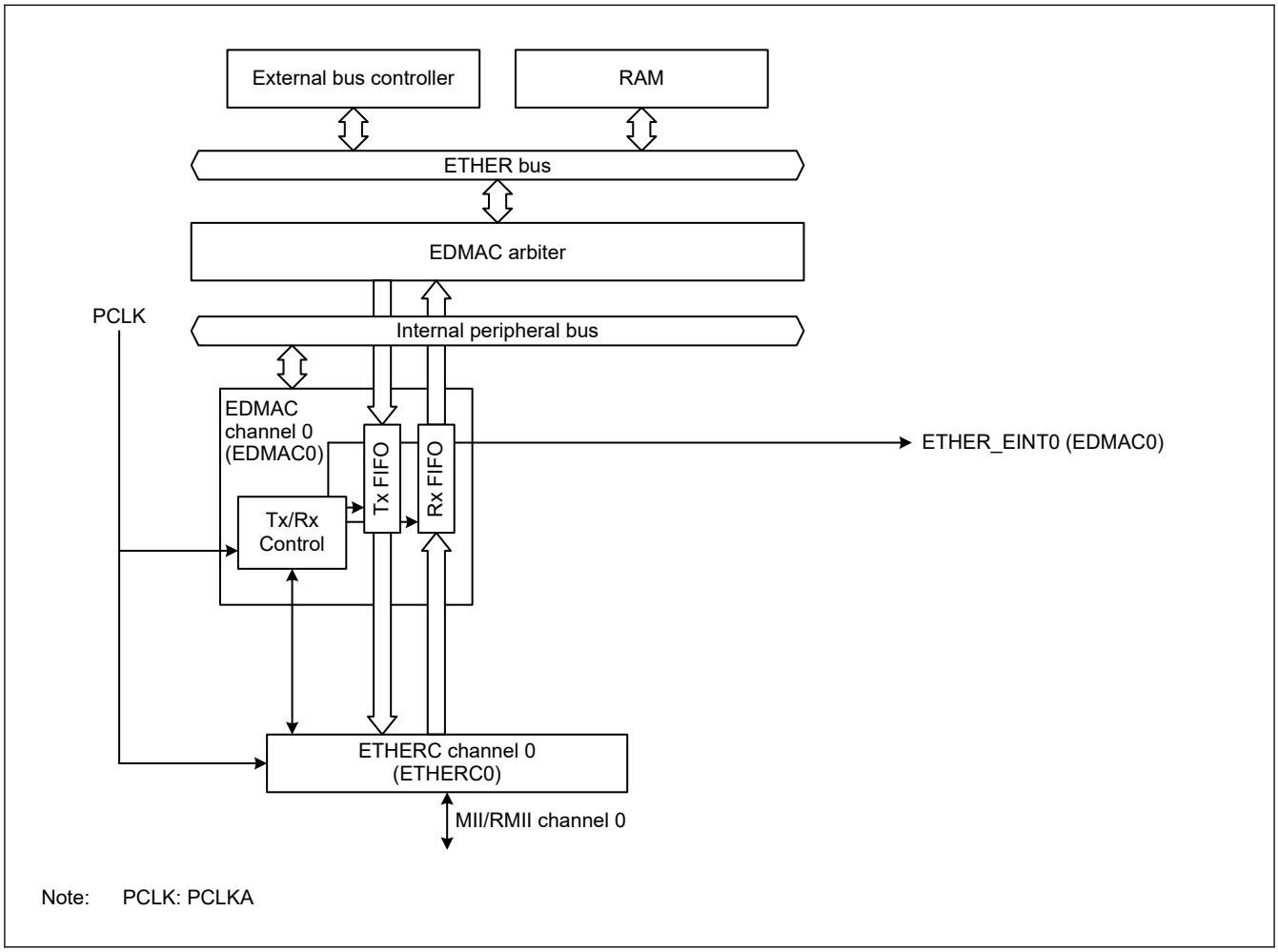


Figure 28.1 EDMAC configuration

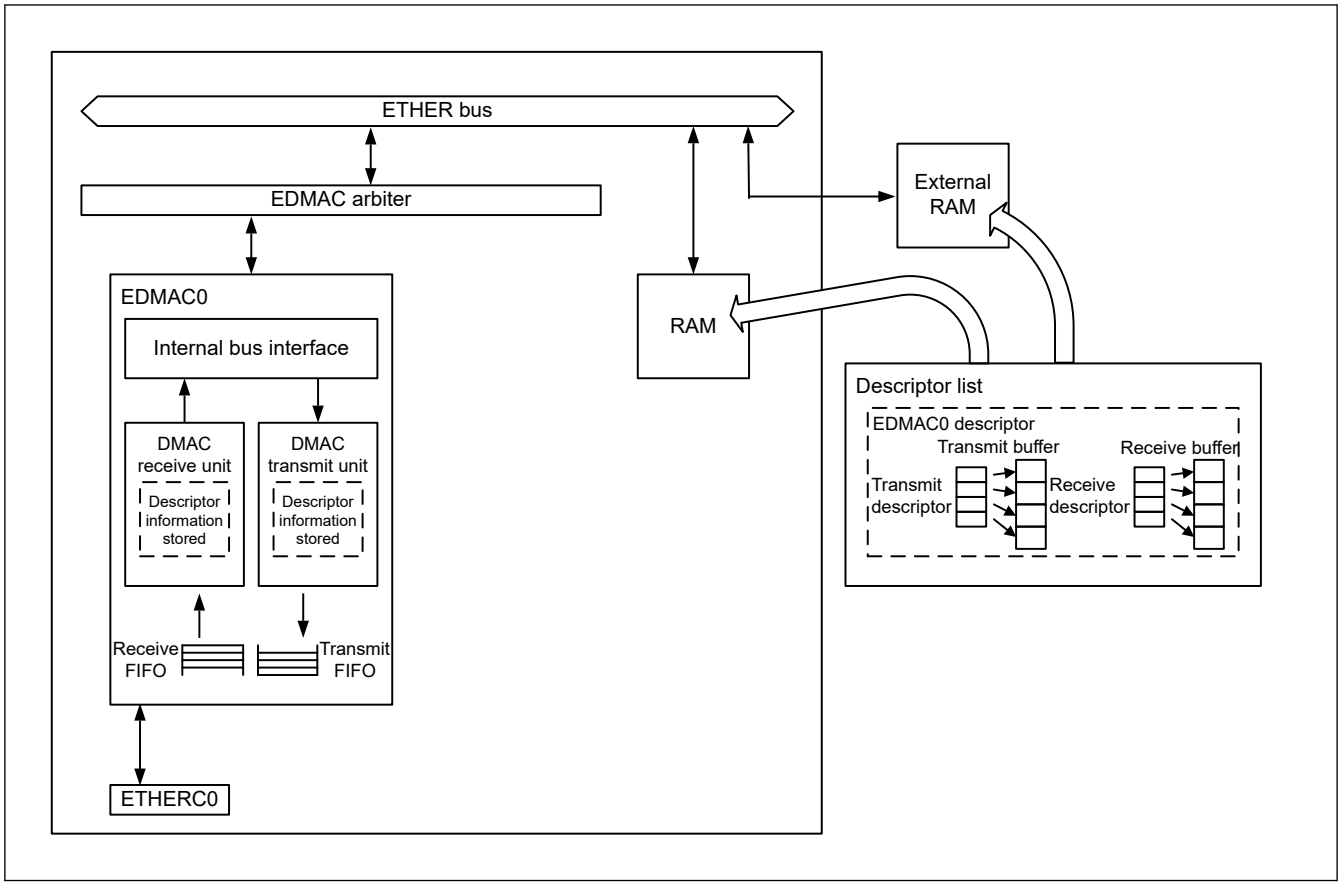


Figure 28.2 Configuration of descriptors and transmit and receive buffers in memory

28.2 Register Descriptions

28.2.1 EDMR : EDMAC Mode Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	DE	DL[1:0]		—	—	—	SWR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SWR	Software Reset When 1 is written, the associated channels of the EDMAC and ETHERC are reset. The TDLAR, RDLAR, RMFCR, TFUCR, and RFOCR registers are not reset with this bit.The read value is 0.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
5:4	DL[1:0]	Transmit/Receive Descriptor Length 0 0: 16 bytes 0 1: 32 bytes 1 0: 64 bytes 1 1: 16 bytes.	R/W
6	DE	Big Endian Mode/Little Endian Mode*1 0: Big endian mode 1: Little endian mode.	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This setting applies to data for the transmit and receive buffers. It does not apply to transmit and receive descriptors and registers.

The EDMR register controls EDMAC operation. Set the EDMR register during initialization process after a reset. When rewriting this register outside of the initialization process, set the SWR bit to 1 to reset the EDMAC and ETHERC, then set this register again. If the ETHERC and EDMAC are reset during data transmission or reception, abnormal data might be sent on the line. Do not rewrite this register while the ETHERC transmit or receive function is enabled. It takes 64 cycles of the peripheral module clock (PCLKA) to initialize the ETHERC and EDMAC. Complete the initialization before accessing registers in the ETHERC and EDMAC.

28.2.2 EDTRR : EDMAC Transmit Request Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TR	Transmit Request When 1 is written, the EDMAC reads the associated descriptor and transmits frames where the TD0.TACT bit is 1. The TR bit clears to 0 after all the valid frames are transmitted. Writing 0 to this bit has no effect.	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The EDTRR register controls EDMAC transmission. After the EDMAC transmits one frame, it reads the next descriptor. When the TD0.TACT bit in the descriptor is 1, the EDMAC continues transmission. When the TD0.TACT bit is 0, the EDMAC sets the TR bit to 0 and stops transmission.

28.2.3 EDRRR : EDMAC Receive Request Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RR	Receive Request 0: Disable the receive function*1 1: Read receive descriptor and enable the receive function.	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

Note 1. If the receive function is disabled during frame reception, write-back to the receive descriptor is not performed successfully. Subsequent pointers for reading a receive descriptor become abnormal and the EDMAC cannot operate normally. In this case, to enable the EDMAC receive function again, execute a software reset by setting the EDMR.SWR bit to 1. To disable the EDMAC receive function without resetting the EDMAC, set the ETHERC0.ECMR.RE bit to 0. After the EDMAC completes reception and write-back to the receive descriptor is confirmed, set the RR bit to 0.

The EDRRR register controls EDMAC reception. When the RR bit sets to 1, the EDMAC reads the receive descriptor.

When the RD0.RACT bit is 1, the EDMAC waits for a receive request from the ETHERC. When the EDMAC has received data for the receive buffer size, it reads the next descriptor and waits to receive a frame. If the RD0.RACT bit is 0, the EDMAC sets the RR bit to 0 and stops reception.

28.2.4 TDLAR : Transmit Descriptor List Start Address Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0x18

Bit position:	31	0
Bit field:		
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	—	These bits specify the start address of the transmit descriptor list. Set the start address according to the descriptor length selected in the EDMR.DL[1:0] bits. ● 16-byte boundary: Lower 4 bits = 0000b ● 32-byte boundary: Lower 5 bits = 00000b ● 64-byte boundary: Lower 6 bits = 000000b.	R/W

Note: S-TYPE3, P-TYPE3

The TDLAR register specifies the start address of the transmit descriptor list. Align each descriptor on the associated boundary to the descriptor length selected in the EDMR.DL[1:0] bits. Do not rewrite the TDLAR register during transmission. Rewrite the TDLAR register while the EDTRR.TR bit is 0.

28.2.5 RDLAR : Receive Descriptor List Start Address Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0x20

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	—	The start address of the receive descriptor list is set. Set the start address according to the descriptor length selected in the EDMR.DL[1:0] bits. ● 16-byte boundary: Lower 4 bits = 0000b ● 32-byte boundary: Lower 5 bits = 00000b ● 64-byte boundary: Lower 6 bits = 000000b.	R/W

Note: S-TYPE3, P-TYPE3

The RDLAR register specifies the start address of the receive descriptor list. Allocate each descriptor on the associated boundary to the descriptor length selected in the EDMR.DL[1:0] bits. Do not rewrite the RDLAR register during reception. Rewrite the RDLAR register while the EDRRR.RR bit is 0.

28.2.6 EESR : ETHERC/EDMAC Status Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0x28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	TWB	—	—	—	TABT	RABT	RFCOF	ADE	ECI	TC	TDE	TFUF	FR	RDE	RFOF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CND	DLC	CD	TRO	RMAF	—	—	RRF	RTLF	RTSF	PRE	CERF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CERF	CRC Error Flag 0: CRC error not detected 1: CRC error detected.	R/W
1	PRE	PHY-LSI Receive Error Flag 0: PHY-LSI receive error not detected 1: PHY-LSI receive error detected.	R/W
2	RTSF	Frame-Too-Short Error Flag 0: Frame-too-short error not detected 1: Frame-too-short error detected.	R/W
3	RTLF	Frame-Too-Long Error Flag 0: Frame-too-long error not detected 1: Frame-too-long error detected.	R/W
4	RRF	Alignment Error Flag 0: Alignment error not detected 1: Alignment error detected.	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	RMAF	Multicast Address Frame Receive Flag 0: Multicast address frame not received 1: Multicast address frame received.	R/W
8	TRO	Transmit Retry Over Flag 0: Transmit retry-over condition not detected 1: Transmit retry-over condition detected.	R/W
9	CD	Late Collision Detect Flag 0: Late collision not detected 1: Late collision detected during frame transmission.	R/W
10	DLC	Loss of Carrier Detect Flag 0: Loss of carrier not detected 1: Loss of carrier detected during frame transmission.	R/W
11	CND	Carrier Not Detect Flag 0: Carrier detected when transmission started 1: Carrier not detected during preamble transmission.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	RFOF	Receive FIFO Overflow Flag 0: No overflow occurred 1: Overflow occurred.	R/W
17	RDE	Receive Descriptor Empty Flag 0: EDMAC detected that the receive descriptor valid bit (RD0.RACT) is 1 1: EDMAC detected that the receive descriptor valid bit (RD0.RACT) is 0.	R/W
18	FR	Frame Receive Flag 0: Frame not received 1: Frame received and update of the receive descriptor is complete.	R/W
19	TFUF	Transmit FIFO Underflow Flag 0: No underflow occurred 1: Underflow occurred.	R/W
20	TDE	Transmit Descriptor Empty Flag 0: EDMAC detected that the transmit descriptor valid bit (TD0.TACT) is 1 1: EDMAC detected that the transmit descriptor valid bit (TD0.TACT) is 0.	R/W
21	TC	Frame Transfer Complete Flag 0: Transfer not complete or no transfer requested 1: All frames indicated in the transmit descriptor were completely transferred to the transmit FIFO.	R/W
22	ECI	ETHERC Status Register Source Flag 0: ETHERC status interrupt source not detected 1: ETHERC status interrupt source detected.	R*1
23	ADE	Address Error Flag 0: Invalid memory address not detected (normal operation) 1: Invalid memory address detected.*2	R/W
24	RFCOF	Receive Frame Counter Overflow Flag 0: Receive frame counter did not overflow 1: Receive frame counter overflowed.	R/W
25	RABT	Receive Abort Detect Flag 0: Frame reception not aborted or no reception requested 1: Frame reception aborted.	R/W
26	TABT	Transmit Abort Detect Flag 0: Frame transmission not aborted or no transmission requested. 1: Frame transmission aborted.	R/W
29:27	—	These bits are read as 0. The write value should be 0.	R/W
30	TWB	Write-Back Complete Flag 0: Write-back not complete or no transmission requested 1: Write-back to the transmit descriptor completed.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. The ECI flag is read-only. When the source in the ECSR register is cleared, the ECI flag is also cleared.
Note 2. When an address error is detected, the EDMAC halts the process. To resume operation, set the EDMR.SWR bit to 1 (resetting the EDMAC and ETHERC), and then reconfigure the EDMAC and ETHERC.

The EESR register indicates the ETHERC and EDMAC communication status. Each flag in the EESR register can be output as an interrupt request signal (ETHER_EINT0) from the EDMAC. Writing 1 clears all of the flags except ECI to 0. Writing 0 does not affect any of the flag values. The interrupt sources are enabled by setting the associated bits in the EESIPR register.

CERF flag (CRC Error Flag)

The CERF flag sets to 1 when an error is detected while checking the frame check sequence (FCS) field of the receive frame.

PRE flag (PHY-LSI Receive Error Flag)

The PRE flag indicates that the RX_ER signal output from the PHY-LSI is high.

RTSF flag (Frame-Too-Short Error Flag)

The RTSF flag indicates that a received frame is less than 64 bytes.

RTLF flag (Frame-Too-Long Error Flag)

The RTLF flag indicates that a received frame is greater than the upper limit of the receive frame length set in the ETHERC0.RFLR register. The excess data is discarded.

RRF flag (Alignment Error Flag)

The RRF flag indicates that a frame is not an integral number of octets. The last word that is not an integral number of octets is not transferred.

RMAF flag (Multicast Address Frame Receive Flag)

The RMAF flag indicates that a multicast frame was received.

TRO flag (Transmit Retry Over Flag)

The TRO flag indicates that a collision occurred again during the 15th retry of frame transmission.

CD flag (Late Collision Detect Flag)

The CD flag indicates that a late collision was detected during frame transmission.

DLC flag (Loss of Carrier Detect Flag)

The DLC flag indicates that a loss of carrier was detected during frame transmission.

CND flag (Carrier Not Detect Flag)

The CND flag sets to 1 when a carrier is not detected during preamble transmission.

RFOF flag (Receive FIFO Overflow Flag)

The RFOF flag indicates that the receive FIFO overflowed during frame reception.

RDE flag (Receive Descriptor Empty Flag)

The RDE flag indicates that the read receive descriptor is invalid. When this flag sets to 1, set the RD0.RACT bit in the receive descriptor to 1 and set the EDRRR.RR bit to 1 to resume reception.

FR flag (Frame Receive Flag)

The FR flag indicates that a frame was received and the receive descriptor was updated. The FR flag sets to 1 every time a frame is received.

TFUF flag (Transmit FIFO Underflow Flag)

The TFUF flag indicates that no data remains in the transmit FIFO during frame transmission. Incomplete data is sent to the line.

TDE flag (Transmit Descriptor Empty Flag)

The TDE flag indicates that the TD0.TACT bit of the transmit descriptor is 0 while the previous transmit descriptor indicates that the frame is not complete (TD0.TFP[1:0] bits are 10b or 00b) in multi-buffer frame transmission. As a result, an incomplete frame might be sent.

When this flag sets to 1, perform a software reset and then set the EDTRR.TR bit to 1 to resume transmission. Transmission starts from the address stored in the TDLAR register.

TC flag (Frame Transfer Complete Flag)

The TC flag indicates that all the data specified in the transmit descriptor was transmitted from the ETHERC. This flag is set to 1 when one frame was transmitted in a single-buffer frame transmission or when the last data of a frame is transmitted in multi-buffer frame transmission and the TD0.TACT bit in the next transmit descriptor is 0. After frame transmission is complete, the EDMAC writes the transfer status back to the descriptor.

ECI flag (ETHERC Status Register Source Flag)

The ECI flag is set to 1 when an interrupt request is generated by the ECSR register.

ADE flag (Address Error Flag)

The ADE flag indicates that the memory address that the EDMAC tried to use for transfer is invalid.

RFCOF flag (Receive Frame Counter Overflow Flag)

The RFCOF flag indicates that the next frame reception started while the number of frames stored in the receive FIFO reached the maximum number of frames (16 frames). The received frame is discarded while the RFCOF flag is 1.

RABT flag (Receive Abort Detect Flag)

The RABT flag indicates that the ETHERC aborted frame reception because of a CRC error, PHY-LSI receive error, frame-too-short error, frame-too-long error, or other errors.

TABT flag (Transmit Abort Detect Flag)

The TABT flag indicates that the ETHERC aborted frame transmission because of transmit retry over, loss of carrier, no carrier detection, or other errors.

TWB flag (Write-Back Complete Flag)

The TWB flag indicates the EDMAC completed writing back to the descriptor after frame transmission. This flag is set to 1 after each frame transmission when the TRIMD.TIM bit is 0. It only sets to 1 when the TRIMD.TIS bit is 1.

28.2.7 EESIPR : ETHERC/EDMAC Status Interrupt Enable Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	TWBIP	—	—	—	TABTIP	RABTIP	RFCOFIP	ADEIP	ECIIP	TCIP	TDEIP	TFUFIP	FRIP	RDEIP	RFOFIP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CNDIP	DLCIP	CDIP	TROIP	RMAFIP	—	—	RRFIP	RTLFIP	RTSFIP	PREIP	CERFIP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CERFIP	CRC Error Interrupt Request Enable 0: Disable CRC error interrupt requests 1: Enable CRC error interrupt requests.	R/W

Bit	Symbol	Function	R/W
1	PREIP	PHY-LSI Receive Error Interrupt Request Enable 0: Disable PHY-LSI receive error interrupt requests 1: Enable PHY-LSI receive error interrupt requests.	R/W
2	RTSFIP	Frame-Too-Short Error Interrupt Request Enable 0: Disable frame-too-short error interrupt requests 1: Enable frame-too-short error interrupt requests.	R/W
3	RTLFIP	Frame-Too-Long Error Interrupt Request Enable 0: Disable frame-too-long error interrupt requests 1: Enable frame-too-long error interrupt requests.	R/W
4	RRFIP	Alignment Error Interrupt Request Enable 0: Disable alignment error interrupt requests 1: Enable alignment error interrupt requests.	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	RMAFIP	Multicast Address Frame Receive Interrupt Request Enable 0: Disable multicast address frame receive interrupt requests 1: Enable multicast address frame receive interrupt requests.	R/W
8	TROIP	Transmit Retry Over Interrupt Request Enable 0: Disable transmit retry over interrupt requests 1: Enable transmit retry over interrupt requests.	R/W
9	CDIP	Late Collision Detect Interrupt Request Enable 0: Disable late collision detected interrupt requests 1: Enable late collision detected interrupt requests.	R/W
10	DLCIP	Loss of Carrier Detect Interrupt Request Enable 0: Disable loss of carrier detected interrupt requests 1: Enable loss of carrier detected interrupt requests.	R/W
11	CNDIP	Carrier Not Detect Interrupt Request Enable 0: Disable carrier not detected interrupt requests 1: Enable carrier not detected interrupt requests.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	RFOFIP	Receive FIFO Overflow Interrupt Request Enable 0: Disable overflow interrupt requests 1: Enable overflow interrupt requests.	R/W
17	RDEIP	Receive Descriptor Empty Interrupt Request Enable 0: Disable receive descriptor empty interrupt requests 1: Enable receive descriptor empty interrupt requests.	R/W
18	FRIP	Frame Receive Interrupt Request Enable 0: Disable frame reception interrupt requests 1: Enable frame reception interrupt requests.	R/W
19	TFUFIP	Transmit FIFO Underflow Interrupt Request Enable 0: Disable underflow interrupt requests 1: Enable underflow interrupt requests.	R/W
20	TDEIP	Transmit Descriptor Empty Interrupt Request Enable 0: Disable transmit descriptor empty interrupt requests 1: Enable transmit descriptor empty interrupt requests.	R/W
21	TCIP	Frame Transfer Complete Interrupt Request Enable 0: Disable frame transmission complete interrupt requests 1: Enable frame transmission complete interrupt requests.	R/W
22	ECIIP	ETHERC Status Register Source Interrupt Request Enable 0: Disable ETHERC status interrupt requests 1: Enable ETHERC status interrupt requests.	R/W
23	ADEIP	Address Error Interrupt Request Enable 0: Disable address error interrupt requests 1: Enable address error interrupt requests.	R/W

Bit	Symbol	Function	R/W
24	RFCOFIP	Receive Frame Counter Overflow Interrupt Request Enable 0: Disable receive frame counter overflow interrupt requests 1: Enable receive frame counter overflow interrupt requests.	R/W
25	RABTIP	Receive Abort Detect Interrupt Request Enable 0: Disable receive abort detected interrupt requests 1: Enable receive abort detected interrupt requests.	R/W
26	TABTIP	Transmit Abort Detect Interrupt Request Enable 0: Disable transmit abort detected interrupt requests 1: Enable transmit abort detected interrupt requests.	R/W
29:27	—	These bits are read as 0. The write value should be 0.	R/W
30	TWBIP	Write-Back Complete Interrupt Request Enable 0: Disable write-back complete interrupt requests 1: Enable write-back complete interrupt requests.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The EESIPR register enables interrupt requests associated with bits in the EESR register. When a bit in this register is 1, the associated interrupt request is enabled.

28.2.8 TRSCER : ETHERC/EDMAC Transmit/Receive Status Copy Enable Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0x38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	RMAFCE	—	—	RRFCE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	RRFCE	RRF Flag Copy Enable 0: Reflect the EESR.RRF flag status in the RD0.RFE bit of the receive descriptor 1: Do not reflect the EESR.RRF flag status in the RD0.RFE bit of the receive descriptor.	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	RMAFCE	RMAF Flag Copy Enable 0: Reflect the EESR.RMAF flag status in the RD0.RFE bit of the receive descriptor 1: Do not reflect the EESR.RMAF flag status in the RD0.RFE bit of the receive descriptor.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

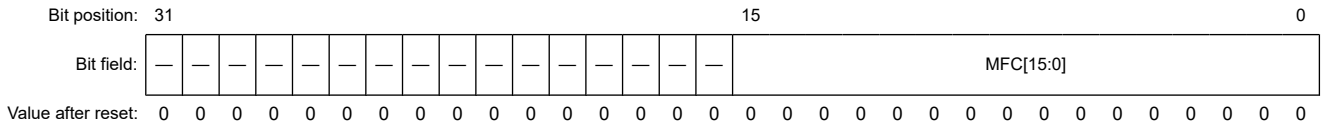
Note: S-TYPE3, P-TYPE3

The TRSCER register selects whether the receive status indicated in the EESR.RMAF and RRF flags is reflected in the RFE bit of the receive descriptor as a summary. The bits in this register are associated with bits in the EESR register that have the same number. When the RMAFCE or RRFCE bit is set to 0, the associated receive status is reflected in the RFE bit. When the RMAFCE or RRFCE bit is set to 1, the associated receive status is not reflected.

28.2.9 RMFCR : Missed-Frame Counter Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0x40



Bit	Symbol	Function	R/W
15:0	MFC[15:0]	Missed-Frame Counter These bits indicate the number of frames that are discarded and not transferred to the receive buffer during reception.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

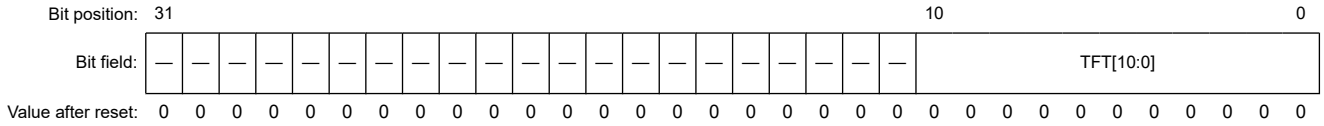
The RMFCR register indicates that the number of frames that could not be stored in the receive FIFO and so were discarded during reception. When the receive FIFO overflows, it stops receiving data, and the rest of frames are discarded. At the same time, the RMFCR register value is incremented. When the RMFCR register value reaches 0xFFFF, count-up is halted. Writing any value to the RMFCR register clears the counter value to 0.

For frames that are not completely received, after data in the receive FIFO is transferred to the receive buffer, the RACT bit in the receive descriptor 0 (RD0) clears to 0 (descriptor disabled), the RFS9 bit sets to 1 (receive FIFO overflowed), and the EESR.RFOF flag is set to 1 (overflow detected).

28.2.10 TFTR : Transmit FIFO Threshold Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0x48



Bit	Symbol	Function	R/W
10:0	TFT[10:0]	Transmit FIFO Threshold 0x000: Store-and-forward mode 0x001 to 0x00C: Setting prohibited 0x00D to 0x200: The threshold is the set value multiplied by 4. 0x201 to 0x7FF: Setting prohibited Example: TFT[10:0] = 0x00D : 52 bytes TFT[10:0] = 0x040 : 256 bytes TFT[10:0] = 0x100 : 1024 bytes TFT[10:0] = 0x200 : 2048 bytes	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: When starting transmission before one frame data is completely written, take care to prevent an underflow. To prevent a transmit underflow, Renesas recommends using the initial value (store-and-forward mode).

The TFTR register specifies the transmit FIFO threshold at which the first transmission starts. The actual threshold is the set value multiplied by 4.

The ETHERC starts transmission when the amount of data in the transmit FIFO exceeds the number of bytes set in this register, when the transmit FIFO is full, or when one frame of data is completely written. Set the TFTR register while the EDTRR.TR bit is 0.

28.2.11 FDR : FIFO Depth Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0x50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TFD[4:0]					—	—	—	RFD[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	RFD[4:0]	Receive FIFO Depth 0x0F: 4096 bytes Others: settings prohibited	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
12:8	TFD[4:0]	Transmit FIFO Depth 0x07: 2048 bytes Others: settings prohibited	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The FDR register specifies the transmit and receive FIFO depths. Set this register to 0x0000_070F before starting transmission and reception.

28.2.12 RMCR : Receive Method Control Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0x58

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RNR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RNR	Receive Request Reset 0: EDRRR.RR bit (receive request bit) is cleared to 0 when one frame is received 1: EDRRR.RR bit (receive request bit) is not cleared to 0 when one frame is received.	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

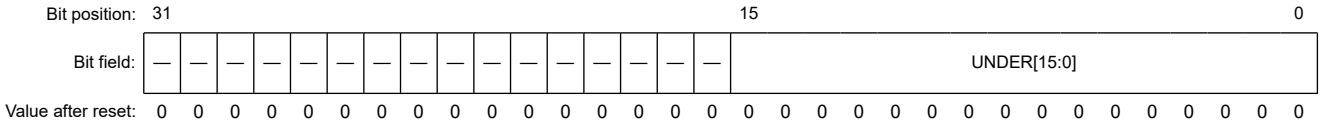
The RMCR register specifies how to control the EDRRR.RR bit when receiving a frame. When the RNR bit is 0, the EDRRR.RR bit clears to 0 when one frame is received, so it must be set to 1 by software to receive the subsequent frame. When the RNR bit is 1, the EDRRR.RR bit does not clear to 0 when one frame is received, and the EDMAC reads the

next receive descriptor and continues frame reception. Renesas recommends setting the RNR bit to 1 when receiving data continuously. Set the RMCR register while the EDRRR.RR bit is 0.

28.2.13 TFUCR : Transmit FIFO Underflow Counter

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0x64



Bit	Symbol	Function	R/W
15:0	UNDER[15:0]	Transmit FIFO Underflow Count These bits indicate how many times the transmit FIFO underflows. The counter stops when the counter value reaches 0xFFFF.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

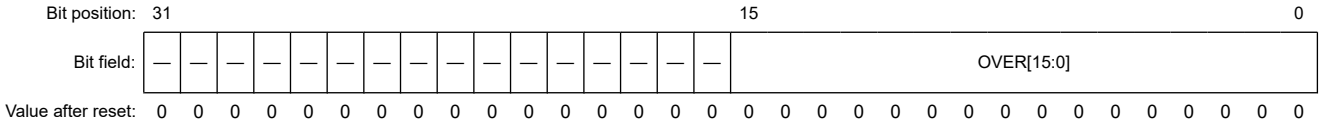
Note: S-TYPE3, P-TYPE3

The TFUCR register indicates how many times the transmit FIFO underflows. Writing any value to the TFUCR register clears the counter value to 0.

28.2.14 RFOCR : Receive FIFO Overflow Counter

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0x68



Bit	Symbol	Function	R/W
15:0	OVER[15:0]	Receive FIFO Overflow Count These bits indicate how many times the receive FIFO overflows. The counter stops when the counter value reaches 0xFFFF.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RFOCR register indicates how many times the receive FIFO overflows. Writing any value to the RFOCR register clears the counter value to 0.

28.2.15 IOSR : Independent Output Signal Setting Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0x6C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ELB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ELB	External Loopback Mode 0: Output low on the ET0_EXOUT pin 1: Output high on the ET0_EXOUT pin.	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The IOSR register selects the output level of the ETHERC external output pin (ET0_EXOUT) in external loopback mode. The ELB bit value is output on the ET0_EXOUT pin, which can be used to set loopback mode for the PHY-LSI.

To use the loopback function of the PHY-LSI through this register, you must connect the PHY-LSI to the ET0_EXOUT pin.

28.2.16 FCFTR : Flow Control Start FIFO Threshold Setting Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0x70

Bit position:	31	18	16	2	0
Bit field:	—	RFFO[2:0]	—	RFDO[2:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0	1 1 1	0 0 0 0 0 0 0 0 0 0 0 0 0	1 1 1	

Bit	Symbol	Function	R/W
2:0	RFDO[2:0]	Receive FIFO Data PAUSE Output Threshold 0 0 0: When 224 (256 to 32) bytes of data is stored in the receive FIFO 0 0 1: When 480 (512 to 32) bytes of data is stored in the receive FIFO ⋮ 1 1 0: When 1760 (1792 to 32) bytes of data is stored in the receive FIFO 1 1 1: When 2016 (2048 to 32) bytes of data is stored in the receive FIFO.	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
18:16	RFFO[2:0]	Receive FIFO Frame PAUSE Output Threshold 0 0 0: When 2 receive frames are stored in the receive FIFO 0 0 1: When 4 receive frames are stored in the receive FIFO 0 1 0: When 6 receive frames are stored in the receive FIFO ⋮ 1 1 0: When 14 receive frames are stored in the receive FIFO 1 1 1: When 16 receive frames are stored in the receive FIFO.	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

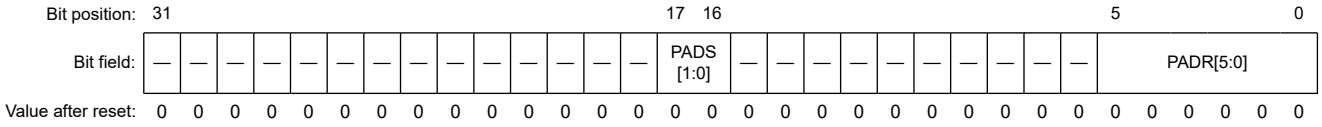
The FCFTR register specifies the ETHERC flow control. Set the threshold to automatically transmit a PAUSE frame.

The threshold can be set using the data size (RFDO[2:0] bits) and the number of frames (RFFO[2:0] bits) stored in the receive FIFO. Flow control starts when the stored data size or the number of stored frames reaches its threshold.

28.2.17 RPADIR : Receive Data Padding Insert Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0x78



Bit	Symbol	Function	R/W
5:0	PADR[5:0]	Padding Slot 0x00: Insert padding at the head of received data 0x01: Insert padding between the 1st and 2nd bytes of received data ⋮ 0x3E: Insert padding between the 62nd and 63rd bytes of received data 0x3F: Insert padding between the 63rd and 64th bytes of received data.	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W
17:16	PADS[1:0]	Padding Size 0 0: Do not insert padding 0 1: Insert 1 byte 1 0: Insert 2 bytes 1 1: Insert 3 bytes	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

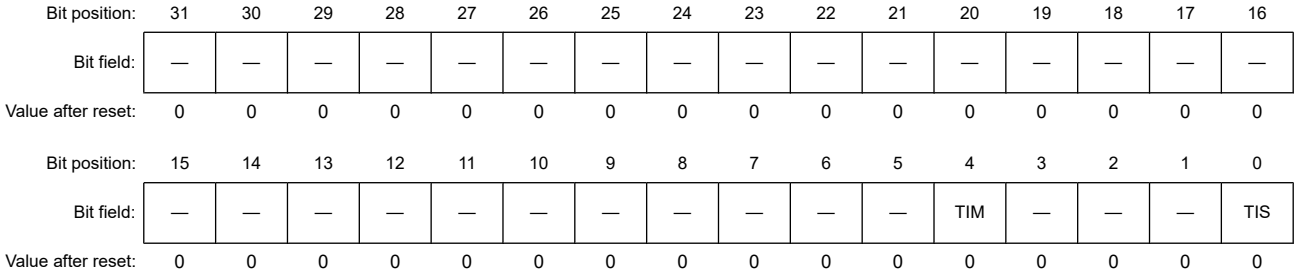
Note: S-TYPE3, P-TYPE3

The RPADIR register specifies insertion of padding for received data. The padding value is 0x00. Set the EDMR.SWR bit to 1 to reset before rewriting the PRADIR register.

28.2.18 TRIMD : Transmit Interrupt Setting Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0x7C



Bit	Symbol	Function	R/W
0	TIS	Transmit Interrupt Enable 0: Disable transmit interrupts 1: Enable transmit Interrupts. Set the EESR.TWB flag to 1 in the mode selected in the TIM bit to report an interrupt.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
4	TIM	Transmit Interrupt Mode 0: Select transmission complete interrupt mode, where an interrupt occurs when a frame is transmitted 1: Select write-back complete interrupt mode, where an interrupt occurs when write-back to the transmit descriptor is complete while the TWBI bit is 1.	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

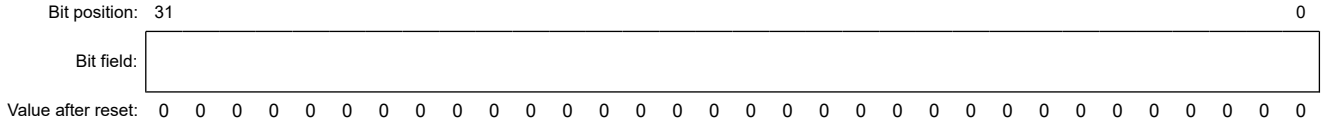
Note: S-TYPE3, P-TYPE3

The TRIMD register specifies the transmit interrupt mode and enables or disables transmit interrupts. When the condition selected in this register is satisfied, the EESR.TWB flag sets to 1, and an interrupt request is output when the EESIPR.TWBIP bit is 1.

28.2.19 RBWAR : Receive Buffer Write Address Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0xC8



Bit	Symbol	Function	R/W
31:0	n/a	The RBWAR register indicates the last address that the EDMAC wrote data to, when writing to the receive buffer.	R

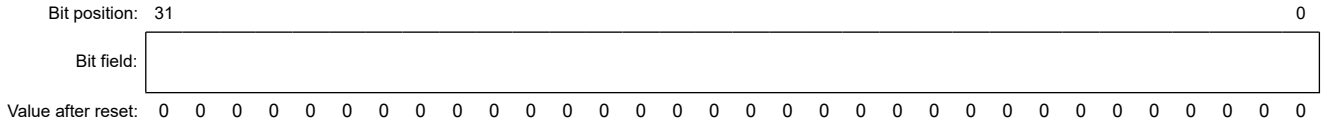
Note: S-TYPE3, P-TYPE3

The RBWAR register indicates the last address that the EDMAC wrote data to, when writing to the receive buffer. Check the contents of this register to identify the address in the receive buffer to which the EDMAC is writing data to. The address that the EDMAC is outputting to the receive buffer might not match the read value of the RBWAR register during data reception. The RBWAR register is read-only. Do not write to this register.

28.2.20 RDFAR : Receive Descriptor Fetch Address Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0xCC



Bit	Symbol	Function	R/W
31:0	n/a	The RDFAR register indicates the start address of the last fetched receive descriptor when the EDMAC is fetching descriptor information from the receive descriptor.	R

Note: S-TYPE3, P-TYPE3

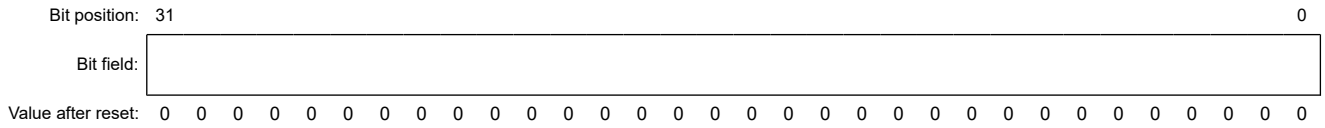
The RDFAR register indicates the start address of the last fetched receive descriptor when the EDMAC is fetching descriptor information from the receive descriptor. Check the contents of this register to identify which receive descriptor information the EDMAC is using for active processing. The address of the receive descriptor that the EDMAC is fetching might not match the read value of the RDFAR register during data reception. The RDFAR register is read-only.

Do not write to this register.

28.2.21 TBRAR : Transmit Buffer Read Address Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0xD4



Bit	Symbol	Function	R/W
31:0	n/a	The TBRAR register indicates the last address from which the EDMAC read data, when reading data from the transmit buffer.	R

Note: S-TYPE3, P-TYPE3

The TBRAR register indicates the last address from which the EDMAC read data, when reading data from the transmit buffer. Check the contents of this register to identify which address in the transmit buffer the EDMAC is reading from.

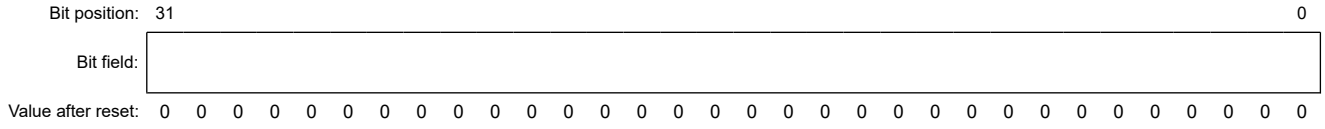
The address that the EDMAC is outputting to the transmit buffer might not match the read value of the TBRAR register.

The TBRAR register is read-only. Do not write to this register.

28.2.22 TDFAR : Transmit Descriptor Fetch Address Register

Base address: EDMAC0 = 0x4035_4000
EDMAC0_NS = 0x5035_4000

Offset address: 0xD8



Bit	Symbol	Function	R/W
31:0	n/a	The TDFAR register indicates the start address of the last fetched transmit descriptor when the EDMAC is fetching the descriptor information from the transmit descriptor.	R

Note: S-TYPE3, P-TYPE3

The TDFAR register indicates the start address of the last fetched transmit descriptor when the EDMAC is fetching the descriptor information from the transmit descriptor. Check the contents of this register to identify which transmit descriptor information the EDMAC is using for active processing. The address of transmit descriptor that the EDMAC fetches might not match the read value of the TDFAR register. The TDFAR is read only. Do not write to this register.

28.3 Operation

The EDMAC transfers data according to the information written in the descriptor. Two descriptors are provided: transmit and receive. A descriptor includes the buffer size, address, and transmit or receive status. The EDMAC transmits or receives data continuously by using sequentially arranged descriptors.

28.3.1 Descriptor Lists and Data Buffers

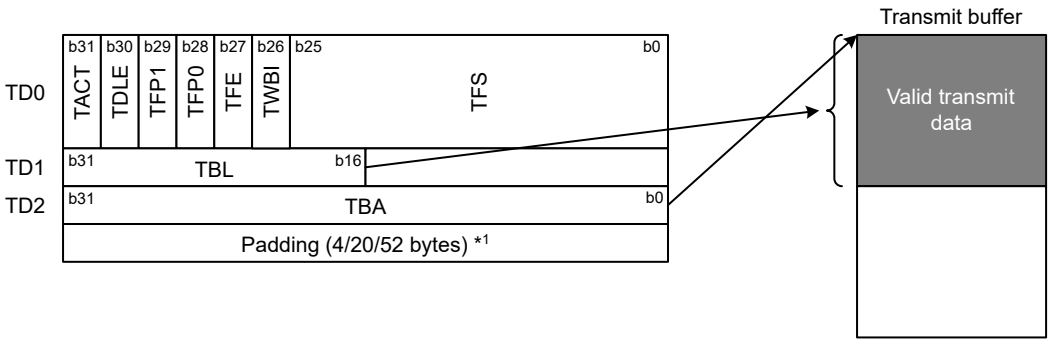
To transfer data using the EDMAC, create the transmit and receive descriptor lists in memory, set the start address of the transmit descriptor list in the TDLAR register, and set the start address of the receive descriptor list in the RDLAR register. Also, the transmit and receive buffers associated with each descriptor are required.

Align the descriptor list on the appropriate address boundary according to the descriptor length set in the EDMR.DL[1:0] bits. The transmit buffer can be aligned on a word boundary, halfword boundary, or byte boundary. When the valid transmit buffer size is 16 bytes or less, align it on a 32-byte boundary. When the valid transmit buffer size is larger than 16bytes, the transmit buffer setting on not aligned 32-byte boundary is permitted. However EDMAC0 might read the transmit buffer aligned on 32-byte boundary, therefore, initialize the transmit buffer area at the beginning of 32 byte-boundary where the

transmit buffer set in. Align the receive buffer on a 32-byte boundary. Set different addresses for the transmit and receive descriptors and buffers for EDMAC0.

28.3.1.1 Transmit descriptor

Figure 28.3 shows the relationship between a transmit descriptor and transmit buffer. A transmit descriptor consists of TD0 to TD2. The transmit frame and transmit buffer configuration can be specified as one buffer per frame (single-buffer frame transmission) or multiple buffers per frame (multi-buffer frame transmission) by setting the transmit descriptor.



Note 1. The padding size is determined by the descriptor length (16/32/64 bytes).

Figure 28.3 Relationship between transmit descriptor and transmit buffer

(1) Transmit descriptor 0 (TD0)

Note: Bits for write-back are underlined.

Bit	Symbol	Function	R/W
25:0	TFS	Transmit Frame Status Set all bits to 0 when creating a descriptor. After write-back, the bits indicate the following: TFS25 to TFS9: Reserved TFS8: Transmit abort was detected (value is equivalent to the EESR.TABT flag) TFS7 to TFS4: Reserved TFS3: No carrier was detected (value is equivalent to the EESR.CND flag) TFS2: Loss of carrier was detected (value is equivalent to the EESR.DLC flag) TFS1: Late collision during transmission was detected (value is equivalent to the EESR.CD flag) TFS0: Transmit retry over (value is equivalent to the EESR.TRO flag). When a bit sets to 1, it indicates that the associated error occurred during frame transmission. When any of the TFS bits sets to 1, the TFE bit also is set to 1. When any of bits TFS3 to TFS0 is set to 1, TFS8 is also set to 1.	R/W
26	TWBI	Write-Back Complete Interrupt Enable 0: Do not generate interrupt when write-back to this descriptor is complete 1: Generate interrupt when write-back to this descriptor is complete.	R/W
27	TFE	Transmit Frame Error 0: Frame transmission is successfully complete 1: Error occurred during frame transmission (transmission aborted).	R/W
29:28	TFP[1:0]	Transmit Frame Position 0 0: Transmit buffer indicated in this descriptor is the middle of a transmit frame (frame information is incomplete) 0 1: Transmit buffer indicated in this descriptor is the end of a transmit frame (frame information is complete) 1 0: Transmit buffer indicated in this descriptor is the head of a transmit frame (frame information is incomplete) 1 1: Transmit buffer indicated in this descriptor is all of a transmit frame (one buffer per frame).	R/W

Note: Bits for write-back are underlined.

Bit	Symbol	Function	R/W
30	TDLE	Transmit Descriptor List End When this bit is 1, it indicates that this descriptor is the last in the descriptor list.	R/W
31	TACT	Transmit Descriptor Valid This bit indicates that this descriptor is valid.	R/W

TD0 specifies the transmit frame settings and indicates the status after transmission.

TFE bit (Transmit Frame Error)

When the TFE bit is 1, it indicates that any of the TFS bits is 1.

TFP[1:0] bits (Transmit Frame Position)

The TFP[1:0] bits indicate which part of a transmit frame corresponds to the transmit buffer indicated in this descriptor.

The TFP[1:0] and TD1.TBL bit settings must be logically consistent in the previous and next descriptors.

TACT bit (Transmit Descriptor Valid)

The TACT bit indicates that this descriptor is valid. The TACT bit is set to 1 by software. This bit clears to 0 when the transmit frame is transferred or when the transmission is aborted.

(2) Transmit descriptor 1 (TD1)

Bit	Symbol	Function	R/W
15:0	—	The read value is 0. The write value should be 0.	R/W
31:16	TBL	Transmit Buffer Length Specifies the valid byte length of the associated transmit buffer. Set a value equal to or greater than 1.	R/W

TD1 specifies the valid byte length of the transmit buffer.

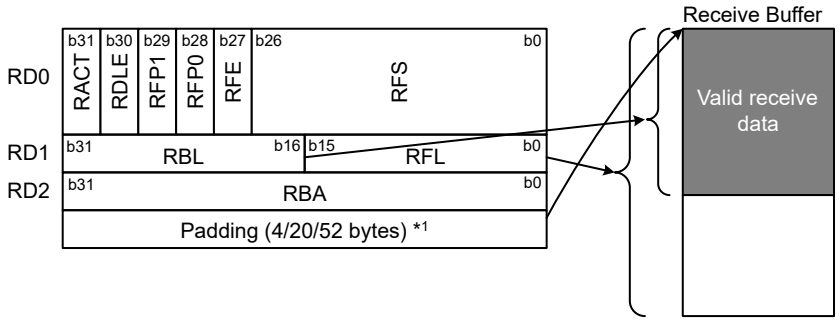
(3) Transmit descriptor 2 (TD2)

Bit	Symbol	Function	R/W
31:0	TBA	Transmit Buffer Address Specifies the start address of the transmit buffer. When the TD1.TBL bit value is 1 to 16 bytes, align it on a 32-byte boundary.	R/W

TD2 specifies the start address of the transmit buffer.

28.3.1.2 Receive descriptor

Figure 28.4 shows the relationship between a receive descriptor and receive buffer. The receive frame and receive buffer configuration can be specified as one buffer per frame (single-buffer frame transmission) or multiple buffers per frame (multi-buffer frame transmission) by setting the receive descriptor. If the receive buffer length (RBL) is set to 0, the operation indicated in the descriptor is not guaranteed.



Note 1. The padding size is determined by the descriptor length (16/32/64 bytes).

Figure 28.4 Relationship between receive descriptor and receive buffer

(1) Receive descriptor 0 (RD0)

Note: Bits for write-back are underlined.

Bit	Symbol	Function	R/W
26:0	RFS	Receive Frame Status Set all bits to 0 when creating a descriptor. After write-back, the bits indicate the following: RFS26 to RFS10: Reserved RFS9: Receive FIFO overflow (value is equivalent to the EESR.RFOF flag) RFS8: Receive abort was detected (value is equivalent to the EESR.RABT flag) RFS7: Multicast address frame was received (value is equivalent to the EESR.RMAF flag) RFS6 and RFS5: Reserved RFS4: Alignment error was detected (value is equivalent to the EESR.RRF flag) RFS3: Frame-too-long error (value is equivalent to the EESR.RTLF flag) RFS2: Frame-too-short error (value is equivalent to the EESR.RTSF flag) RFS1: PHY-LSI receive error (value is equivalent to the EESR.PRE flag) RFS0: CRC error (value is equivalent to the EESR.CERF flag). When a bit sets to 1, it indicates that the associated error occurred during frame reception. When any of the RFS bits sets to 1, the RFE bit also sets to 1. Set the TRSCER register to select whether bits RFS7 and RFS4 are reflected in the RFE bit. When any of bits RFS3 to RFS0 sets to 1, RFS8 also sets to 1.	R/W
27	RFE	Receive Frame Error 0: No error occurred in the received frame 1: Error occurred in the received frame.	R/W
29:28	RFP[1:0]	Receive Frame Position 0 0: Receive buffer indicated in this descriptor is the middle of a receive frame (frame information is incomplete) 0 1: Receive buffer indicated in this descriptor is the end of a receive frame (frame information is complete) 1 0: Receive buffer indicated in this descriptor is the head of a receive frame (frame information is incomplete) 1 1: Receive buffer indicated in this descriptor is all of a receive frame (one buffer per frame).	R/W
30	RDLE	Receive Descriptor List End When this bit is 1, it indicates that this descriptor is the last in the descriptor list.	R/W
31	RACT	Receive Descriptor Valid Indicates that this descriptor is valid.	R/W

RD0 indicates the receive frame status.

RFE bit (Receive Frame Error)

When the RFE bit is 1, it indicates that any of the RFS bits is 1. Set the TRSCER register to select whether the RFS7 and RFS4 bits of EDMAC0 are reflected in the RFE bit.

RFP[1:0] bits (Receive Frame Position)

The RFP[1:0] bits indicate which part of a receive frame corresponds to the receive buffer indicated in this descriptor.

RACT bit (Receive Descriptor Valid)

The RACT bit indicates that this descriptor is valid. The RACT bit is set to 1 by software. This bit clears to 0 when all the data is transferred to the receive buffer indicated in RD2 or when the receive buffer becomes full.

(2) Receive descriptor 1 (RD1)

Bit	Symbol	Function	R/W
15:0	RFL	Receive Frame Length Specifies the length (number of bytes) of the receive frame stored in the buffer. This does not include the number of bytes for padding set in the RPADIR register. These bits are written back to the descriptor associated with the end of a frame.	R/W
31:16	RBL	Receive Buffer Length Specifies the byte length of the associated receive buffer. Set an integral multiple of 32 as the buffer length.	R/W

Note: Bits for write-back are underlined.

RD1 specifies the receive buffer length. When reception is complete, the receive frame length is written back.

(3) Receive descriptor 2 (RD2)

Bit	Symbol	Function	R/W
31:0	RBA	Receive Buffer Address Specifies the start address of the receive buffer. Align the buffer address on a 32-byte boundary.	R/W

RD2 specifies the start address of the receive buffer.

28.3.2 Transmission

When the EDTRR.TR bit is set to 1 while the ETHERC0.ECMR.TE bit is 1, the EDMAC reads the descriptor following the previously used descriptor in the transmit descriptor list (or the descriptor indicated in the TDLAR register after a reset). When the TACT bit is 1 in the transmit descriptor (TD0), the EDMAC sequentially reads transmit data from the start address of the transmit buffer indicated in transmit descriptor 2 (TD2) and transfers it to the ETHERC through the transmit FIFO. The ETHERC creates a transmit frame and starts transmission to the MII or RMII. When all data indicated in the TD1.TBL bit is transferred, write-back is performed based on the TD0.TFP[1:0] bit setting as follows:

- When the TD0.TFP[1:0] bits are 00b or 10b (frame is incomplete), the TD0.TACT bit is written back
- When the TD0.TFP[1:0] bits are 01b or 11b (frame is complete), the TD0.TACT, TD0.TFS, and TD0.TFE bits are written back.

When the TD0.TACT bit in the read descriptor is 1, the EDMAC continues reading descriptors and transmit frames. When the TD0.TACT bit in the read descriptor is 0, the EDMAC sets the EDTRR.TR bit to 0 and stops transmission.

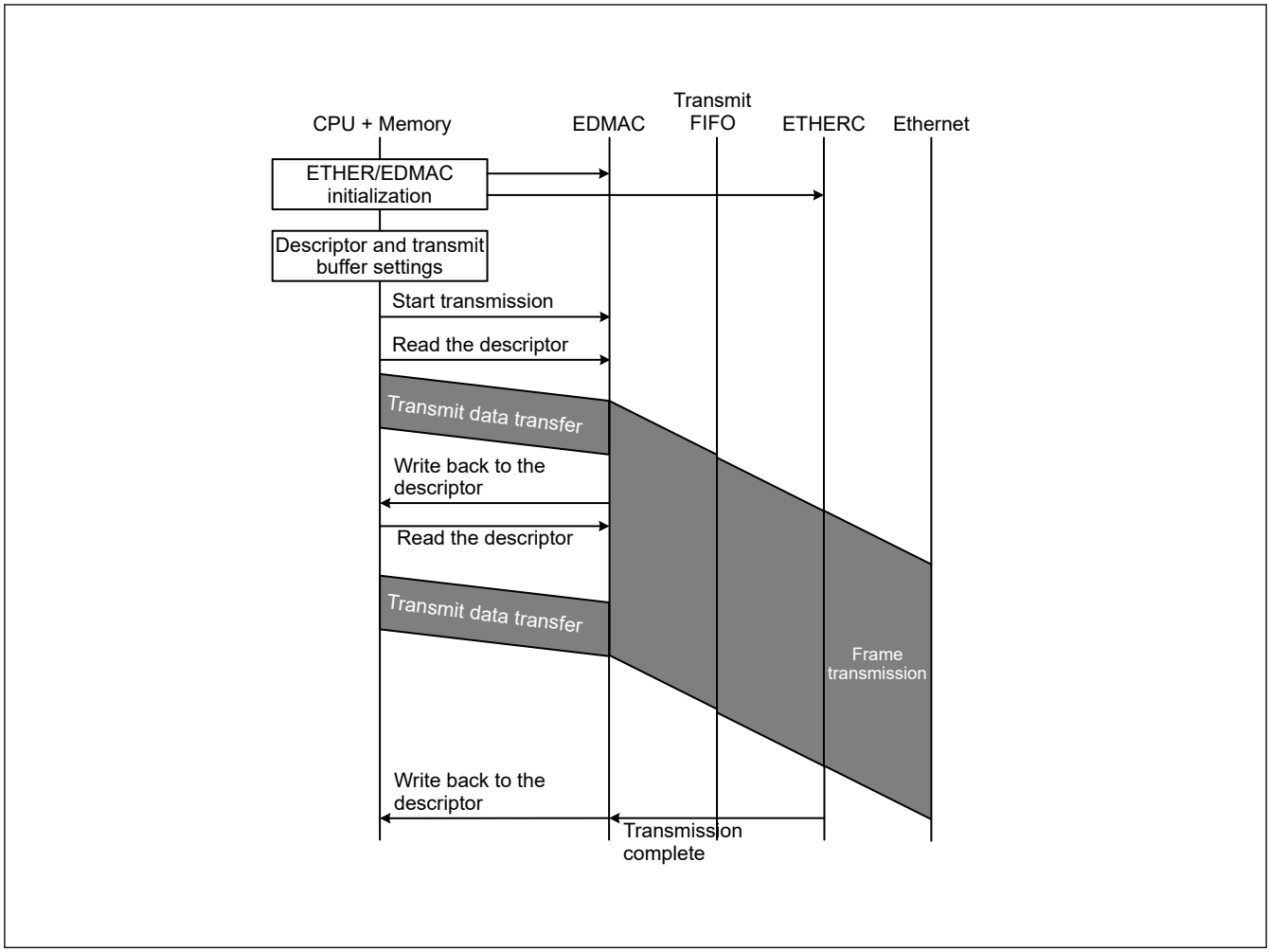


Figure 28.5 Example of transmission flow

28.3.3 Reception

When the EDRRR.RR bit is set to 1 while the ETHERC0.ECMR.RE bit is 1, the EDMAC reads the receive descriptor following the previously used descriptor (or the descriptor indicated in the RDLAR register after a reset) and then waits for reception. When the RD0.RACT bit is 1, if the data stored in the receive FIFO is 32 bytes or more, or if the end byte of the frame is stored in the receive buffer, the EDMAC transfers data from the receive FIFO to the receive buffer indicated in receive descriptor 2 (RD2).

If the data length of the received frame is longer than the buffer length set in the RBL bit in receive descriptor 1 (RD1), the EDMAC writes back 10b or 00b to the RD0.RFP[1:0] bits and 0 to the RD0.RACT bit when the receive buffer becomes full, and then the EDMAC reads the next data. After that, the EDMAC transfers data to another receive buffer.

When the frame reception is complete or when the frame reception is aborted by an error, the EDMAC writes back 11b or 01b to the RD0.RFP[1:0] bits, 0 to the RD0.RACT bit, and the receive frame length to the RD1.RFL bit. When the RMCR.RNR bit is 1, the EDMAC reads the next descriptor and waits for reception. When the RNR bit is 0, the EDMAC sets the EDRRR.RR bit to 0 and stops reception.

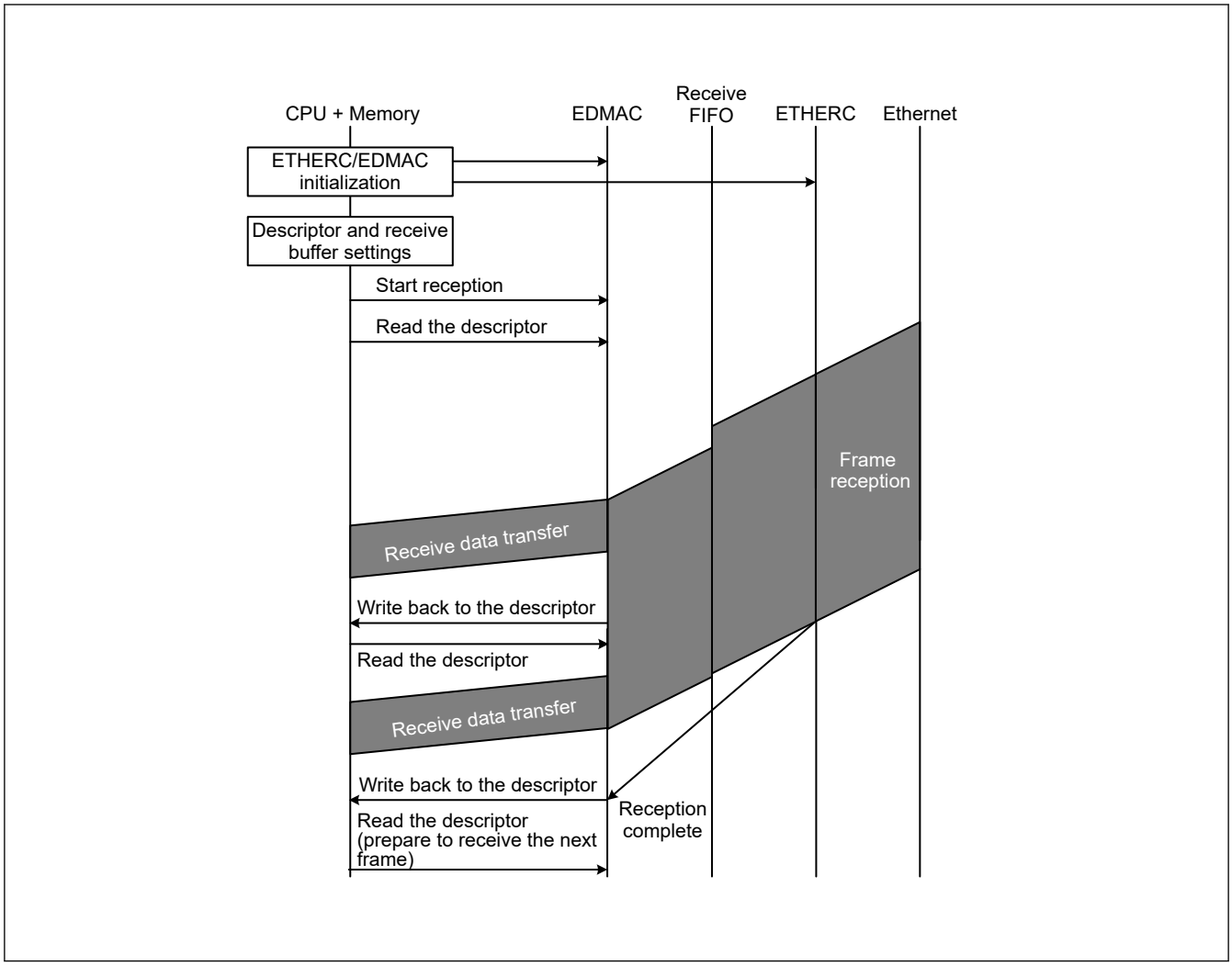


Figure 28.6 Example of reception flow

28.3.4 Multi-Buffer Frame Transmission

28.3.4.1 Error processing while transmitting multi-buffer frame

If an error occurs during multi-buffer frame transmission, the EDMAC performs the processing shown in Figure 28.7. In the figure, when the TACT bit of transmit descriptor 0 (TD0) is 0, the descriptor indicates that all data in the buffer is successfully transmitted. When the TACT bit is 1, the descriptor indicates that data in the buffer is not yet transmitted. If a frame transmit error*1 occurs in the head or at the middle of the frame while the TD0.TACT bit is 1, the EDMAC stops data transmission from the transmit FIFO and EDMAC data transfer, and sets the TD0.TACT bit to 0.

After that, the EDMAC reads the next descriptor to see if the descriptor indicates the middle of the frame (TD0.TFP[1:0] bits are 00b) or the end of the frame (TD0.TFP[1:0] bits are 01b). When the descriptor indicates the middle of the frame, the EDMAC sets the TD0.TACT bit to 0 and reads the next descriptor. When the descriptor indicates the end of the frame, in addition to setting the TD0.TACT bit to 0, the EDMAC also writes back to the TD0.TFE and TD0.TFS bits.

After an error occurs, data in the buffer is not transmitted until write-back to the descriptor for the end of the frame. When the associated transmit error interrupt is enabled in the EESIPR register, an interrupt request is generated immediately after write-back to the descriptor for the end of the frame.

Note 1. A transmit retry-over condition, late collision, or loss of carrier is detected, or a carrier is not detected.

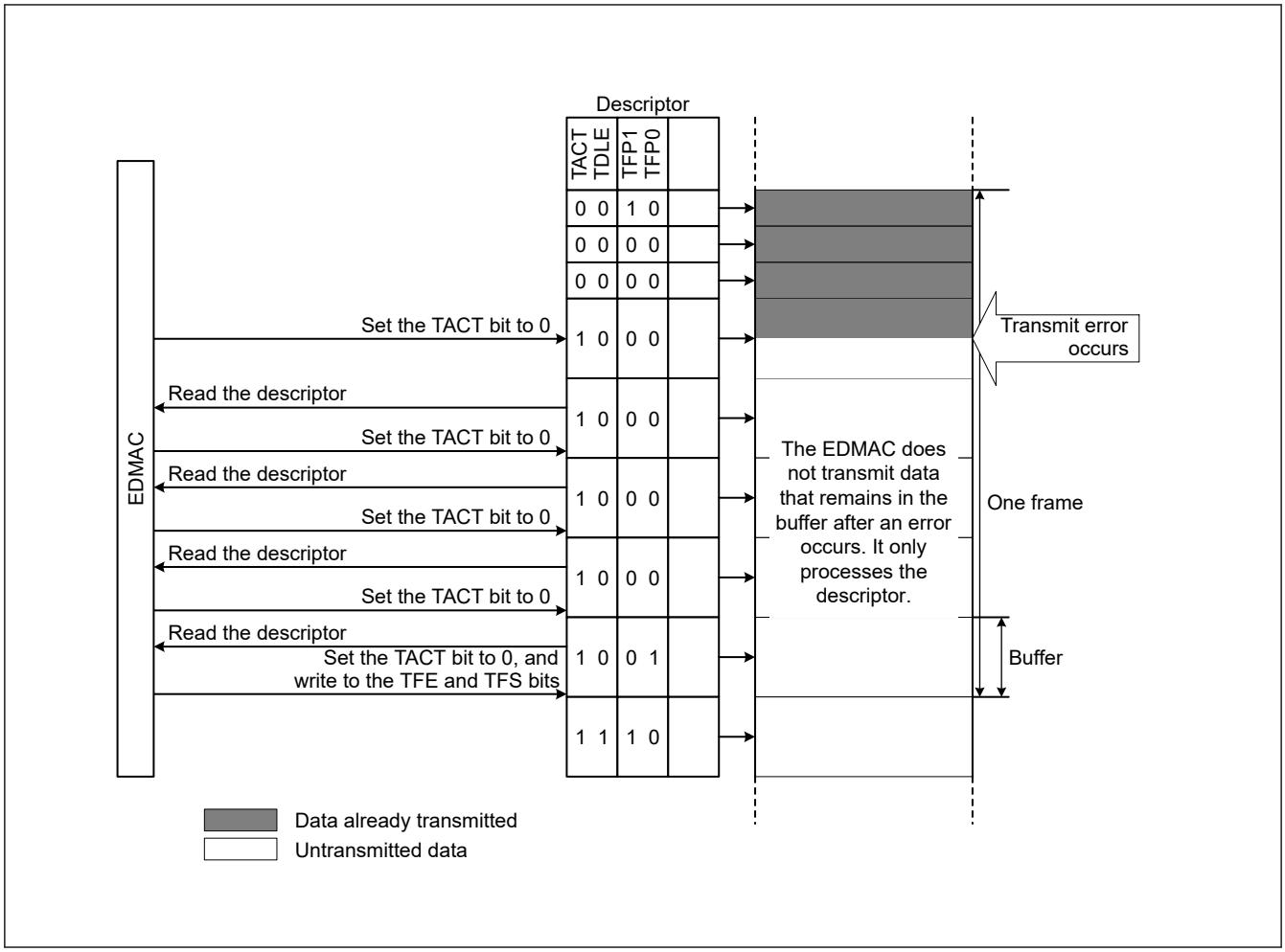


Figure 28.7 EDMAC operation after transmit error occurs

28.3.4.2 Error processing while receiving multi-buffer frame

If an error occurs during multi-buffer frame reception, the EDMAC performs the processing shown in Figure Figure 28.8. In the figure, when the RACT bit of receive descriptor 0 (RD0) is 0, the descriptor indicates that data was successfully received in the buffer. When the RACT bit is 1, the descriptor indicates that data is not yet received in the buffer. If a frame receive error*1 occurs, the EDMAC stops receiving new data, but it transfers data that is already stored in the receive FIFO to the receive buffer.

When the receive buffer becomes full during transfer, the EDMAC sets the RACT bit to 0 and the RFP[1:0] bits to 10b or 00b and reads the next descriptor. After all data in the receive FIFO is transferred, the EDMAC writes back the status to the descriptor.

When the associated receive error interrupt is enabled in the EESIPR register, an interrupt request is generated immediately after write-back to the descriptor. When there is a request to receive a new frame, the EDMAC continues reception using the descriptor following the descriptor where the error occurred.

Note 1. A CRC error, PHY-LSI receive error, frame-too-short error, frame-too-long error, or alignment error is detected.

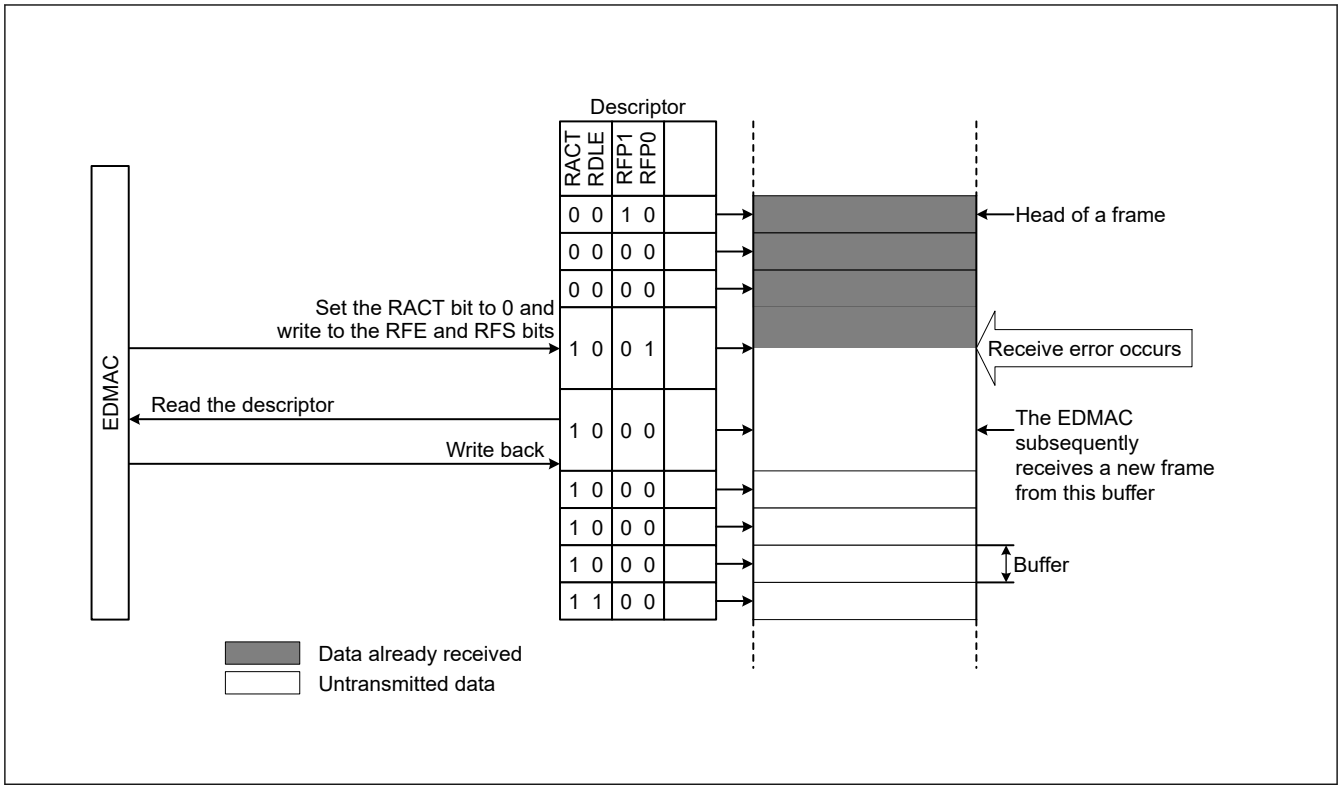


Figure 28.8 EDMAC operation after receive error occurs

28.3.5 Bus Transfer Error

Bus transfer error occurs with the slave TrustZone filter error, the master MPU error, the slave bus error or the illegal address access error. When the bus transfer error is detected, the EDMAC halts the process and EESR.ADE bit is set to 1.

Bus transfer error can be output as an interrupt request signal (ETHER_EINT0). The slave TrustZone error and the master MPU error can be output as NMI. When ETHER_EINT0 and NMI are generated, NMI always responds first. Figure 28.9 and Figure 28.10 show the processing for the bus error in NMI handler and ETHER_EINT0 handler.

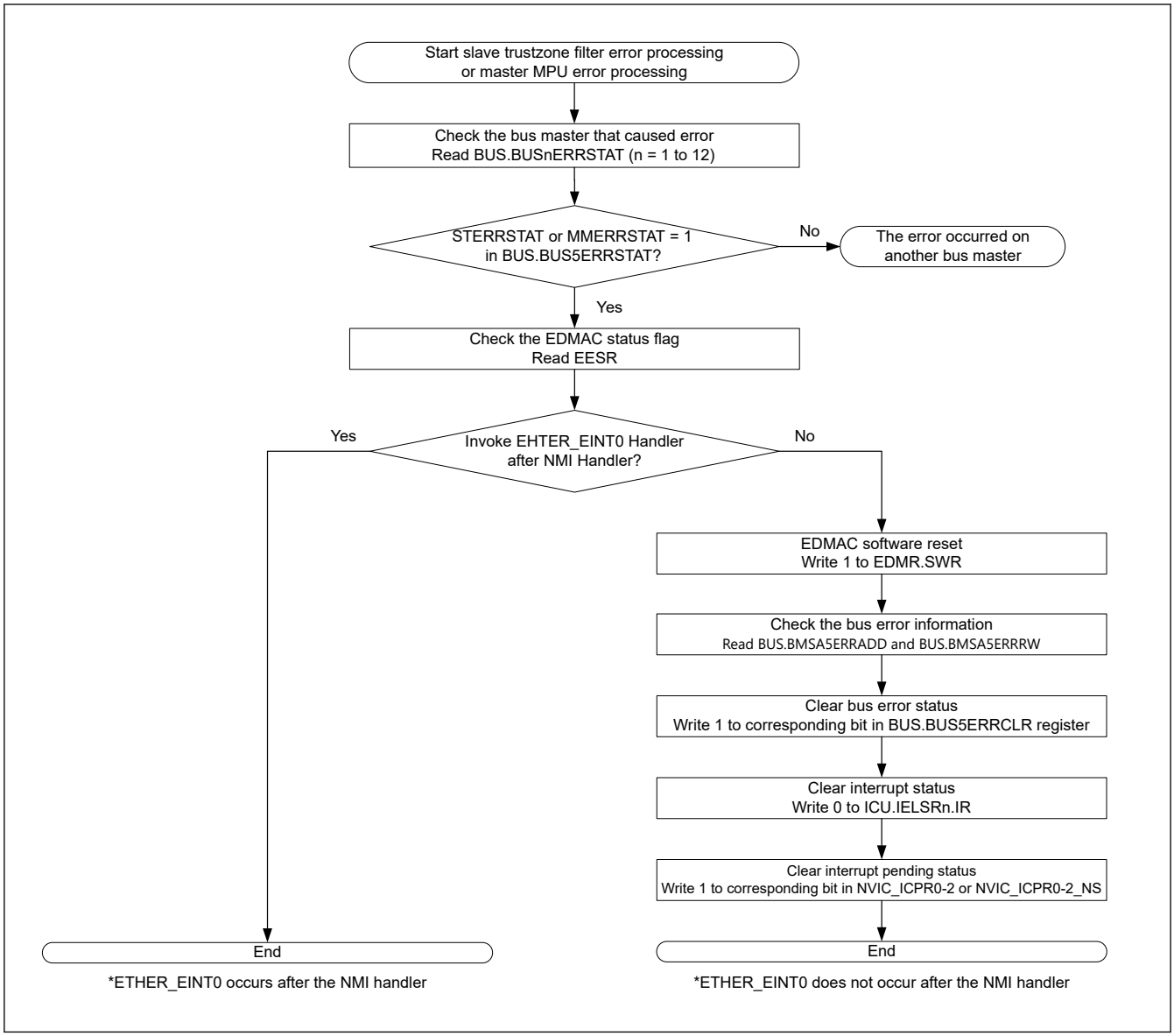


Figure 28.9 Processing for bus transfer error in NMI handler

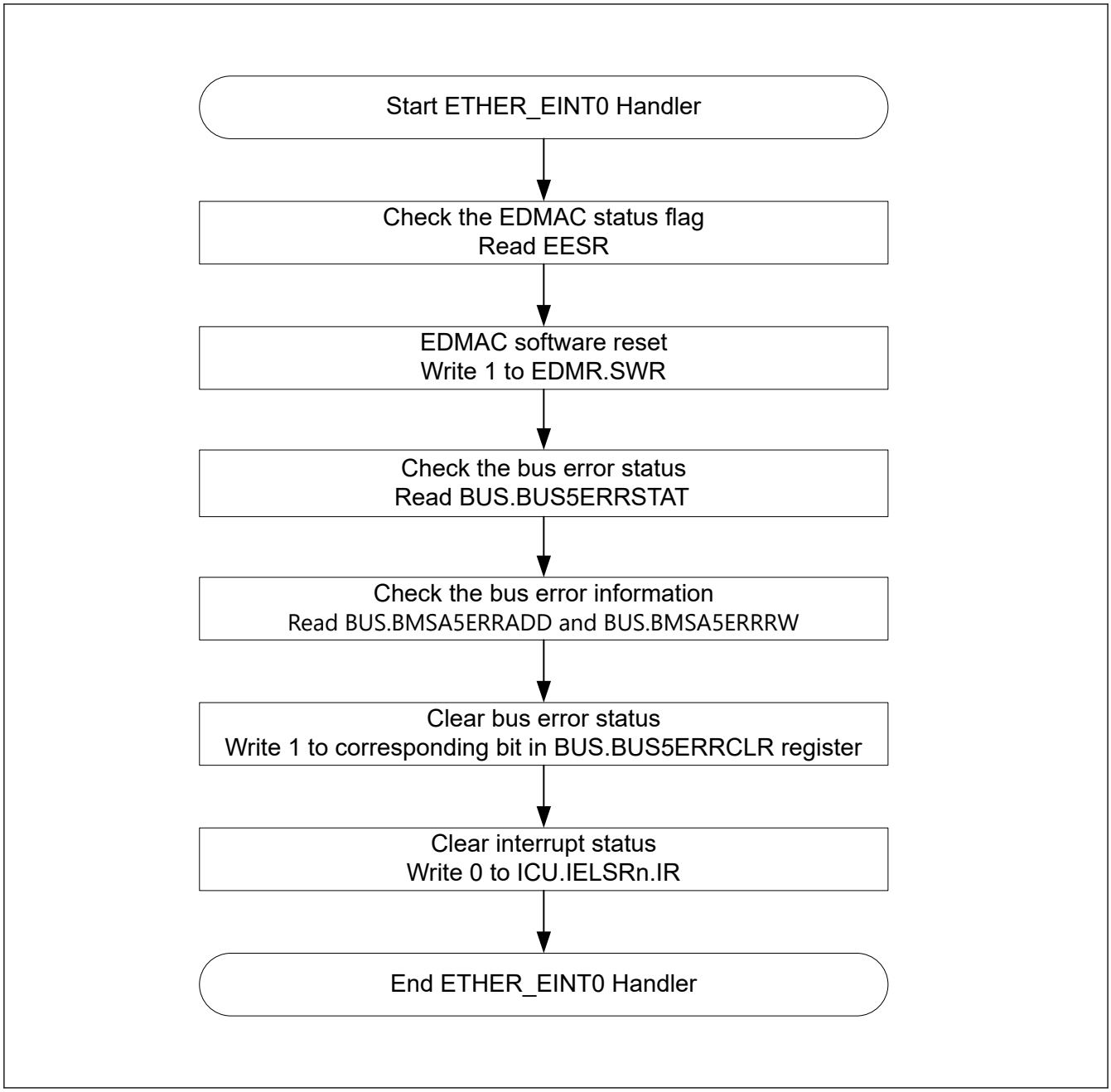


Figure 28.10 Processing for bus transfer error in EHTER_EINT0 handler

28.4 Interrupts

When any of the status flags in the EESR register sets to 1 while the associated interrupt request enable bit in the EESIPR register is 1, EDMAC0 issues an ETHER_EINT0 interrupt request.

28.5 Usage Notes

28.5.1 Settings for the Module-Stop Function

The following bit in Module Stop Control Register B (MSTPCRB) enables or disables EDMAC module operation:

- The MSTPCRB.MSTPB15 bit in enables or disables ETHERC0 and EDMAC0 operation

The modules are initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

28.5.2 Stopping the EDMAC during Operation

When stopping EDMAC operation by using a Software Standby mode or the module-stop function while the EDMAC is running, confirm that the EDTRR.TR and EDRRR.RR bits are 0. If the EDMAC is stopped while the EDTRR.TR or EDRRR.RR bit is 1, data for the frame that is being transmitted or received might not be complete, and EDMAC operation after exiting Software Standby mode or the module-stop state is not guaranteed.

29. USB 2.0 Full-Speed Module (USBFS)

29.1 Overview

The USB 2.0 Full-Speed module (USBFS) operates as a host or device controller compliant with the Universal Serial Bus (USB) specification revision 2.0. The host controller supports USB 2.0 full-speed and low-speed transfers, and the device controller supports USB 2.0 full-speed transfers. The USBFS has an internal USB transceiver and supports all of the transfer types defined in the USB 2.0 specification

The USBFS has FIFO buffer for data transfers, providing a maximum of 10 pipes. Any endpoint number can be assigned to pipes 1 to 9, based on the peripheral devices or the communication requirements for your system.

Table 29.1 lists the USBFS specifications, Figure 29.1 shows a block diagram, and Table 29.2 lists the I/O pins.

Table 29.1 USBFS specifications

Parameter	Specifications
Features	● USB Device Controller (UDC) and USB 2.0 transceiver supporting host controller, and device controller, and On-The-Go (OTG) functions ● Host and device controller can be switched by software ● Self-power and bus power mode can be used
	Host controller features: ● Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) ● Automatic scheduling for SOF and packet transmissions ● Programmable intervals for isochronous and interrupt transfers ● Communications with multiple peripheral devices connected through a single hub
	Device controller features: ● Full-speed transfer (12 Mbps)*1 ● Control transfer stage control function ● Device state control function ● Auto response function for SET_ADDRESS request ● SOF interpolation
Supported transfer types	● Control transfer ● Bulk transfer ● Interrupt transfer ● Isochronous transfer
Pipe configuration	● FIFO buffer for USB communication ● Up to 10 pipes selectable, including the Default Control Pipe (DCP) ● Pipes 1 to 9 assignable to any endpoint number
	Transfer conditions specifiable for each pipe: ● Pipe 0: Control transfer with 64-byte single buffer ● Pipes 1 and 2: Selectable to bulk transfer with 64-byte double buffer or isochronous transfer with 256-byte double buffer ● Pipes 3 to 5: Bulk transfer with 64-byte double buffer ● Pipes 6 to 9: Interrupt transfer with 64-byte single buffer
Other features	● Reception end function using transaction count ● Function that changes the BRDY interrupt event notification timing (BFRE) ● Automatic clearing of the FIFO buffer after the data for the pipe specified in the DnFIFO port (n = 0, 1) is read (DCLRM) ● NAK setting function for response PID generated on transfer end (SHTNAK) ● On-chip pull-up and pull-down resistors for D+ and D–
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security and Privilege attribution can be set

Note 1. Low-speed transfer (1.5 Mbps) is not supported.

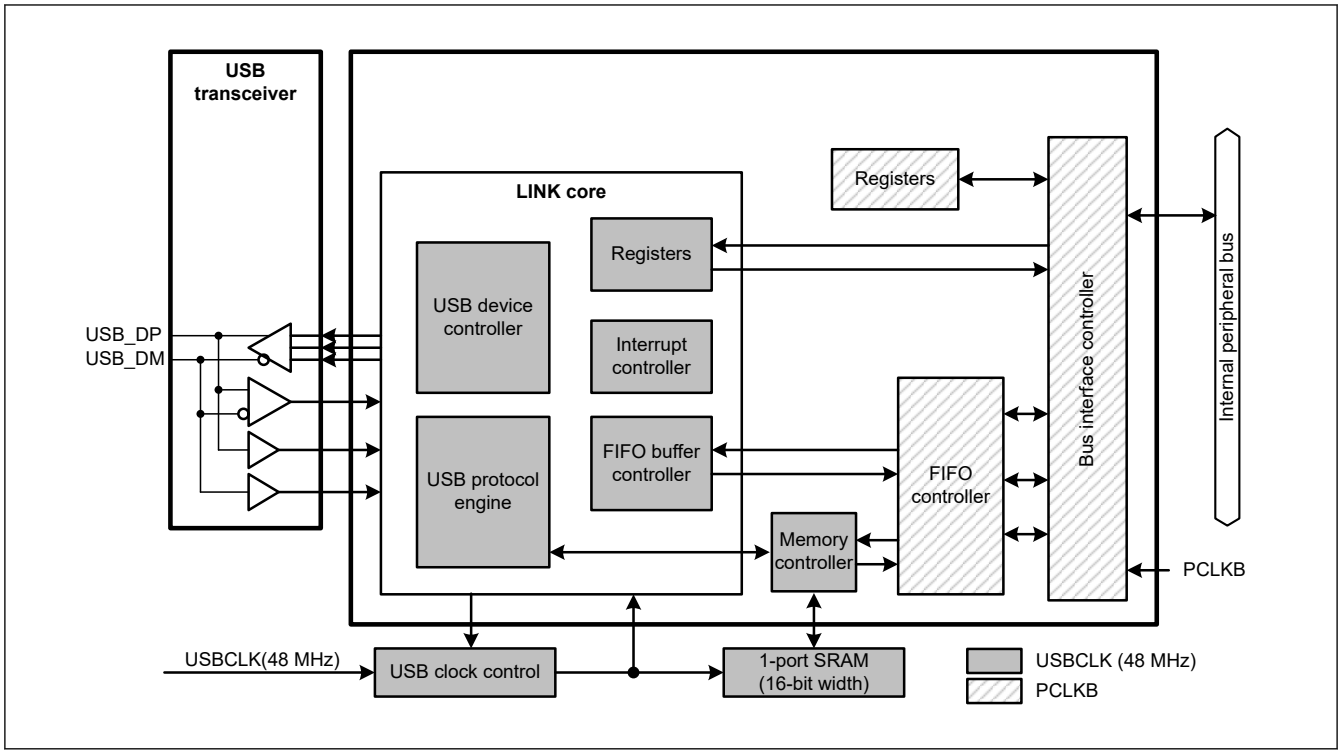


Figure 29.1 USBFS block diagram

Table 29.2 USBFS pin configuration

Function	Pin name	I/O	Description
USBFS	USB_DP	I/O	D+ I/O pin of the USB on-chip transceiver. This pin should be connected to the D+ pin of the USB bus.
	USB_DM	I/O	D– I/O pin of the USB on-chip transceiver. This pin should be connected to the D– pin of the USB bus.
	USB_VBUS	Input	USB cable connection monitor pin. This pin should be connected to VBUS of the USB bus. The VBUS pin status (connected or disconnected) can be detected when the USB module is operating as a function controller.
	USB_EXICEN	Output	Low-power control signal for external power supply (OTG) chip.
	USB_VBUSEN	Output	VBUS (5 V) supply enable signal for external power supply chip.
	USB_OVRCURA, USB_OVRCURB USB_OVRCURA-DS, USB_OVRCURB-DS	Input	External overcurrent detection signals should be connected to these pins. VBUS comparator signals should be connected to these pins when the OTG power supply chip is connected. USB_OVRCURA or USB_OVRCURB pins can be used in Software standby mode or in normal mode. USB_OVRCURA-DS, USB_OVRCURB-DS are dedicated pins that can generate interrupt in Software standby mode or in normal mode as well as for cancel Deep Software Standby mode 1.
	USB_ID	Input	MicroAB connector ID input signal should be connected to this pin during operation in OTG mode.
	VCC_USB	Input	Power supply pins.
	VSS_USB	Input	Ground pins.

29.2 Register Descriptions

29.2.1 SYSCFG : System Configuration Control Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x000

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SCKE	—	—	—	DCFM	DRPD	DPRPU	—	—	—	USBE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	USBE	USBFS Operation Enable 0: Disable 1: Enable	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	DPRPU	D+ Line Resistor Control 0: Disable line pull-up 1: Enable line pull-up	R/W
5	DRPD	D+/D– Line Resistor Control 0: Disable line pull-down 1: Enable line pull-down	R/W
6	DCFM	Controller Function Select 0: Select device controller 1: Select host controller	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
9:8	—	These bits are read as 0. The write value should be 0.	R/W
10	SCKE	USB Clock Enable 0: Stop clock supply to the USBFS 1: Enable clock supply to the USBFS	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: After writing 1 to the SCKE bit, read it to confirm that it is set to 1.

USBE bit (USBFS Operation Enable)

The USBE bit enables or disables operation of the USBFS.

Changing the USBE bit from 1 to 0 initializes the bits listed in Table 29.3. Only change this bit while the SCKE bit is 1. In host controller mode, this bit must be set to 1 after setting the DRPD bit to 1, eliminating SYSSTS0.LNST[1:0] flags chattering, and confirming that the USB bus state is stable.

Table 29.3 Registers initialized by writing 0 to the SYSCFG.USBE bit

Selected function	Register	Bit	Remarks
Device controller	SYSSTS0	LNST[1:0]	Value is saved in host controller mode
	DVSTCTR0	RHST[2:0]	—
	INTSTS0	DVSQ[2:0]	Value is saved in host controller mode
	USBADDR	USBADDR[6:0]	Value is saved in host controller mode
	USBREQ	BREQUEST[7:0], BMREQUESTTYPE[7:0]	Value is saved in host controller mode
	USBVAL	WVALUE[15:0]	Value is saved in host controller mode
	USBINDX	WINDEX[15:0]	Value is saved in host controller mode
	USBLENG	WLENTUH[15:0]	Value is saved in host controller mode
Host controller	DVSTCTR0	RHST[2:0]	—
	FRMNUM	FRNM[10:0]	Value is saved in device controller mode

DPRPU bit (D+ Line Resistor Control)

The DPRPU bit enables or disables pulling up the D+ line in device controller mode.

When the DPRPU bit is set to 1 in device controller mode, the USBFS pulls up the D+ line to notify the USB host that it attached. Changing the DPRPU bit from 1 to 0 releases the pull-up, thereby notifying the USB host that it detached.

Set this bit to 1 in device controller mode and to 0 in host controller mode.

DRPD bit (D+/D– Line Resistor Control)

TheDRPD bit enables or disables pulling down D+ and D– lines in host controller mode.

Set this bit to 1 in host controller mode and to 0 in device controller mode.

DCFM bit (Controller Function Select)

The DCFM bit selects the host or device function of the USBFS.

Only change this bit when the DPRPU and DRPD bits are both 0.

SCKE bit (USB Clock Enable)

The SCKE bit stops or enables the 48-MHz clock supply to the USBFS.

When this bit is 0, only SYSCFG is permitted to be read from and written to; the other registers related to the USB should not be read from or written to.

29.2.2 SYSSTS0 : System Configuration Status Register 0

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x004

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVCMON[1:0]		—	—	—	—	—	—	—	HTACT	SOFEA	—	—	IDMON	LNST[1:0]	
Value after reset:	x	x	0	0	0	0	0	0	0	0	0	0	0	x	0	0

Bit	Symbol	Function	R/W
1:0	LNST[1:0]	USB Data Line Status Monitor Indicates the status of the USB data lines, see Table 29.4	R
2	IDMON	External ID0 Input Pin Monitor 0: USB_ID pin is low 1: USB_ID pin is high	R
4:3	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
5	SOFEA	Active Monitor When the Host Controller Is Selected 0: SOF output stopped 1: SOF output operating	R
6	HTACT	USB Host Sequencer Status Monitor 0: Host sequencer completely stopped 1: Host sequencer not completely stopped	R
13:7	—	These bits are read as 0.	R
15:14	OVCMON[1:0]	External USB_OVRCURA, USB_OVRCURA-DS, USB_OVRCURB or USB_OVRCURB-DS Input Pin Monitor OVCMON[1] indicates the USB_OVRCURA or USB_OVRCURA-DS pin status. OVCMON[0] indicates the USB_OVRCURB or USB_OVRCURB-DS pin status.	R

Note: S-TYPE-3, P-TYPE-3
Note: The values of the OVCMON[1:0] and IDMON bits depend on the status of the USB_OVRCURA, USB_OVRCURA-DS, USB_OVRCURB or USB_OVRCURB-DS and USB_ID pins.

LNST[1:0] bits (USB Data Line Status Monitor)

The LNST[1:0] bits indicate the state of the USB data lines, D+ and D–. For details, see Table 29.4.

In device controller mode, read the LNST[1:0] bits after connection processing (SYSCFG.DPRPU bit = 1). In host controller mode, read them after enabling pull-down of the lines (SYSCFG.DRPD bit = 1).

Table 29.4 Status of the USB data bus lines (D+ and D–)

LNST[1:0] bits	During full-speed operation	During low-speed operation
00b	SE0	SE0
01b	J-State	K-State
10b	K-State	J-State
11b	SE1	SE1

SOFEA bit (Active Monitor When the Host Controller Is Selected)

The SOFEA bit is used in host controller mode to check whether the output of the last SOF is complete when the USBFS is suspended because of a 0 setting to the DVSTCTR0.UACT bit.

In host controller mode, check that both the HTACT and SOFEA bits are 0 before setting the SYSCFG.USBE bit to 0 to stop the USBFS or setting the SYSCFG.SCKE bit to 0 to stop the clock signal supply during communication.

HTACT bit (USB Host Sequencer Status Monitor)

The HTACT bit is set to 0 when the host sequencer of the USBFS is completely stopped.

In host controller mode, check that the HTACT bit is 0 before setting the DVSTCTR0.UACT bit to 0 to place the USBFS in the suspended state or setting the SYSCFG.SCKE bit to 0 to stop the clock signal supply during communication.

OVCMON[1:0] bits (External USB_OVRCURA, USB_OVRCURA-DS, USB_OVRCURB or USB_OVRCURB-DS Input Pin Monitor)

The OVCMON[1:0] bits indicate the status of the overcurrent signals from an external power supply IC.

29.2.3 DVSTCTR0 : Device State Control Register 0

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x008

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	HNPBTOA	EXICEN	VBUSEN	WKUP	RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	RHST[2:0]	USB Bus Reset Status 0 0 0: In host controller mode: Communication speed indeterminate (powered state or no connection) In device controller mode: Communication speed indeterminate 0 0 1: In host controller mode: Low-speed connection In device controller mode: USB bus reset in progress 0 1 0: In host controller mode: Full-speed connection In device controller mode: USB bus reset in progress or full-speed connection 0 1 1: Setting prohibited Others: In host controller mode: USB bus reset in progress In device controller mode: Setting prohibited	R
3	—	This bit is read as 0. The write value should be 0.	R/W
4	UACT	USB Bus Enable 0: Disable downstream port (disable SOF transmission) 1: Enable downstream port (enable SOF transmission)	R/W
5	RESUME	Resume Output 0: Do not output resume signal 1: Output resume signal	R/W
6	USBRST	USB Bus Reset Output 0: Do not output USB bus reset signal 1: Output USB bus reset signal	R/W
7	RWUPE	Wakeup Detection Enable 0: Disable downstream port remote wakeup 1: Enable downstream port remote wakeup	R/W
8	WKUP	Wakeup Output 0: Do not output remote wakeup signal 1: Output remote wakeup signal	R/W
9	VBUSEN	USB_VBUSEN Output Pin Control 0: Output low on external USB_VBUSEN pin 1: Output high on external USB_VBUSEN pin	R/W
10	EXICEN	USB_EXICEN Output Pin Control 0: Output low on external USB_EXICEN pin 1: Output high on external USB_EXICEN pin	R/W
11	HNPBTOA	Host Negotiation Protocol (HNP) Control Use this bit when switching from device B to device A in OTG mode. If the HNPBTOA bit is 1, the internal function control remains in the Suspend state until the HNP processing ends even if SYSCFG.DPRPU = 0 or SYSCFG.DCFM = 1.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The USBFS controller does not support low-speed connections in device controller mode. When this value is read, abnormal connection processing must be executed in higher level application software.

RHST[2:0] bits (USB Bus Reset Status)

RHST[2:0] bits indicate the status of the USB bus reset.

In host controller mode, writing 1 to the USBRST bit causes the RHST[2:0] bits to set to 100b. When 0 is written to the USBRST bit and the USBFS ends the SE0 state, the RHST[2:0]bits update to a new value.

In device controller mode, if the USBFS detects a USB bus reset, the RHST[2:0] bits indicate 010b if the DPRPU bit is 1, and a DVST interrupt is generated.

UACT bit (USB Bus Enable)

When set to 1 in host controller mode, the UACT bit enables USB bus operation by controlling SOF packet transmission to the USB bus in addition to data and reception. The USBFS starts SOF packet output within one frame period after the UACT bit is set to 1. When UACT is set to 0, the USBFS enters the idle state after the SOF packet output.

The USBFS sets the UACT bit to 0 on any of the following conditions:

- A DTCH interrupt is detected during communication (when UACT = 1)

- An EOFERR interrupt is detected during communication (when UACT = 1)

Always write 1 to the UACT bit at the end of the USB bus reset processing (writing 0 to the USBRST bit) or at the end of resume processing from the suspended state (writing 0 to the RESUME bit).

In device controller mode, always set this bit to 0.

RESUME bit (Resume Output)

The RESUME bit controls the resume signal output in host controller mode.

When this bit is set to 1, the USBFS drives the USB port to the K-state and outputs the resume signal. The USBFS sets the bit to 1 on detection of a remote wakeup signal while the RWUPE bit is 1 and in the USB Suspend state.

The USBFS continues outputting the K-state while the RESUME bit is 1, until the bit is cleared to 0 by software. The RESUME bit must be 1 (resume period) for the time defined in the USB 2.0 specification. Only set this bit to 1 while the interface is in the Suspend state. Write 1 to the UACT bit simultaneously with the end of the resume processing (writing 0 to the RESUME bit).

Always set this bit to 0 in device controller mode.

USBRST bit (USB Bus Reset Output)

The USBRST bit controls the output of the USB bus signal in host controller mode. When this bit set to 1, the USBFS drives the USB port to the SE0 state to reset the USB bus. The USBFS continues outputting SE0 while the USBRST bit is 1, until the bit is cleared to 0 by software. The USBRST bit must be 1 (USB bus reset period) for the time defined in the USB 2.0 specification. Writing 1 to the USBRST bit during communication (UACT = 1) or during resume processing (RESUME = 1) prevents the USBFS from starting USB bus reset processing until both the UACT and RESUME bits become 0. Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (writing 0 to the USBRST bit).

Always set this bit to 0 in device controller mode.

RWUPE bit (Wakeup Detection Enable)

The RWUPE bit enables or disables remote wakeup signals (resume signals) from downstream peripheral devices in host controller mode. When this bit is set to 1, the USBFS detects a remote wakeup signal (K-state for 2.5 µs) from a downstream peripheral device, and performs resume processing, driving the K-state. When the RWUPE bit is set to 0, the USBFS ignores remote wakeup signals (K-states) from peripheral devices connected to the USB port.

Do not stop the internal clock when the RWUPE bit is 1, even in the Suspend state (SYSCFG.SCKE bit must be set to 1).

Always set this bit to 0 in device controller mode.

WKUP bit (Wakeup Output)

The WKUP bit enables or disables remote wakeup signals (resume signals) to the USB bus in device controller mode.

The USBFS controls the output timing of the remote wakeup signals. When this bit is set to 1, the USBFS clears it to 0 after outputting the K-state for 10 ms. The USB 2.0 specification specifies that the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. If the USBFS writes 1 to the WKUP bit immediately after detecting the Suspend state, the K-state is output after 2 ms.

Only write 1 to the WKUP bit when the device is in the Suspend state (INTSTS0.DVSQ[2:0] = 1xxb) and the USB host enables the remote wakeup signal. Do not stop the internal clock while this bit is 1, even in the Suspend state (SYSCFG.SCKE bit must be set to 1).

Always set this bit to 0 in host controller mode.

HNPBTOA bit (Host Negotiation Protocol (HNP) Control)

TheHNPBTOA bit is used when switching from device B to device A while in OTG mode.

If the HNPBTOA bit is 1, the internal function control maintains the Suspend state until HNP processing ends, even if the SYSCFG.DPRPU bit is set to 0 or the SYSCFG.DCFM bit is set to 1. Resume interrupts (RESM) are not generated even if a falling edge of D+ is detected.

After this bit is set to 1, the HNP processing ends when a host attach event is detected, because of a pull-up by the initiating party, or the HNPBTOA bit is cleared to 0 by software because the HNP processing times out.

29.2.4 CFIFO/CFIFOL : CFIFO Port Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x014

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FIFOPORT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	FIFOPORT[15:0]*1	FIFO Port Read receive data from the FIFO buffer or write transmit data to the FIFO buffer by accessing these bits	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The valid bits depend on the MBW settings (CFIFOSEL.MBW) and BIGEND settings (CFIFOSEL.BIGEND) in the associated port selection register. See Table 29.5 and Table 29.6.

Three FIFO ports are available:

- CFIFO
- D0FIFO
- D1FIFO

Each FIFO port is configured with:

- A port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer and writing of data to the FIFO buffer
- A port selection register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that selects the pipe assigned to the FIFO port
- A port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR)

Each FIFO port has the following constraints:

- Access to the FIFO buffer for DCP control transfers is through the CFIFO port
- Access to the FIFO buffer for DMA or DTC transfers is through the D0FIO or D1FIFO port
- The D0FIFO and D1FIFO ports can also be accessed by the CPU
- When using functions specific to the FIFO port, such as the DMA or DTC transfer function, you cannot change the pipe number selected in the CURPIPE[3:0] bits of the port selection register
- Registers configuring a FIFO port do not affect other FIFO ports
- The same pipe must not be assigned to two or more FIFO ports
- There are two FIFO buffer states, one giving access rights to the CPU and the other to the serial interface engine (SIE). When the SIE has access rights, the FIFO buffer cannot be accessed by the CPU

FIFOPORT[15:0] bits (FIFO Port)

When the FIFOPORT[15:0] bit is accessed, the USBFS reads the received data from the FIFO buffer or writes the transmit data to the FIFO buffer. The FIFO port register can be accessed only when the FRDY bit in the associated port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in the FIFO port register depend on the MBW and BIGEND settings in the port selection register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL). See Table 29.5 and Table 29.6.

Table 29.5 Endian operation in 16-bit access

CFIFOSEL.BIGEND bit	Bits [15:8]	Bits [7:0]
0	N + 1 data	N + 0 data
1	N + 0 data	N + 1 data

Table 29.6 Endian operation in 8-bit access

CFIFOSEL.BIGEND bit	Bits [15:8]	Bits [7:0]
0	Access prohibited*1	N + 0 data
1	Access prohibited*1	N + 0 data

Note 1. Writing to or reading from these areas is not allowed.

29.2.5 DnFIFO/DnFIFOL : DnFIFO Port Register (n = 0, 1)

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x018 + 0x4 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FIFOPORT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	FIFOPORT[15:0]*1	FIFO Port Read receive data from the FIFO buffer or write transmit data to the FIFO buffer by accessing these bits	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The valid bits depend on the MBW settings (CFIFOSEL.MBW, D0FIFOSEL.MBW, and D1FIFOSEL.MBW) and BIGEND settings (CFIFOSEL.BIGEND, D0FIFOSEL.BIGEND, and D1FIFOSEL.BIGEND) in the associated port selection register. See Table 29.7 and Table 29.8.

Three FIFO ports are available:

- CFIFO
- D0FIFO
- D1FIFO

Each FIFO port is configured with:

- A port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer and writing of data to the FIFO buffer
- A port selection register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that selects the pipe assigned to the FIFO port
- A port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR)

Each FIFO port has the following constraints:

- Access to the FIFO buffer for DCP control transfers is through the CFIFO port
- Access to the FIFO buffer for DMA or DTC transfers is through the D0FIO or D1FIFO port
- The D0FIFO and D1FIFO ports can also be accessed by the CPU
- When using functions specific to the FIFO port, such as the DMA or DTC transfer function, you cannot change the pipe number selected in the CURPIPE[3:0] bits of the port selection register
- Registers configuring a FIFO port do not affect other FIFO ports
- The same pipe must not be assigned to two or more FIFO ports
- There are two FIFO buffer states, one giving access rights to the CPU and the other to the serial interface engine (SIE). When the SIE has access rights, the FIFO buffer cannot be accessed by the CPU

FIFOPORT[15:0] bits (FIFO Port)

When the FIFOPORT bit is accessed, the USBFS reads the received data from the FIFO buffer or writes the transmit data to the FIFO buffer. The FIFO port register can be accessed only when the FRDY bit in the associated port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in the FIFO port register depend on the MBW and BIGEND settings in the port selection register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL). See Table 29.7 and Table 29.8.

Table 29.7 Endian operation in 16-bit access

CFIFOSEL.BIGEND bit D0FIFOSEL.BIGEND bit D1FIFOSEL.BIGEND bit	Bits [15:8]	Bits [7:0]
0	N + 1 data	N + 0 data
1	N + 0 data	N + 1 data

Table 29.8 Endian operation in 8-bit access

CFIFOSEL.BIGEND bit D0FIFOSEL.BIGEND bit D1FIFOSEL.BIGEND bit	Bits [15:8]	Bits [7:0]
0	Access prohibited*1	N + 0 data
1	Access prohibited*1	N + 0 data

Note 1. Writing to or reading from these areas is not allowed.

29.2.6 CFIFOSEL : CFIFO Port Select Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x020

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RCNT	REW	—	—	—	MBW	—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CURPIPE[3:0]	CFIFO Port Access Pipe Specification 0x0: Default Control Pipe 0x1: Pipe 1 0x2: Pipe 2 0x3: Pipe 3 0x4: Pipe 4 0x5: Pipe 5 0x6: Pipe 6 0x7: Pipe 7 0x8: Pipe 8 0x9: Pipe 9 Others: Setting prohibited	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	ISEL	CFIFO Port Access Direction When DCP Is Selected 0: Select reading from the FIFO buffer 1: Select writing to the FIFO buffer	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	BIGEND	CFIFO Port Endian Control 0: Little endian 1: Big endian	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	MBW	CFIFO Port Access Bit Width 0: 8-bit width 1: 16-bit width	R/W
13:11	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
14	REW	Buffer Pointer Rewind 0: Do not rewind buffer pointer 1: Rewind buffer pointer	W*1
15	RCNT	Read Count Mode 0: The DTLN[8:0] bits (CFIFOCTR.DTLN[8:0], D0FIFOCTR.DTLN[8:0], D1FIFOCTR.DTLN[8:0]) are cleared when all receive data is read from the CFIFO. In double buffer mode, the DTLN[8:0] value is cleared when all data is read from only a single plane. 1: The DTLN[8:0] bits are decremented each time the receive data is read from the CFIFO.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Only 0 can be read.

Do not specify the same pipe number in the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

Do not change the pipe number while DMA or DTC transfer is enabled.

CURPIPE[3:0] bits (CFIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number to use for reading or writing data through the CFIFO port. After writing to these bits, read them to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

During FIFO buffer access, even when an attempt is made to change the CURPIPE[3:0] setting, the current access setting is retained until access is complete.

ISEL bit (CFIFO Port Access Direction When DCP Is Selected)

After writing a new value to the ISEL bit with the DCP as the selected pipe, read the ISEL bit to check that the written value agrees with the read value before proceeding to the next process. Set the ISEL and CURPIPE[3:0] bits simultaneously.

MBW bit (CFIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the CFIFO port.

When the selected pipe is receiving, set the CURPIPE[3:0] and MBW bits simultaneously. After a write to these bits starts a data read from the FIFO buffer, do not change the MBW bit until all of the data is read.

When the selected pipe is transmitting, the bit width cannot be changed from 8-bit to 16-bit while data is being written to the FIFO buffer.

An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

REW bit (Buffer Pointer Rewind)

The REW bit specifies whether to rewind the buffer pointer.

When the selected pipe is receiving, setting this bit to 1 while the FIFO buffer is being read allows re-reading of the FIFO buffer from the first data. In double buffering, this setting enables re-reading of the currently-read FIFO buffer plane from the first entry.

Do not set this bit to 1 while simultaneously changing the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY bit is 1.

To rewrite to the FIFO buffer from the first data for the transmitting pipe, use the BCLR bit.

29.2.7 DnFIFOSEL : DnFIFO Port Select Register (n = 0, 1)

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x028 + 0x4 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RCNT	REW	DCLRM	DREQE	—	MBW	—	BIGEND	—	—	—	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CURPIPE[3:0]	FIFO Port Access Pipe Specification 0x0: Default Control Pipe 0x1: Pipe 1 0x2: Pipe 2 0x3: Pipe 3 0x4: Pipe 4 0x5: Pipe 5 0x6: Pipe 6 0x7: Pipe 7 0x8: Pipe 8 0x9: Pipe 9 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	BIGEND	FIFO Port Endian Control 0: Little endian 1: Big endian	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	MBW	FIFO Port Access Bit Width 0: 8-bit width 1: 16-bit width	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	DREQE	DMA/DTC Transfer Request Enable 0: Disable DMA/DTC transfer request 1: Enable DMA/DTC transfer request	R/W
13	DCLRM	Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read 0: Disable auto buffer clear mode 1: Enable auto buffer clear mode	R/W
14	REW	Buffer Pointer Rewind 0: Do not rewind buffer pointer 1: Rewind buffer pointer	W
15	RCNT	Read Count Mode 0: Clear DTLN[8:0] bits in (CFIFOCTR.DTLN[8:0], D0FIFOCTR.DTLN[8:0], D1FIFOCTR.DTLN[8:0]) when all receive data is read from DnFIFO (after read of a single plane in double buffer mode) 1: Decrement DTLN[8:0] bits each time receive data is read from DnFIFO	R/W

Note: S-TYPE-3, P-TYPE-3

The same pipe must not be specified in the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected. The pipe number must not be changed while DMA or DTC transfer is enabled.

CURPIPE[3:0] bits (FIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number to use for reading or writing data through the DnFIFO port. After writing to these bits, read them to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

During FIFO buffer access, even when an attempt is made to change the CURPIPE[3:0] setting, the current access setting is retained until access is complete.

MBW bit (FIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the DnFIFO port.

When the selected pipe is receiving, after a write to these bits starts a data read from the FIFO buffer, do not change the MBW bit until all of the data is read. Set the CURPIPE[3:0] and MBW bits simultaneously.

When the selected pipe is transmitting, the bit width cannot be changed from 8-bit to 16-bit while data is being written to the FIFO buffer.

An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

DREQE bit (DMA/DTC Transfer Request Enable)

The DREQE bit enables or disables issuing of DMA or DTC transfer requests. To enable DMA or DTC transfer requests, set this bit to 1 after setting the CURPIPE[3:0] bits. To change the CURPIPE[3:0] setting, first set this bit to 0.

DCLRM bit (Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read)

The DCLRM bit enables or disables automatic FIFO buffer clearing after data in the selected pipe is read.

When this bit is set to 1, on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or when reading of a received short packet is complete while the PIPECFG.BFRE bit is 1, the USBFS sets the BCLR bit in the FIFO port control register to 1.

When using the USBFS with the SOFCFG.BRDYM bit set to 1, set this bit to 0.

REW bit (Buffer Pointer Rewind)

The REW bit specifies whether to rewind the buffer pointer.

When the selected pipe is receiving, setting this bit to 1 while the FIFO buffer is being read allows re-reading of the FIFO buffer from the first data. In double buffering, this setting enables re-reading of the currently-read FIFO buffer plane from the first entry.

Do not set this bit to 1 while simultaneously changing the CURPIPE[3:0] bits. Before setting the bit to 1, be sure to check that the FRDY bit is 1.

To rewrite to the FIFO buffer from the first data for the transmitting pipe, use the BCLR bit.

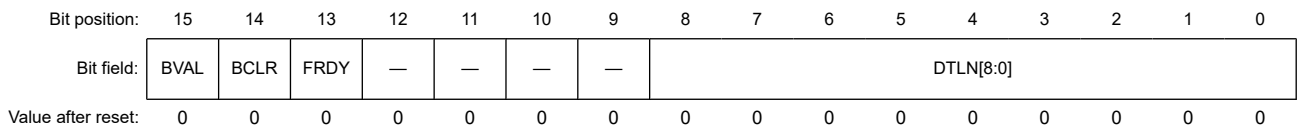
RCNT bit (Read Count Mode)

The RCNT bit specifies the read mode for the value in the D0FIFOCTL.DTLN bit and D1FIFOCTL.DTLN bit. When accessing DnFIFO with the PIPECFG.BFRE bit set to 1, set the RCNT bit to 0.

29.2.8 CFIFOCTR : CFIFO Port Control Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x022



Bit	Symbol	Function	R/W
8:0	DTLN[8:0]	Receive Data Length Indicates the receive data length. The meaning of the values differs depending on the RCNT bit setting in the port select register. For details, see the description of the DTLN[8:0] bits.	R
12:9	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
13	FRDY	FIFO Port Ready 0: FIFO port access disabled 1: FIFO port access enabled	R
14	BCLR	CPU Buffer Clear 0: No operation 1: Clear FIFO buffer on the CPU side	W
15	BVAL	Buffer Memory Valid Flag 0: Invalid (writing 0 has no effect) 1: Writing ended	R/W

Note: S-TYPE-3, P-TYPE-3

The CFIFOCTR, D0FIFOCTR, and D1FIFOCTR registers correspond to the CFIFO, D0FIFO, and D1FIFO buffers.

DTLN[8:0] bits (Receive Data Length)

The DTLN[8:0] bits indicate the length of the receive data.

While the FIFO buffer is being read, the DTLN[8:0] bits indicate different values depending on the DnFIFOSEL.RCNT bit (n = 0, 1), as follows:

- RCNT = 0

  The USBFS sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU or DMA/DTC has read all of the received data from a single FIFO buffer plane.

  While the PIPECFG.BFRE bit = 1, the USBFS retains the length of the receive data until the BCLR bit is set to 1, even after all the data is read.

- RCNT = 1

  The USBFS decrements the value indicated in the DTLN[8:0] bits each time data is read from the FIFO buffer. The value is decremented by 1 when MBW = 0, and by 2 when MBW = 1.

  The USBFS sets these bits to 0 when all the data is read from one FIFO buffer plane. In double buffer mode, if data is received in one FIFO buffer plane before all of the data is read from the other plane, the USBFS sets these bits to indicate the length of the receive data in the former plane when all of the data is read from the latter plane.

FRDY bit (FIFO Port Ready)

The FRDY bit indicates whether the FIFO port can be accessed by the CPU or DMA/DTC.

In the following cases, the USBFS sets the FRDY bit to 1 but data cannot be read through the FIFO port because there is no data to be read:

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty
- A short packet is received and the data is completely read while the PIPECFG.BFRE bit = 1

In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

BCLR bit (CPU Buffer Clear)

Set the BCLR bit to 1 to clear the FIFO buffer on the CPU side for the selected pipe.

When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USBFS clears only one plane of the FIFO buffer even when both planes are read-enabled.

When the DCP is the selected pipe, setting the BCLR bit to 1 allows the USBFS to clear the FIFO buffer regardless of whether the CPU or SIE has access rights. To clear the buffer when the SIE has access rights, set the DCPCTR.PID[1:0] bits to 00b (NAK response) before setting the BCLR bit to 1.

When the selected pipe is transmitting, if 1 is written to the BVAL flag and the BCLR bit simultaneously, the USBFS clears the data that is already written, enabling transmission of a zero-length packet.

When the selected pipe is not the DCP, only write 1 to the BCLR bit while the FRDY bit in the FIFO port control register is 1 (set by the USBFS).

BVAL flag (Buffer Memory Valid Flag)

Set the BVAL flag to 1 when data is completely written to the FIFO buffer on the CPU side for the pipe selected in CURPIPE[3:0].

When the selected pipe is transmitting, set this flag to 1 in the following cases:

- To transmit a short packet, set this flag to 1 after data is written
- To transmit a zero-length packet, set this flag to 1 before data is written to the FIFO buffer

The USBFS then switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

When data of the maximum packet size is written for the pipe in continuous transfer mode, the USBFS sets the BVAL flag to 1 and switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

Only write 1 to the BVAL flag while the FRDY bit is 1 (set by the USBFS). When the selected pipe is receiving, do not set the BVAL flag to 1.

29.2.9 DnFIFOCTR : DnFIFO Port Control Register (n = 0, 1)

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x02A + 0x4 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	DTLN[8:0]	Receive Data Length Indicates the receive data length. The meaning of the values differs depending on the RCNT bit setting in the port select register. For details, see the description of the DTLN[8:0] bits.	R
12:9	—	These bits are read as 0. The write value should be 0.	R/W
13	FRDY	FIFO Port Ready 0: FIFO port access disabled 1: FIFO port access enabled	R
14	BCLR	CPU Buffer Clear 0: No operation 1: Clear FIFO buffer on the CPU side	R/W*1
15	BVAL	Buffer Memory Valid Flag 0: Invalid (writing 0 has no effect) 1: Writing ended	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Only 0 can be read.

The CFIFOCTR, D0FIFOCTR, and D1FIFOCTR registers correspond to the CFIFO, D0FIFO, and D1FIFO buffers.

DTLN[8:0] bits (Receive Data Length)

The DTLN[8:0] bits indicate the length of the receive data.

While the FIFO buffer is being read, the DTLN[8:0] bits indicate different values depending on the DnFIFOSEL.RCNT bit (n = 0, 1), as follows:

- RCNT = 0

  The USBFS sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU or DMA/DTC has read all of the received data from a single FIFO buffer plane.

  While the PIPECFG.BFRE bit = 1, the USBFS retains the length of the receive data until the BCLR bit is set to 1, even after all the data is read.

- RCNT = 1

The USBFS decrements the value indicated in the DTLN[8:0] bits each time data is read from the FIFO buffer. The value is decremented by 1 when MBW = 0, and by 2 when MBW = 1.

The USBFS sets these bits to 0 when all the data is read from one FIFO buffer plane. In double buffer mode, if data is received in one FIFO buffer plane before all of the data is read from the other plane, the USBFS sets these bits to indicate the length of the receive data in the former plane when all of the data is read from the latter plane.

FRDY bit (FIFO Port Ready)

The FRDY bit indicates whether the FIFO port can be accessed by the CPU or DMA/DTC.

In the following cases, the USBFS sets the FRDY bit to 1 but data cannot be read through the FIFO port because there is no data to be read:

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty
- A short packet is received and the data is completely read while the PIPECFG.BFRE bit = 1

In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

BCLR bit (CPU Buffer Clear)

Set the BCLR bit to 1 to clear the FIFO buffer on the CPU side for the selected pipe.

When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USBFS clears only one plane of the FIFO buffer even when both planes are read-enabled.

When the DCP is the selected pipe, setting the BCLR bit to 1 allows the USBFS to clear the FIFO buffer regardless of whether the CPU or SIE has access rights. To clear the buffer when the SIE has access rights, set the DCPCTR.PID[1:0] bits to 00b (NAK response) before setting the BCLR bit to 1.

When the selected pipe is transmitting, if 1 is written to the BVAL flag and the BCLR bit simultaneously, the USBFS clears the data that is already written, enabling transmission of a zero-length packet.

When the selected pipe is not the DCP, only write 1 to the BCLR bit while the FRDY bit in the FIFO port control register is 1 (set by the USBFS).

BVAL flag (Buffer Memory Valid Flag)

Set the BVAL flag to 1 when data is completely written to the FIFO buffer on the CPU side for the pipe selected in CURPIPE[3:0].

When the selected pipe is transmitting, set this flag to 1 in the following cases:

- To transmit a short packet, set this flag to 1 after data is written
- To transmit a zero-length packet, set this flag to 1 before data is written to the FIFO buffer

The USBFS then switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

When data of the maximum packet size is written for the pipe in continuous transfer mode, the USBFS sets the BVAL flag to 1 and switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

Only write 1 to the BVAL flag while the FRDY bit is 1 (set by the USBFS). When the selected pipe is receiving, do not set the BVAL flag to 1.

29.2.10 INTENB0 : Interrupt Enable Register 0

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x030

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	BRDYE	Buffer Ready Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
9	NRDYE	Buffer Not Ready Response Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
10	BEMPE	Buffer Empty Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
11	CTRE	Control Transfer Stage Transition Interrupt Enable *1 0: Disable interrupt request 1: Enable interrupt request	R/W
12	DVSE	Device State Transition Interrupt Enable *1 0: Disable interrupt request 1: Enable interrupt request	R/W
13	SOFE	Frame Number Update Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
14	RSME	Resume Interrupt Enable*1 0: Disable interrupt request 1: Enable interrupt request	R/W
15	VBSE	VBUS Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The RSME, DVSE, and CTRE bits can only be set to 1 in device controller mode. Do not set these bits to 1 in host controller mode.

When a status flag in the INTSTS0 register sets to 1 and the associated interrupt request enable bit setting in theINTENB0 register is 1, the USBFS issues a USBFS interrupt request.

Regardless of the INTENB0 register setting, the status flag in the INTSTS0 register sets to 1 in response to a state change that satisfies the associated condition.

When an interrupt request enable bit in the INTENB0 register is switched from 0 to 1 while the associated status flag in the INTSTS0 register is set to 1, a USBFS interrupt is requested.

29.2.11 INTENB1 : Interrupt Enable Register 1

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x032

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVRCRE	BCHGE	—	DTCHE	ATTCHE	—	—	—	—	EOFERRE	SIGNE	SACKE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	SACKE	Setup Transaction Normal Response Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
5	SIGNE	Setup Transaction Error Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W

Bit	Symbol	Function	R/W
6	EOFERRE	EOF Error Detection Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
10:7	—	These bits are read as 0. The write value should be 0.	R/W
11	ATTCHE	Connection Detection Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
12	DTCHE	Disconnection Detection Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
14	BCHGE	USB Bus Change Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
15	OVRCRE	Overcurrent Input Change Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W

Note: S-TYPE-3, P-TYPE-3
Note: The bits in INTENB1 can only be set to 1 in host controller mode. Do not set these bits to 1 in device controller mode.

INTENB1 specifies the interrupt masks in host controller mode and for the setup transaction.

When a status flag in the INTSTS1 register sets to 1 and the associated interrupt request enable bit setting in theINTENB1 register is 1, the USBFS issues a USBFS interrupt request.

Regardless of the INTENB1 register setting, the status flag in the INTSTS1 register sets to 1 in response to a state change that satisfies the associated condition.

When an interrupt request enable bit in the INTENB1 register is switched from 0 to 1 while the associated status flag in the INTSTS1 register is set to 1, a USBFS interrupt is requested.

Do not enable interrupts in device controller mode.

29.2.12 BRDYENB : BRDY Interrupt Enable Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x036

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPE9BRDYE	PIPE8BRDYE	PIPE7BRDYE	PIPE6BRDYE	PIPE5BRDYE	PIPE4BRDYE	PIPE3BRDYE	PIPE2BRDYE	PIPE1BRDYE	PIPE0BRDYE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIPE0BRDYE	BRDY Interrupt Enable for Pipe 0 0: Disable interrupt request 1: Enable interrupt request	R/W
1	PIPE1BRDYE	BRDY Interrupt Enable for Pipe 1 0: Disable interrupt request 1: Enable interrupt request	R/W
2	PIPE2BRDYE	BRDY Interrupt Enable for Pipe 2 0: Disable interrupt request 1: Enable interrupt request	R/W
3	PIPE3BRDYE	BRDY Interrupt Enable for Pipe 3 0: Disable interrupt request 1: Enable interrupt request	R/W

Bit	Symbol	Function	R/W
4	PIPE4BRDYE	BRDY Interrupt Enable for Pipe 4 0: Disable interrupt request 1: Enable interrupt request	R/W
5	PIPE5BRDYE	BRDY Interrupt Enable for Pipe 5 0: Disable interrupt request 1: Enable interrupt request	R/W
6	PIPE6BRDYE	BRDY Interrupt Enable for Pipe 6 0: Disable interrupt request 1: Enable interrupt request	R/W
7	PIPE7BRDYE	BRDY Interrupt Enable for Pipe 7 0: Disable interrupt request 1: Enable interrupt request	R/W
8	PIPE8BRDYE	BRDY Interrupt Enable for Pipe 8 0: Disable interrupt request 1: Enable interrupt request	R/W
9	PIPE9BRDYE	BRDY Interrupt Enable for Pipe 9 0: Disable interrupt request 1: Enable interrupt request	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The BRDYENB register enables or disables the INTSTS0.BRDY bit to be set to 1 when a BRDY interrupt is detected for each pipe.

When a status flag in the BRDYSTS register sets to 1 and the associated PIPEnBRDYE bit (n = 0 to 9) setting in theBRDYENB register is 1, the INTSTS0.BRDY flag sets to 1. In this case, if the BRDYE bit in INTENB0 is 1, the USBFS generates a BRDY interrupt request. While at least one PIPEnBRDY bit indicates 1, the USB generates the BRDY interrupt request when the associated interrupt request enable bit in the BRDYENB register is changed from 0 to 1 by software.

29.2.13 NRDYENB : NRDY Interrupt Enable Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x038

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPE9NRDYE	PIPE8NRDYE	PIPE7NRDYE	PIPE6NRDYE	PIPE5NRDYE	PIPE4NRDYE	PIPE3NRDYE	PIPE2NRDYE	PIPE1NRDYE	PIPE0NRDYE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIPE0NRDYE	NRDY Interrupt Enable for Pipe 0 0: Disable interrupt request 1: Enable interrupt request	R/W
1	PIPE1NRDYE	NRDY Interrupt Enable for Pipe 1 0: Disable interrupt request 1: Enable interrupt request	R/W
2	PIPE2NRDYE	NRDY Interrupt Enable for Pipe 2 0: Disable interrupt request 1: Enable interrupt request	R/W
3	PIPE3NRDYE	NRDY Interrupt Enable for Pipe 3 0: Disable interrupt request 1: Enable interrupt request	R/W
4	PIPE4NRDYE	NRDY Interrupt Enable for Pipe 4 0: Disable interrupt request 1: Enable interrupt request	R/W

Bit	Symbol	Function	R/W
5	PIPE5NRDYE	NRDY Interrupt Enable for Pipe 5 0: Disable interrupt request 1: Enable interrupt request	R/W
6	PIPE6NRDYE	NRDY Interrupt Enable for Pipe 6 0: Disable interrupt request 1: Enable interrupt request	R/W
7	PIPE7NRDYE	NRDY Interrupt Enable for Pipe 7 0: Disable interrupt request 1: Enable interrupt request	R/W
8	PIPE8NRDYE	NRDY Interrupt Enable for Pipe 8 0: Disable interrupt request 1: Enable interrupt request	R/W
9	PIPE9NRDYE	NRDY Interrupt Enable for Pipe 9 0: Disable interrupt request 1: Enable interrupt request	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The NRDYENB register enables or disables the INTSTS0.NRDY bit to be set to 1 when a NRDY interrupt is detected for each pipe.

When a status flag in the NRDYSTS register sets to 1 and the associated PIPEnNRDYE (n = 0 to 9) bit setting in theNRDYENB register is 1, the INTSTS0.NRDY flag sets to 1. In this case, if the NRDYE bit in INTENB0 is 1, the USBFS generates a NRDY interrupt request. While at least one PIPEnNRDY bit indicates 1, the USBFS generates the NRDY interrupt request when the associated interrupt request enable bit in the NRDYENB register is changed from 0 to 1 by software.

29.2.14 BEMPENB : BEMP Interrupt Enable Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x03A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPE9BEMPE	PIPE8BEMPE	PIPE7BEMPE	PIPE6BEMPE	PIPE5BEMPE	PIPE4BEMPE	PIPE3BEMPE	PIPE2BEMPE	PIPE1BEMPE	PIPE0BEMPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIPE0BEMPE	BEMP Interrupt Enable for Pipe 0 0: Disable interrupt request 1: Enable interrupt request	R/W
1	PIPE1BEMPE	BEMP Interrupt Enable for Pipe 1 0: Disable interrupt request 1: Enable interrupt request	R/W
2	PIPE2BEMPE	BEMP Interrupt Enable for Pipe 2 0: Disable interrupt request 1: Enable interrupt request	R/W
3	PIPE3BEMPE	BEMP Interrupt Enable for Pipe 3 0: Disable interrupt request 1: Enable interrupt request	R/W
4	PIPE4BEMPE	BEMP Interrupt Enable for Pipe 4 0: Disable interrupt request 1: Enable interrupt request	R/W

Bit	Symbol	Function	R/W
5	PIPE5BEMPE	BEMP Interrupt Enable for Pipe 5 0: Disable interrupt request 1: Enable interrupt request	R/W
6	PIPE6BEMPE	BEMP Interrupt Enable for Pipe 6 0: Disable interrupt request 1: Enable interrupt request	R/W
7	PIPE7BEMPE	BEMP Interrupt Enable for Pipe 7 0: Disable interrupt request 1: Enable interrupt request	R/W
8	PIPE8BEMPE	BEMP Interrupt Enable for Pipe 8 0: Disable interrupt request 1: Enable interrupt request	R/W
9	PIPE9BEMPE	BEMP Interrupt Enable for Pipe 9 0: Disable interrupt request 1: Enable interrupt request	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The BEMPENB register enables or disables the INTSTS0.BEMP bit to be set to 1 when a BEMP interrupt is detected for each pipe.

When a status flag in the BEMPSTS register sets to 1 and the associated PIPEnBEMPE (n = 0 to 9) bit setting in the BEMPENB register is 1, the INTSTS0.BEMP flag sets to 1. In this case, if the BEMPE bit in INTENB0 is 1, the USBFS generates a BEMP interrupt request. While at least one PIPEnBEMP bit indicates 1, the USBFS generates the BEMP interrupt request when the associated interrupt request enable bit in the BEMPENB register is changed from 0 to 1 by software.

29.2.15 SOFCFG : SOF Output Configuration Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x03C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TRNENSEL	—	BRDYM	—	EDGESTS	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	EDGESTS	Edge Interrupt Output Status Monitor*1 Indicates 1 during the edge processing of an edge interrupt output signal.	R
5	—	This bit is read as 0. The write value should be 0.	R/W
6	BRDYM	BRDY Interrupt Status Clear Timing 0: Clear BRDY flag by software 1: Clear BRDY flag by the USBFS through a data read from the FIFO buffer or data write to the FIFO buffer	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	TRNENSEL	Transaction-Enabled Time Select*1 0: Not low-speed communication 1: Low-speed communication	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Confirm that the EDGESTS flag is 0 before stopping the clock supply to the USBFS.

EDGESTS bit (Edge Interrupt Output Status Monitor)

The EDGESTS bit indicates 1 during the edge processing of an edge interrupt output signal. Confirm that this bit is 0 before stopping the clock supply to the USBFS.

BRDYM bit (BRDY Interrupt Status Clear Timing)

The BRDYM bit specifies how the BRDY interrupt status flags for the pipes are cleared.

TRNENSEL bit (Transaction-Enabled Time Select)

When the USB port is in use for full- or low-speed communications, the TRNENSEL bit specifies the timing with which the USBFS issues tokens in a frame (transaction-enabled time).

Set this bit to 1 when a low-speed device is connected. The bit is only valid in host controller mode. Set this bit to 0 in device controller mode.

29.2.16 INTSTS0 : Interrupt Status Register 0

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x040

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
Value after reset:	0	0	0	x	0	0	0	0	x	0	0	x	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CTSQ[2:0]	Control Transfer Stage 0 0 0: Idle or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error	R
3	VALID	USB Request Reception 0: Setup packet not received 1: Setup packet received	R/W
6:4	DVSQ[2:0]	Device State Indicates the device state. 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state Others: Suspend state	R
7	VBSTS	VBUS Input Status 0: USB_VBUS pin is low 1: USB_VBUS pin is high	R
8	BRDY	Buffer Ready Interrupt Status 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R
9	NRDY	Buffer Not Ready Interrupt Status 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R
10	BEMP	Buffer Empty Interrupt Status 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R
11	CTRT	Control Transfer Stage Transition Interrupt Status*2 0: No control transfer stage transition interrupt occurred 1: Control transfer stage transition interrupt occurred	R/W*1

Bit	Symbol	Function	R/W
12	DVST	Device State Transition Interrupt Status*2 0: No device state transition interrupt occurred 1: Device state transition interrupt occurred	R/W*1
13	SOFR	Frame Number Refresh Interrupt Status 0: No SOF interrupt occurred 1: SOF interrupt occurred	R/W*1
14	RESM	Resume Interrupt Status*2 *3 0: No resume interrupt occurred 1: Resume interrupt occurred	R/W*1
15	VBINT	VBUS Interrupt Status*3 0: No VBUS interrupt occurred 1: VBUS interrupt occurred	R/W*1

Note: S-TYPE-3, P-TYPE-3
Note: The value of the DVST bit is 0 when the MCU is reset and 1 after a USB bus reset.
Note: The value of the VBSTS bit is 1 when the USB_VBUS pin is high and 0 when the USB_VBUS pin is low.
Note: The value of the DVSQ[2:0] bits is 000b when the MCU is reset and 001b after a USB bus reset.
Note 1. To clear the VBINT, RESM, SOFR, DVST, CTRT, or VALID bits, write 0 only to the bits to be cleared. Write 1 to the other bits. Do not write 0 to the status bits indicating 0.
Note 2. The status of the RESM, DVST, and CTRT bits are changed only in device controller mode. Set the associated interrupt enable bits to 0 (disabled) in host controller mode.
Note 3. The USBFS detects a change in the status indicated in the VBINT andRESM bits even while the clock supply is stopped (SYSCFG.SCKE bit = 0), and it requests the interrupt when the associated interrupt request bit is 1. Enable the clock supply before clearing the status by software.

CTSQ[2:0] bits (Control Transfer Stage)

In host controller mode, the read value of the CTSQ[2:0] bits is invalid.

VALID bit (USB Request Reception)

In host controller mode, the read value of the VALID bit is invalid.

DVSQ[2:0] bits (Device State)

The DVSQ[2:0] bits are initialized by a USB bus reset. In host controller mode, the read value is invalid.

BRDY flag (Buffer Ready Interrupt Status)

The BRDY flag indicates the BRDY interrupt status.

The USBFS sets the BRDY bit to 1 when it detects a BRDY interrupt status (PIPEnBRDY = 1, n = 0 to 9) on at least one pipe for which BRDY interrupts are enabled (BRDYENB.PIPEnBRDYE = 1).

For the conditions that cause the PIPEnBRDY status to be asserted, see section 29.3.3.1. BRDY interrupt.

The USBFS sets the BRDY bit to 0 when the software writes 0 to all of the PIPEnBRDY bits associated with the PIPEnBRDYE bits that are set to 1. Writing 0 to the BRDY flag in the software does not clear the flag.

NRDY flag (Buffer Not Ready Interrupt Status)

The NRDY flag indicates the NRDY interrupt status.

The USBFS sets the NRDY bit to 1 when it detects a NRDY interrupt status (PIPEnNRDY = 1, n = 0 to 9) on at least one pipe for which NRDY interrupts are enabled (NRDYENB.PIPEnNRDYE = 1).

For the conditions that cause the PIPEnNRDY status to be asserted, see section 29.3.3.2. NRDY interrupt.

The USBFS sets the NRDY bit to 0 when the software writes 0 to all of the PIPEnNRDY bits associated with the PIPEnNRDYE bits that are set to 1. Writing 0 to the NRDY flag in the software does not clear the flag.

BEMP flag (Buffer Empty Interrupt Status)

The BEMP flag indicates the BEMP interrupt status.

The USBFS sets the BEMP bit to 1 when it detects a BEMP interrupt status (PIPEnBEMP = 1, n = 0 to 9) on at least one pipe for which BEMP interrupts are enabled (BEMPENB.PIPEnBEMPE = 1).

For the conditions that cause the PIPEnBEMP status to be asserted, see section 29.3.3.3. BEMP interrupt.

The USBFS sets the BEMP bit to 0 when the software writes 0 to all of the PIPEnBEMP bits associated with the PIPEnBEMPE bits that are set to 1. Writing 0 to the BEMP flag in the software does not clear the flag.

CTRT flag (Control Transfer Stage Transition Interrupt Status)

In device controller mode, the USBFS updates the value of the CTSQ[2:0] bits and sets the CTRT flag to 1 on detecting a transition in the control transfer stage. When a control transfer stage transition interrupt occurs, clear the CTRT flag before the USBFS detects the next control transfer stage transition.

Values read from the CTRT flag in host controller mode are invalid.

DVST flag (Device State Transition Interrupt Status)

In device controller mode, the USBFS updates the value of the DVSQ[2:0] bits and sets the DVST flag to 1 on detecting a change in the device state. When a device state transition interrupt occurs, clear theDVST flag before the USBFS detects the next device state transition.

Values read from the DVST flag in host controller mode are invalid.

SOFR flag (Frame Number Refresh Interrupt Status)

In host controller mode, the USBFS sets the SOFR flag to 1 on updating the frame number when the DVSTCTR0.UACT bit is set to 1 by software. A SOFR interrupt is detected every 1 ms.

In device controller mode, the USBFS sets the SOFR flag to 1 on updating the frame number. A frame number refresh interrupt is detected every 1 ms.

The USBFS can detect an SOFR interrupt through the internal interpolation function even when a corrupted SOF packet is received from the USB host.

RESM flag (Resume Interrupt Status)

In device controller mode, the USBFS sets the RESM flag to 1 on detecting the falling edge of the signal on the USB_DP pin in the Suspend state (DVSQ[2:0] = 1xxb). Values read from the RESM flag in host controller mode are invalid.

VBINT flag (VBUS Interrupt Status)

The USBFS sets the VBINT flag to 1 on detecting a level change (high to low or low to high) in the USB_VBUS pin input value. The USBFS sets the VBSTS flag to indicate the USB_VBUS pin input value. When a VBUS interrupt occurs, eliminate transient elements by reading the VBSTS flag at least three times through software processing and check that the values read are the same.

29.2.17 INTSTS1 : Interrupt Status Register 1

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x042

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVRCR	BCHG	—	DTCH	ATTCH	—	—	—	—	EOFERR	SIGN	SACK	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	SACK	Setup Transaction Normal Response Interrupt Status 0: No SACK interrupt occurred 1: SACK interrupt occurred	R/W*1
5	SIGN	Setup Transaction Error Interrupt Status 0: No SIGN interrupt occurred 1: SIGN interrupt occurred	R/W*1
6	EOFERR	EOF Error Detection Interrupt Status 0: No EOFERR interrupt occurred 1: EOFERR interrupt occurred	R/W*1

Bit	Symbol	Function	R/W
10:7	—	These bits are read as 0. The write value should be 0.	R/W
11	ATTCH	ATTCH Interrupt Status 0: No ATTCH interrupt occurred 1: ATTCH interrupt occurred	R/W*1
12	DTCH	USB Disconnection Detection Interrupt Status 0: No DTCH interrupt occurred 1: DTCH interrupt occurred	R/W*1
13	—	This bit is read as 0. The write value should be 0.	R/W
14	BCHG	USB Bus Change Interrupt Status*2 0: No BCHG interrupt occurred 1: BCHG interrupt occurred	R/W*1
15	OVRCR	Overcurrent Input Change Interrupt Status*2 0: No OVRCR interrupt occurred 1: OVRCR interrupt occurred	R/W*1

Note: S-TYPE-3, P-TYPE-3
Note 1. To clear the bits in INTSTS1, write 0 only to the bits to be cleared. Write 1 to the other bits.
Note 2. The USBFS detects a change in the status in the OVRCR or BCHG bit even when the clock supply is stopped (SYSCFG.SCKE = 0), and it requests the interrupt when the associated interrupt request bit is 1. Enable the clock supply (SYSCFG.SCKE = 1) before clearing the status through the software. No other interrupts can be detected while the clock supply is stopped (SYSCFG.SCKE bit = 0).

INTSTS1 is used to confirm the status of each interrupt in host controller mode. Only enable the status change interrupts indicated in the bits in INTSTS1 in host controller mode.

SACK flag (Setup Transaction Normal Response Interrupt Status)

The SACK flag indicates the status of the setup transaction normal response interrupt in host controller mode.

The USBFS detects the SACK interrupt and sets this flag to 1 when an ACK response is returned from a peripheral device during the setup transactions issued by the USBFS. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

Values read from the SACK flag in device controller mode are invalid.

SIGN flag (Setup Transaction Error Interrupt Status)

The SIGN flag indicates the status of setup transaction error interrupts in host controller mode.

The USBFS detects the SIGN interrupt and sets this flag to 1 when an ACK response is not returned from a peripheral device three consecutive times during the setup transactions issued by the USBFS. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

The USBFS detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions:

- Timeout is detected by the USBFS when the peripheral device has returned no response
- A corrupted ACK packet is received
- A handshake other than ACK (NAK, NYET, or STALL) is received

Values read from the SIGN flag in device controller mode are invalid.

EOFERR flag (EOF Error Detection Interrupt Status)

The EOFERR flag indicates the status of EOF error detection interrupts in host controller mode.

The USBFS detects the EOFERR interrupt and sets this flag to 1 on detecting that communication did not complete at the EOF2 timing defined in the USB 2.0 specification. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

After detecting the EOFERR interrupt, the USBFS controls the hardware as follows, regardless of the associated interrupt enable bit setting:

- Sets the DVSTCTR0.UACT bit for the port in which the EOFERR interrupt was detected to 0
- Puts the port in which the EOFERR interrupt occurred into the idle state

The software must terminate all pipes in which communications are being carried out and re-enumerate the USB port.

Values read from the EOFERR flag in device controller mode are invalid.

ATTCH flag (ATTCH Interrupt Status)

The ATTCH flag indicates the status of USB attach detection interrupts in host controller mode.

The USBFS detects the ATTCH interrupt and sets this flag to 1 on detecting a J- or K-state on the full- or low-speed signal level for 2.5 µs. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

The USBFS detects the ATTCH interrupt on any of the following conditions.

- K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5 µs
- J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5 µs

Values read from the ATTCH flag in device controller mode are invalid.

DTCH flag (USB Disconnection Detection Interrupt Status)

The DTCH flag indicates the status of USB disconnection detection interrupts in host controller mode.

The USBFS detects the DTCH interrupt and sets this flag to 1 on detecting a USB bus detach event. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

The USBFS detects bus detach events based on the USB 2.0 specification.

After detecting the DTCH interrupt, the USBFS controls hardware as follows, regardless of the associated interrupt enable bit setting:

- Sets the DVSTCTR0.UACT bit for the port in which the DTCH interrupt was detected to 0
- Puts the port in which the DTCH interrupt occurred into the idle state

The software must terminate all pipes in which communications are being carried out and transition to a wait state for connecting to the USB port (waiting for ATTCH interrupt generation).

Values read from the DTCH flag in device controller mode are invalid.

BCHG flag (USB Bus Change Interrupt Status)

The BCHG flag indicates the status of USB bus change interrupts in host controller mode.

The USBFS detects the BCHG interrupt and sets this flag to 1 when a change in the full- or low-speed signal level occurs on the USB port. This includes any change from J-state, K-state, or SE0 to J-state, K-state, or SE0. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

The USBFS sets the LNST[1:0] flags to indicate the input state of the USB port. When a BCHG interrupt occurs, eliminate transient elements by repeat reading the LNST[1:0] flags by software until the same value is read at least three times.

Change in the USB bus state can be detected while the internal clock is stopped.

Values read from the BCHG flag in device controller mode are invalid.

OVRCR flag (Overcurrent Input Change Interrupt Status)

The OVRCR flag indicates the status of USB_OVRCURA, USB_OVRCURA-DS, USB_OVRCURB or USB_OVRCURB-DS input pin change interrupts.

The USBFS detects the OVRCR interrupt and sets this flag to 1 when a change (high to low or low to high) occurs in at least one of the input values to the USB_OVRCURA, USB_OVRCURA-DS, USB_OVRCURB or USB_OVRCURB-DS pins. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

29.2.18 BRDYSTS : BRDY Interrupt Status Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x046

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPE9 BRDY	PIPE8 BRDY	PIPE7 BRDY	PIPE6 BRDY	PIPE5 BRDY	PIPE4 BRDY	PIPE3 BRDY	PIPE2 BRDY	PIPE1 BRDY	PIPE0 BRDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIPE0BRDY	BRDY Interrupt Status for Pipe 0 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
1	PIPE1BRDY	BRDY Interrupt Status for Pipe 1 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
2	PIPE2BRDY	BRDY Interrupt Status for Pipe 2*2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
3	PIPE3BRDY	BRDY Interrupt Status for Pipe 3*2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
4	PIPE4BRDY	BRDY Interrupt Status for Pipe 4 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
5	PIPE5BRDY	BRDY Interrupt Status for Pipe 5 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
6	PIPE6BRDY	BRDY Interrupt Status for Pipe 6*2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
7	PIPE7BRDY	BRDY Interrupt Status for Pipe 7 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
8	PIPE8BRDY	BRDY Interrupt Status for Pipe 8 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
9	PIPE9BRDY	BRDY Interrupt Status for Pipe 9 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. When the SOFCFG.BRDYM bit is set to 0, to clear the status indicated in the bits in BRDYSTS, write 0 only to the bits to be cleared. Write 1 to the other bits.
Note 2. When the SOFCFG.BRDYM bit is set to 0, clear BRDY interrupts before accessing the FIFO.

29.2.19 NRDYSTS : NRDY Interrupt Status Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x048

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPE9 NRDY	PIPE8 NRDY	PIPE7 NRDY	PIPE6 NRDY	PIPE5 NRDY	PIPE4 NRDY	PIPE3 NRDY	PIPE2 NRDY	PIPE1 NRDY	PIPE0 NRDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIPE0NRDY	NRDY Interrupt Status for Pipe 0 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
1	PIPE1NRDY	NRDY Interrupt Status for Pipe 1 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
2	PIPE2NRDY	NRDY Interrupt Status for Pipe 2 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
3	PIPE3NRDY	NRDY Interrupt Status for Pipe 3 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
4	PIPE4NRDY	NRDY Interrupt Status for Pipe 4 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
5	PIPE5NRDY	NRDY Interrupt Status for Pipe 5 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
6	PIPE6NRDY	NRDY Interrupt Status for Pipe 6 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
7	PIPE7NRDY	NRDY Interrupt Status for Pipe 7 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
8	PIPE8NRDY	NRDY Interrupt Status for Pipe 8 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
9	PIPE9NRDY	NRDY Interrupt Status for Pipe 9 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. To clear the status indicated in the bits in NRDYSTS, write 0 only to the bits to be cleared. Write 1 to the other bits.

29.2.20 BEMPSTS : BEMP Interrupt Status Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x04A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPE9 BEMP	PIPE8 BEMP	PIPE7 BEMP	PIPE6 BEMP	PIPE5 BEMP	PIPE4 BEMP	PIPE3 BEMP	PIPE2 BEMP	PIPE1 BEMP	PIPE0 BEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIPE0BEMP	BEMP Interrupt Status for Pipe 0 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
1	PIPE1BEMP	BEMP Interrupt Status for Pipe 1 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
2	PIPE2BEMP	BEMP Interrupt Status for Pipe 2 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
3	PIPE3BEMP	BEMP Interrupt Status for Pipe 3 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
4	PIPE4BEMP	BEMP Interrupt Status for Pipe 4 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
5	PIPE5BEMP	BEMP Interrupt Status for Pipe 5 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
6	PIPE6BEMP	BEMP Interrupt Status for Pipe 6 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
7	PIPE7BEMP	BEMP Interrupt Status for Pipe 7 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
8	PIPE8BEMP	BEMP Interrupt Status for Pipe 8 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
9	PIPE9BEMP	BEMP Interrupt Status for Pipe 9 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. To clear the status indicated in the bits in BEMPSTS, write 0 only to the bits to be cleared. Write 1 to the other bits.

29.2.21 FRMNUM : Frame Number Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x04C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVRN	CRCE	—	—	—	FRNM[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	FRNM[10:0]	Frame Number Latest frame number.	R
13:11	—	These bits are read as 0. The write value should be 0.	R/W
14	CRCE	Receive Data Error 0: No error occurred 1: Error occurred	R/W*1
15	OVRN	Overrun/Underrun Detection Status 0: No error occurred 1: Error occurred	R/W*1

Note: S-TYPE-3, P-TYPE-3
Note 1. To clear the status, write 0 only to the bits to be cleared. Write 1 to the other bits.

FRNM[10:0] flags (Frame Number)

The USBFS sets the FRNM[10:0] flags to indicate the latest frame number, which is updated every 1 ms, when an SOF packet is issued or received.

CRCE flag (Receive Data Error)

The CRCE flag sets to 1 when a CRC error or bit stuffing error occurs during isochronous transfer. On detecting a CRC error in host controller mode, the USBFS generates an internal NRDY interrupt.

To clear the CRCE flag, write 0 to it while writing 1 to the other bits in the FRMNUM register.

OVRN flag (Overrun/Underrun Detection Status)

The OVRN flag sets to 1 when an overrun or underrun error occurs during isochronous transfer. To clear the flag, write 0 to it while writing 1 to the other bits in the FRMNUM register.

In host controller mode, the OVRN flag sets to 1 on any of the following conditions:

- For a transmitting isochronous pipe, the time to issue an OUT token comes before all of the transmit data is written to the FIFO buffer
- For a receiving isochronous pipe, the time to issue an IN token comes when no FIFO buffer planes are empty

In device controller mode, the OVRN flag sets to 1 on any of the following conditions:

- For a transmitting isochronous pipe, the IN token is received before all of the transmit data is written to the FIFO buffer
- For a receiving isochronous pipe, the OUT token is received when no FIFO buffer planes are empty

29.2.22 DVCHGR : Device State Change Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x04E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DVCHG	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
14:0	—	These bits are read as 0. The write value should be 0.	R/W
15	DVCHG	Device State Change 0: Disable writes to the USBADDR.STSRECOV[3:0] and USBADDR.USBADDR[6:0] bits 1: Enable writes to the USBADDR.STSRECOV[3:0] and USBADDR.USBADDR[6:0] bits	R/W

Note: S-TYPE-3, P-TYPE-3

For details, see section 29.3.1.5. Release from Deep Software Standby mode1 because of USB suspend/resume interrupts.

29.2.23 USBADDR : USB Address Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x050

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	STSRECOV[3:0]				—	USBADDR[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	USBADDR[6:0]	USB Address In device controller mode, these bits indicate the USB address assigned by the host when the USBFS processed the SET_ADDRESS request successfully.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
11:8	STSRECOV[3:0]	Status Recovery 0x4: Recovery in device controller mode: Setting prohibited Recovery in host controller mode: Return to the low-speed state (bits DVSTCTR0.RHST[2:0] = 001b) 0x8: Recovery in device controller mode: Setting prohibited Recovery in host controller mode: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b) 0x9: Recovery in device controller mode: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 001b (default state) Recovery in host controller mode: Setting prohibited 0xA: Recovery in device controller mode: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 010b (address state) Recovery in host controller mode: Setting prohibited 0xB: Recovery in device controller mode: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 011b (configured state) Recovery in host controller mode: Setting prohibited Others: Setting prohibited	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

USBADDR[6:0] bits (USB Address)

In device controller mode, the USBADDR[6:0] flags indicate the USB address received when the USBFS processed a SetAddress request successfully. The USBFS sets the USBADDR[6:0] bits to 0x00 on detecting a USB bus reset.

Writing to these bits is enabled while the DVCHGR.DVCHG bit is set to 1. On recovering from a USB power shut-off, the operation can resume from the USB address set before the software shut-off.

In host controller mode, the USBADDR[6:0] bits are invalid.

STSRECOV[3:0] bits (Status Recovery)

Use the STSRECOV[3:0] bits to resume the state of the internal sequencer on recovering from USB power shut-off. For details, see section 29.3.1.5. Release from Deep Software Standby mode1 because of USB suspend/resume interrupts.

Writing to these bits is enabled while the DVCHGR.DVCHG bit is set to 1.

29.2.24 USBREQ : USB Request Type Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x054

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BREQUEST[7:0]								BMREQUESTTYPE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	BMREQUESTTYPE[7:0]	Request Type USB request bmRequestType value	R/W*1
15:8	BREQUEST[7:0]	Request USB request bRequest value	R/W*1

Note: S-TYPE-3, P-TYPE-3

Note 1. In device controller mode, these bits can be read, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

USBREQ stores setup requests for control transfers.

In device controller mode, the USBREQ stores the received bRequest and bmRequestType values. In host controller mode, it sets to the bRequest and bmRequestType values to be transmitted.

USBREQ is initialized by a USB bus reset.

BMREQUESTTYPE[7:0] bits (Request Type)

The BMREQUESTTYPE[7:0] bits hold the bmRequestType value of USB requests.

- In host controller mode:

  Set these bits to the value of the USB request data in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.

- In device controller mode:

  These bits indicate the value of the USB request data in reception setup transactions. Writing to the bits has no effect.

BREQUEST[7:0] bits (Request)

The BREQUEST[7:0] bits store bRequest value of the USB request.

- In host controller mode:

  Set these bits to the value of the USB request data in setup transmission transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.

- In device controller mode:

  These bits indicate the value of the USB request data in reception setup transactions. Writing to the bits has no effect.

29.2.25 USBVAL : USB Request Value Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x056

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WVALUE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WVALUE[15:0]	Value USB request wValue value	R/W*1

Note: S-TYPE-3, P-TYPE-3

Note 1. In device controller mode, these bits can be read, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

In device controller mode, USBVAL stores the received wValue value. In host controller mode, it sets to the wValue value to be transmitted is set.

USBVAL is initialized by a USB bus reset.

WVALUE[15:0] bits (Value)

The WVALUE[15:0] bits store wValue value of the USB request.

- In host controller mode:

  Set these bits to the value of the wValue field in USB requests of transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.

- In device controller mode:

These bits indicate the wValue value of USB requests in reception setup transactions. Writing to the bits has no effect.

29.2.26 USBINDX : USB Request Index Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x058

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WINDEX[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WINDEX[15:0]	Index USB request wIndex value	R/W*1

Note: S-TYPE-3, P-TYPE-3
Note 1. In device controller mode, these bits can be read, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

USBINDX stores setup requests for control transfers.

In device controller mode, it stores the received wIndex value. In host controller mode, it sets to the wIndex value to be transmitted.

USBINDX is initialized by a USB bus reset.

WINDEX[15:0] bits (Index)

The WINDEX[15:0] bits hold the wIndex value of a USB request.

- In host controller mode:

  Set these bits to the wIndex value in USB requests in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.

- In device controller mode:

  These bits indicate the wIndex value in USB requests received in reception setup transactions. Writing to the bits has no effect.

29.2.27 USBLENG : USB Request Length Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x05A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WLENTUH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WLENTUH[15:0]	Length USB request wLength value	R/W*1

Note: S-TYPE-3, P-TYPE-3
Note 1. In device controller mode, these bits can be read, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

USBLENG stores setup requests for control transfers.

In device controller mode, the value of wLength that is received is stored. In host controller mode, the value of wLength to be transmitted is set.

USBLENG is initialized by a USB bus reset.

WLENTUH[15:0] bits (Length)

The WLENTUH[15:0]bits hold the wLength value of a USB request.

- In host controller mode:

  Set these bits to the wLength value in USB requests in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.

- In device controller mode:

  These bits indicate the wLength value in USB requests received in reception setup transactions. Writing to the bits has no effect.

29.2.28 DCPCFG : DCP Configuration Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x05C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SHTNAK	—	—	DIR	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	DIR	Transfer Direction*1 0: Data receiving direction 1: Data transmitting direction	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	SHTNAK	Pipe Disabled at End of Transfer*1 0: Keep pipe open after transfer ends 1: Disable pipe after transfer ends	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Only set this bit while the PID is NAK. Before setting this bit, check that the DCPCTR.PBUSY bit is 0, and then change the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

DIR bit (Transfer Direction)

In host controller mode, the DIR bit sets the transfer direction of the data stage and status stage for control transfers. In device controller mode, set the DIR bit to 0.

SHTNAK bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to change PID to NAK on transfer end when the selected pipe is receiving. It is only valid when the selected pipe is receiving.

When the SHTNAK bit is 1, the USBFS changes the DCPCTR.PID[1:0] bits for the DCP to NAK on determining that a transfer has ended. The USBFS determines transfer end on the following condition:

- A short packet, including a zero-length packet, is successfully received.

29.2.29 DCPMAXP : DCP Maximum Packet Size Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x05E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DEVSEL[3:0]				—	—	—	—	—	MXPS[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	MXPS[6:0]	Maximum Packet Size*1 Maximum data payload specification (maximum packet size) for the DCP	R/W
11:7	—	These bits are read as 0. The write value should be 0.	R/W
15:12	DEVSEL[3:0]	Device Select*2 0x0: Address 0000b 0x1: Address 0001b 0x2: Address 0010b 0x3: Address 0011b 0x4: Address 0100b 0x5: Address 0101b Others: Setting prohibited	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Only set the MXPS[6:0] bits while PID is NAK. Before setting these bits, check that the DCPCTR.PBUSY bit is 0, and then change the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary. After the MXPS[6:0] bits are set and the DCP is set to the CURPIPE[3:0] bits in a port select register, clear the buffer by setting the BCLR bit in the port control register to 1.
Note 2. Only set the DEVSEL[3:0] bits while PID is NAK and the DCPCTR.SUREQ bit is 0. Before setting these bits, check that the DCPCTR.PBUSY bit is 0, and then change the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

MXPS[6:0] bits (Maximum Packet Size)

The MXPS[6:0] bits specify the maximum data payload (maximum packet size) for the DCP. The initial value is 0x40 (64 bytes). Set the bits to a USB 2.0-compliant value. Do not write to the FIFO buffer or set PID = BUF whileMXPS[6:0] is set to 0.

DEVSEL[3:0] bits (Device Select)

In host controller mode, the DEVSEL[3:0] bits specify the address of the target peripheral device for a control transfer. Set up the associated DEVADDn (n = 0 to 5) register first, and then set these bits to the corresponding value. To set the DEVSEL[3:0] bits to 0010b, for example, first set the DEVADD2 register.

In device controller mode, set these bits to 0000b.

29.2.30 DCPCTR : DCP Control Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x060

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSTS	SUREQ	—	—	SUREQCLR	—	—	SQCLR	SQSET	SQMON	PBUSY	—	—	CCPL	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PID[1:0]	Response PID 0 0: NAK response 0 1: BUF response (depends on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
2	CCPL	Control Transfer End Enable 0: Disable control transfer completion 1: Enable control transfer completion	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	PBUSY	Pipe Busy 0: DCP not used for the USB bus 1: DCP in use for the USB bus	R
6	SQMON	Sequence Toggle Bit Monitor 0: DATA0 1: DATA1	R
7	SQSET	Sequence Toggle Bit Set*2 Sets the sequence toggle bit in DCP transfers. 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA1	R/W*1
8	SQCLR	Sequence Toggle Bit Clear*2 Clears the sequence toggle bit in DCP transfers. 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0	R/W*1
10:9	—	These bits are read as 0. The write value should be 0.	R/W
11	SUREQCLR	SUREQ Bit Clear Clears the SUREQ bit in host controller mode. 0: Invalid (writing 0 has no effect) 1: Clear SUREQ to 0	R/W*1
13:12	—	These bits are read as 0. The write value should be 0.	R/W
14	SUREQ	Setup Token Transmission Sets up token transmission in host controller mode. 0: Invalid (writing 0 has no effect) 1: Transmit setup packet	R/W
15	BSTS	Buffer Status 0: Buffer access disabled 1: Buffer access enabled	R

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is read as 0.
Note 2. Only set the SQSET and SQCLR bits while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

PID[1:0] bits (Response PID)

The PID[1:0] bits control the USB response type during control transfers.

In host controller mode, to change the PID[1:0] setting from NAK to BUF:

- When the transmitting direction is set:
  a. Write all of the transmit data to the FIFO buffer while the DVSTCTR0.UACT bit is 1 and PID is NAK.
  b. Set PID[1:0] bits to 01b (BUF).
     The USBFS then executes the OUT transaction.
- When the receiving direction is set:
  a. Check that the FIFO buffer is empty (or empty the buffer) while the DVSTCTR0.UACT bit is 1 and PID is NAK.
  b. Set PID[1:0] bits to 01b (BUF).
     The USBFS then executes the IN transaction.

The USBFS changes the PID[1:0] setting as follows:

- When the PID[1:0] bits are set to BUF (01b) by software and the USBFS has received data exceeding MaxPacketSize, the USBFS sets the PID[1:0] to STALL (11b)
- When a reception error, such as a CRC error, is detected three times consecutively, the USBFS sets thePID[1:0] bits to NAK (00b)
- On receiving the STALL handshake, the USBFS sets PID[1:0] to STALL (11b)

In device controller mode, the USBFS changes the PID[1:0] setting as follows:

- On receiving a setup packet, the USBFS sets PID[1:0] to NAK (00b). The USBFS then sets the INTSTS0.VALID flag to 1, and the PID[1:0] setting cannot be changed until the software clears the VALID flag to 0.
- When the PID[1:0] bits are set to BUF (01b) by software and the USBFS has received data exceeding MaxPacketSize, the USBFS sets PID[1:0] to STALL (11b)
- On detecting a control transfer sequence error, the USBFS sets PID[1:0] to STALL (1xb)
- On detecting a USB bus reset, the USBFS sets PID[1:0] to NAK

The USBFS does not check the PID[1:0] setting while processing a SET_ADDRESS request.

The PID[1:0] bits are initialized by a USB bus reset.

CCPL bit (Control Transfer End Enable)

In device controller mode, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed. When the bit is set to 1 by software while the associated PID[1:0] bits are set to BUF, the USBFS completes the control transfer status stage.

During control read transfers, the USBFS transmits the ACK handshake in response to the OUT transaction from the USB host. During control write or no-data control transfers, it transmits the zero-length packet in response to the IN transaction from the USB host. On detecting a SET_ADDRESS request, the USBFS operates in auto response mode from the setup stage up to status stage completion regardless of the CCPL bit setting.

The USBFS changes the CCPL bit from 1 to 0 on receiving a new setup packet. The software cannot write 1 to the bit while the INTSTS0.VALID bit is 1. The bit is initialized by a USB bus reset.

In host controller mode, always write 0 to the CCPL bit.

PBUSY bit (Pipe Busy)

The PBUSY bit indicates whether DCP is used for the transaction when USBFS changes the PID[1:0] bits from BUF to NAK. The USBFS changes the PBUSY bit from 0 to 1 on start of a USB transaction for the selected pipe. It changes the PBUSY bit from 1 to 0 on completion of one transaction.

After PID is set to NAK by software, the value in the PBUSY bit indicates whether changes to pipe settings can proceed.

For details, see section 29.3.4.1. Pipe control register switching procedures.

SQMON bit (Sequence Toggle Bit Monitor)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction during a DCP transfer.

The USBFS toggles the bit on normal completion of the transaction. It does not toggle the bit, however, when a DATAPID mismatch occurs during a transfer in the receiving direction.

In device controller mode, the USBFS sets the SQMON bit to 1 (specifies DATA1 as the expected value) on successful reception of the setup packet.

In device controller mode, the USBFS does not reference this bit during IN or OUT transactions at the status stage, and it does not toggle the bit on normal completion.

SQSET bit (Sequence Toggle Bit Set)

The SQSET bit specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during a DCP transfer.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

SQCLR bit (Sequence Toggle Bit Clear)

The SQCLR bit specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during a DCP transfer. It is read as 0.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

SUREQCLR bit (SUREQ Bit Clear)

In host controller mode, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The bit is read as 0.

If transfer stops while the SUREQ bit is set to 1 in a setup transaction, set the SUREQCLR bit to 1 by software. This is not necessary at the end of a normal setup transaction, because the USBFS automatically clears the SUREQ bit to 0.

Only control the SUREQ bit through the SUREQCLR bit while the DVSTCTR0.UACT bit is 0. When UACT is 0, communication is halted or no transfer is occurring because a bus disconnection was detected.

In device controller mode, always write 0 to this bit.

SUREQ bit (Setup Token Transmission)

In host controller mode, setting the SUREQ bit to 1 triggers the USBFS to transmit the setup packet. After completing the setup transaction process, the USBFS generates either the SACK or SIGN interrupt and clears the SUREQ bit to 0. The USBFS also clears the SUREQ bit to 0 when the software sets the SUREQCLR bit to 1.

Before setting the SUREQ bit to 1, set the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, and USBLENG appropriately to transmit the target USB request in the setup transaction. Also check that the PID[1:0] bits for the DCP are set to NAK. After setting the SUREQ bit to 1, do not change the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, or USBLENG until the setup transaction is complete (SUREQ bit = 1). Write 1 to the SUREQ bit only when transmitting the setup token. Otherwise, write 0.

In device controller mode, always write 0 to this bit.

BSTS flag (Buffer Status)

The BSTS flag indicates the status of access to the DCP FIFO buffer. The meaning of this flag varies as follows depending on the CFIFOSEL.ISEL setting:

- When ISEL = 0, the bit indicates whether receive data can be read from the buffer
- When ISEL = 1, the bit indicates whether transmit data can be written to the buffer

29.2.31 PIPESEL : Pipe Window Select Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x064

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PIPESEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	PIPESEL[3:0]	Pipe Window Select 0x0: No pipe selected 0x1: Pipe 1 0x2: Pipe 2 0x3: Pipe 3 0x4: Pipe 4 0x5: Pipe 5 0x6: Pipe 6 0x7: Pipe 7 0x8: Pipe 8 0x9: Pipe 9 Others: Setting prohibited	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Set pipes and functions using the PIPESEL, PIPECFG, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEmTRE, and PIPEmTRN registers (n = 1 to 9, m = 1 to 5).

The pipe is selected by PIPESEL.

After selecting the pipe in the PIPESEL register, pipe functions must be set in the associated PIPECFG, PIPEMAXP, and PIPEPERI registers. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set independently of the pipe selection in this register.

PIPESEL[3:0] bits (Pipe Window Select)

The PIPESEL[3:0] bits select the pipe number associated with the PIPECFG, PIPEMAXP, and PIPEPERI registers used for data writing and reading. Selecting a pipe number in the PIPESEL[3:0] bits allows writing to and reading from PIPECFG, PIPEMAXP, and PIPEPERI associated with the selected pipe number.

When PIPESEL[3:0] = 0000b, 0 is read from all of the bits in PIPECFG, PIPEMAXP, and PIPEPERI. Writing to these bits has no effect.

29.2.32 PIPECFG : Pipe Configuration Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x068

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TYPE[1:0]		—	—	—	BFRE	DBLB	—	SHTNAK	—	—	DIR	EPNUM[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	EPNUM[3:0]	Endpoint Number*1 Specifies the endpoint number for the selected pipe. Setting 0000b indicates that the pipe is not used.	R/W
4	DIR	Transfer Direction*2 *3 0: Receiving direction 1: Transmitting direction	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	SHTNAK	Pipe Disabled at End of Transfer*1 0: Continue pipe operation after transfer ends 1: Disable pipe after transfer ends	R/W
8	—	This bit is read as 0. The write value should be 0.	R/W
9	DBLB	Double Buffer Mode*2 *3 0: Single buffer 1: Double buffer	R/W
10	BFRE	BRDY Interrupt Operation Specification*2 *3 0: Generate BRDY interrupt on transmitting or receiving data 1: Generate BRDY interrupt on completion of reading data	R/W
13:11	—	These bits are read as 0. The write value should be 0.	R/W
15:14	TYPE[1:0]	Transfer Type*1 0 0: Pipe not used 0 1: Pipes 1 and 2: Bulk transfer Pipes 3 to 5: Bulk transfer Pipes 6 to 9: Setting prohibited 1 0: Pipes 1 and 2: Setting prohibited Pipes 3 to 5: Setting prohibited Pipes 6 to 9: Interrupt transfer 1 1: Pipes 1 and 2: Isochronous transfer Pipes 3 to 5: Setting prohibited Pipes 6 to 9: Setting prohibited	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Only set the TYPE[1:0], SHTNAK, and EPNUM[3:0] bits while PID is NAK. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.
Note 2. Only set the BFRE, DBLB, and DIR bits while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.
Note 3. To change the BFRE, DBLB, or DIR bits after completing USB communication on the selected pipe, in addition to the constraints described in Note 2, write 1 and 0 to the PIPEnCTR.ACLRM bit continuously through the software and clear the FIFO buffer assigned to the pipe.

PIPECFG specifies the transfer type, FIFO buffer access direction, and endpoint numbers for pipes 1 to 9. It also selects single or double buffer mode, and whether to continue or disable pipe operation at the end of transfer.

EPNUM[3:0] bits (Endpoint Number)

The EPNUM[3:0] bits specify the endpoint number for the selected pipe. Setting 0000b indicates the pipe not used.

Set these bits so that the combination of the DIR and EPNUM[3:0] settings is different from those for other pipes. The EPNUM[3:0] bits can be set to 0000b for all pipes.

DIR bit (Transfer Direction)

The DIR bit specifies the transfer direction for the selected pipe.

When the software sets this bit to 0, the USBFS uses the selected pipe for receiving. When the software sets this bit to 1, the USBFS uses the selected pipe for transmitting.

SHTNAK bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to change the PIPEnCTR.PID[1:0] bits to 00b (NAK) at the end of transfer when the selected pipe is set in the receiving direction. The bit is valid for pipes 1 to 5 in the receiving direction.

When the software sets this bit to 1 for a receiving pipe, the USBFS changes the associated PIPEnCTR.PID[1:0] bits to 00b (NAK) on determining the transfer end. The USBFS determines that the transfer has ended on the following conditions:

- A short packet data (including a zero-length packet) was successfully received
- The transaction counter is used and the number of packets specified for the transaction counter are successfully received

DBLB bit (Double Buffer Mode)

The DBLB bit selects either single or double buffer mode for the FIFO buffer used by the selected pipe. The bit is valid for pipes 1 to 5.

BFRE bit (BRDY Interrupt Operation Specification)

The BFRE bit specifies the BRDY interrupt generation timing from the USBFS to the CPU for the selected pipe.

When the software sets the BFRE bit to 1 and the selected pipe is in the receiving direction, the USBFS detects the transfer completion and generates the BRDY interrupt on reading the packet.

When a BRDY interrupt is generated with this setting, the software must write 1 to the BCLR bit in the port control register. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit.

When the BFRE bit is set to 1 by software and the selected pipe is in the transmitting direction, the USBFS does not generate the BRDY interrupt. For details, see section 29.3.3.1. BRDY interrupt.

TYPE[1:0] bits (Transfer Type)

The TYPE[1:0] bits specify the transfer type for the pipe selected in the PIPESEL.PIPESEL[3:0] bits. Before setting PID to BUF and starting USB communication on the selected pipe, set the TYPE[1:0] bits to a value other than 00b.

29.2.33 PIPEMAXP : Pipe Maximum Packet Size Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x06C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DEVSEL[3:0]				—	—	—	MXPS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	MXPS[8:0]	Maximum Packet Size*1 ● Pipes 1 and 2 1 byte (0x001) to 256 bytes (0x100) (Bit [9] not supported.) ● Pipes 3 to 5 8 bytes (0x008), 16 bytes (0x010), 32 bytes (0x020), 64 bytes (0x040) (Bits [9:7] and [2:0] not supported.) ● Pipes 6 to 9 1 byte (0x001) to 64 bytes (0x040) (Bits [9:7] not supported.)	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
15:12	DEVSEL[3:0]	Device Select*2 0x0: Address 0000b 0x1: Address 0001b 0x2: Address 0010b 0x3: Address 0011b 0x4: Address 0100b 0x5: Address 0101b Others: Setting prohibited	R/W

Note: S-TYPE-3, P-TYPE-3

Note: The value of the MXPS[8:0] bits is 0x000 when no pipe is selected in the PIPESEL.PIPESEL[3:0] bits and 0x040 when a pipe is selected.

Note 1. Only set the MXPS[8:0] bits while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

Note 2. Only set the DEVSEL[3:0] bits while PID is NAK. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00b (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

PIPEMAXP specifies the maximum packet size for pipes 1 to 9.

MXPS[8:0] bits (Maximum Packet Size)

The MXPS[8:0] bits specify the maximum data payload (maximum packet size) for the selected pipe.

Set these bits to the appropriate value for each transfer type based on the USB 2.0 specification. When MXPS[8:0] = 0, do not write to the FIFO buffer or set PID to BUF. These writes have no effect.

DEVSEL[3:0] bits (Device Select)

In host controller mode, the DEVSEL[3:0] bits specify the address of the target device for USB communication. Set up the associated DEVADDn (n = 0 to 5) register first, and then set these bits to the corresponding value. To set the DEVSEL[3:0] bits to 0x2, for example, first set the DEVADD2 register.

In device controller mode, set these bits to 0x0.

29.2.34 PIPEPERI : Pipe Cycle Control Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x06E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	IFIS	—	—	—	—	—	—	—	—	—	IITV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	IITV[2:0]*1	Interval Error Detection Interval Specifies the interval error detection timing for the selected pipe as the n-th power of 2 of the frame timing	R/W
11:3	—	These bits are read as 0. The write value should be 0.	R/W
12	IFIS	Isochronous IN Buffer Flush 0: Do not flush buffer 1: Flush buffer	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Only set the IITV[2:0] bits while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

PIPEPERI selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfers, and sets the interval error detection interval for pipes 1 to 9.

IITV[2:0] bits (Interval Error Detection Interval)

To change the IITV[2:0] bits to another value after they are set and USB communication is performed, set the PIPEnCTR.PID[1:0] bits to 00b (NAK) and then set the PIPEnCTR.ACLRM bit to 1 to initialize the interval timer.

The IITV[2:0] bits are not provided for pipes 3 to 5. Write 000b to bit positions of the IITV[2:0] bits associated with pipes 3 to 5.

IFIS bit (Isochronous IN Buffer Flush)

The IFIS bit specifies whether to flush the buffer when the pipe selected in the PIPESEL.PIPESEL[3:0] bits is used for isochronous IN transfers.

In device controller mode when the selected pipe is for isochronous IN transfers, the USBFS automatically clears the FIFO buffer if the USBFS fails to receive the IN token from the USB host within the interval set in theIITV[2:0] bits in terms of frames.

When double buffering is specified (PIPECFG.DBLB = 1), the USBFS only clears the data in the previously used plane.

The USBFS clears the FIFO buffer on receiving the SOF packet immediately after the frame in which the USBFS expected to receive the IN token. Even if the SOF packet is corrupted, the FIFO buffer is cleared at the time the SOF packet is expected to be received by using the internal interpolation function.

When the host controller function is selected, set this bit to 0.

Set this bit to 0 when the selected pipe is not for isochronous transfer.

29.2.35 PIPEnCTR : PIPEn Control Registers (n = 1 to 5)

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x070 + 0x2 × (n - 1)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PID[1:0]	Response PID 0 0: NAK response 0 1: BUF response (depends buffer state) 1 0: STALL response 1 1: STALL response	R/W
4:2	—	These bits are read as 0. The write value should be 0.	R/W
5	PBUSY	Pipe Busy 0: Pipe n not in use for the transaction 1: Pipe n in use for the transaction	R
6	SQMON	Sequence Toggle Bit Confirmation 0: DATA0 1: DATA1	R
7	SQSET	Sequence Toggle Bit Set*2 Sets the sequence toggle bit for pipe n. 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA1	R/W*1
8	SQCLR	Sequence Toggle Bit Clear*2 Clears the sequence toggle bit for pipe n. 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0	R/W*1
9	ACLRM	Auto Buffer Clear Mode*3 0: Disable 1: Enable (initialize all buffers)	R/W
10	ATREPM	Auto Response Mode*2 0: Disable auto response mode 1: Enable auto response mode	R/W
13:11	—	These bits are read as 0. The write value should be 0.	R/W
14	INBUFM	Transmit Buffer Monitor 0: No data to be transmitted is in the FIFO buffer 1: Data to be transmitted is in the FIFO buffer	R
15	BSTS	Buffer Status 0: Buffer access by the CPU disabled 1: Buffer access by the CPU enabled	R

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is read as 0.
Note 2. Only set the ATREPM bit or write 1 to the SQCLR or SQSETbit while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSYbit through the software is not necessary.
Note 3. Only set the ACLRM bit while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting this bit, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

PIPEnCTR can be set for any pipe selection in the PIPESEL register.

PID[1:0] bits (Response PID)

The PID[1:0]bits specify the response type for the next transaction on the selected pipe.

The default PID[1:0] setting is NAK. Change the PID[1:0] setting to BUF to use the associated pipe for USB transfer. Table 29.9 and Table 29.10 show the basic operations of the USBFS (when there are no errors in the communication packets) based on the PID[1:0] bit setting.

After changing the PID[1:0] setting from BUF to NAK through the software during USB communication on the selected pipe, check that the PBUSY bit is 1 to see if USB transfer on the pipe has actually entered the NAK state. If the USBFS changes the PID[1:0] bits to NAK, checking the PBUSY bit through the software is not necessary.

The USBFS changes the PIPEnCTR.PID[1:0] setting in the following cases:

- The USBFS sets PID to NAK on recognizing completion of the transfer when the selected pipe is in the receiving direction and the PIPECFG.SHTNAK bit for the selected pipe is set to 1 by software
- The USBFS sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the selected pipe
- The USBFS sets PID to NAK on detecting a USB bus reset in device controller mode
- The USBFS sets PID to NAK on detecting a reception error, such as a CRC error, three consecutive times in host controller mode
- The USBFS sets PID to STALL (11b) on receiving the STALL handshake in host controller mode

To specify the response type, set the PID[1:0] bits as follows:

- To transition from NAK (00b) to STALL, set 10b
- To transition from BUF (01b) to STALL, set 11b
- To transition from STALL (11b) to NAK, set 10b and then 00b
- To transition from STALL to BUF, set 00b (NAK) and then 01b (BUF)

Table 29.9 Operation of the USBFS based on the PID[1:0] setting in host controller mode

PID[1:0] value	Transfer type	Transfer direction (DIR bit)	USBFS operation
00b (NAK)	Does not depend on the setting	Does not depend on the setting	Does not issue tokens
01b (BUF)	Bulk or interrupt	Does not depend on the setting	Issues tokens when the DVSTCTR0.UACT bit is 1 and the FIFO buffer associated with the selected pipe is ready for transmission and reception. Does not issue tokens when the DVSTCTR0.UACT bit is 0 or the FIFO buffer associated with the selected pipe is not ready for transmission or reception.
	Isochronous	Does not depend on the setting	Issues tokens regardless of the status of the FIFO buffer associated with the selected pipe.
10b (STALL) or 11b (STALL)	Does not depend on the setting	Does not depend on the setting	Does not issue tokens.

Table 29.10 Operation of the USBFS based on the PID[1:0] setting in device controller mode (1 of 2)

PID[1:0] value	Transfer type	Transfer direction (DIR bit)	USBFS operation
00b (NAK)	Bulk or interrupt	Does not depend on the setting	Returns NAK in response to the token from the USB host
	Isochronous	Does not depend on the setting	Returns nothing in response to the token from the USB host

Table 29.10 Operation of the USBFS based on the PID[1:0] setting in device controller mode (2 of 2)

PID[1:0] value	Transfer type	Transfer direction (DIR bit)	USBFS operation
01b (BUF)	Bulk	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception
	Interrupt	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception
	Bulk or interrupt	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the FIFO buffer associated with the selected pipe is ready for transmission. Otherwise, returns NAK.
	Isochronous	Receiving direction (DIR = 0)	Receives data in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception. Otherwise, discards the data.
	Isochronous	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the associated FIFO buffer is ready for transmission. Otherwise, transmits a zero-length packet.
10b (STALL) or 11b (STALL)	Bulk or interrupt	Does not depend on the setting	Returns STALL in response to the token from the USB host
	Isochronous	Does not depend on the setting	Returns nothing in response to the token from the USB host

PBUSY bit (Pipe Busy)

The PBUSY bit indicates whether the selected pipe is being used for the current transaction.

The USBFS changes the PBUSY bit from 0 to 1 on start of the USB transaction for the selected pipe, and changes the PBUSY bit from 1 to 0 on completion of one transaction.

Reading the PBUSY bit by software after PID is set to NAK allows you to check whether changing the pipe setting is possible. For details, see section 29.3.4.1. Pipe control register switching procedures.

SQMON bit (Sequence Toggle Bit Confirmation)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction of the selected pipe.

When the selected pipe is not the isochronous transfer type, the USBFS toggles the SQMON flag on successful completion of the transaction. However, the USBFS does not toggle the SQMON flag when a DATA-PID mismatch occurs during transfer in the receiving direction.

SQSET bit (Sequence Toggle Bit Set)

Setting the SQSET bit to 1 through the software causes the USBFS to set DATA1 as the expected value of the sequence toggle bit for the next transaction on the selected pipe. The USBFS clears the SQSET bit to 0.

SQCLR bit (Sequence Toggle Bit Clear)

Setting the SQCLR bit to 1 through the software causes the USBFS to clear the expected value of the sequence toggle bit for the next transaction on the selected pipe to DATA0. The USBFS clears the SQCLR bit to 0.

ACLRM bit (Auto Buffer Clear Mode)

The ACLRM bit enables or disables auto buffer clear mode for the selected pipe. To completely clear the data in the FIFO buffer allocated to the selected pipe, write 1 and then 0 to the ACLRM bit continuously.

Table 29.11 shows the data cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which this processing is required.

Table 29.11 Data cleared by the USBFS when ACLRM = 1 (1 of 2)

Number	Data cleared by setting the ACLRM bit	Situations requiring data clear
1	All data in the FIFO buffer allocated to the selected pipe (two FIFO buffers in double buffer mode)	When initializing the selected pipe

Table 29.11 Data cleared by the USBFS when ACLRM = 1 (2 of 2)

Number	Data cleared by setting the ACLRM bit	Situations requiring data clear
2	Interval count value when the selected pipe is the isochronous transfer type	When resetting the interval count value
3	Internal flags related to the PIPECFG.BFRE bit	When changing the PIPECFG.BFRE setting
4	FIFO buffer toggle control	When changing the PIPECFG.DBLB setting
5	Internal flags related to the transaction count	When forcing the transaction count function to terminate

ATREPM bit (Auto Response Mode)

The ATREPM bit enables or disables auto response mode for the selected pipe.

This bit can be set to 1 in device controller mode when the selected pipe is the bulk transfer type. When the bit is set to 1, the USBFS responds to the token from the USB host as follows:

- When the selected pipe is set for bulk IN transfers (PIPECFG.TYPE[1:0] = 01b and PIPECFG.DIR = 1):
  a. When the ATREPM bit = 1 and PID = BUF, the USBFS transmits a zero-length packet in response to the IN token.
  b. The USBFS updates the sequence toggle bit (DATA-PID) each time the USBFS receives ACK from the USB host. In a single transaction, the IN token is received, a zero-length packet is transmitted, and then ACK is received. The USBFS does not generate the BRDY or BEMP interrupt.
- When the selected pipe is set for bulk OUT transfers (PIPECFG.TYPE[1:0] = 01b and PIPECFG.DIR = 0):
  When the ATREPM bit = 1 and PID = BUF, the USBFS returns NAK in response to the OUT token and generates an NRDY interrupt.

For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode. When the selected pipe uses isochronous transfer, always set this bit to 0.

In host controller mode, always set the ATREPM bit to 0.

INBUFM bit (Transmit Buffer Monitor)

The INBUFM bit indicates the FIFO buffer status for the selected pipe in the transmitting direction.

When the selected pipe is set in the transmitting direction (PIPECFG.DIR = 1), the USBFS sets this bit to 1 when the CPU or DMA/DTC completes writing data to at least one FIFO buffer plane.

The USBFS sets this bit to 0 when the USBFS completes transmission of the data from the FIFO buffer plane to which all the data is written. In double buffer mode (PIPECFG.DBLB = 1), the USBFS sets the INBUFM bit to 0 when the USBFS completes transmission of the data from the two FIFO buffer planes before the CPU or DMA/DTC completes writing data to one FIFO buffer plane.

The INBUFM bit indicates the same value as the BSTS bit when the selected pipe is in the receiving direction (PIPECFG.DIR = 0).

BSTS bit (Buffer Status)

The BSTS bit indicates the FIFO buffer status for the selected pipe.

The meaning of the BSTS bit depends on the PIPECFG.DIR, PIPECFG.BFRE, and DnFIFOSEL.DCLRM settings, as shown in Table 29.12.

Table 29.12 BSTS bit operation

DIR value	BFRE value	DCLRM value	BSTS bit function
0	0	0	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 on completion of data read
		1	Setting prohibited
	1	0	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 when the software sets the BCLR bit in the port control register to 1 after the data read is complete
		1	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 on completion of data read
1	0	0	Sets to 1 when transmit data can be written to the FIFO buffer, and clears to 0 on completion of data write
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

29.2.36 PIPEnCTR : PIPEn Control Registers (n = 6 to 9)

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x07A + 0x2 × (n - 6)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSTS	—	—	—	—	—	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PID[1:0]	Response PID 0 0: NAK response 0 1: BUF response (depends buffer state) 1 0: STALL response 1 1: STALL response	R/W
4:2	—	These bits are read as 0. The write value should be 0.	R/W
5	PBUSY	Pipe Busy 0: Pipe n not in use for the transaction 1: Pipe n in use for the transaction	R
6	SQMON	Sequence Toggle Bit Confirmation 0: DATA0 1: DATA1	R
7	SQSET	Sequence Toggle Bit Set*1 Sets the sequence toggle bit for pipe n. 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA0	W
8	SQCLR	Sequence Toggle Bit Clear*1 Clears the sequence toggle bit for pipe n. 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0	W
9	ACLRM	Auto Buffer Clear Mode*2 0: Disable 1: Enable (all buffers initialized)	R/W
14:10	—	These bits are read as 0. The write value should be 0.	R/W
15	BSTS	Buffer Status 0: Buffer access disabled 1: Buffer access enabled	R

Note: S-TYPE-3, P-TYPE-3
Note 1. Only write 1 to the SQCLR or SQSET bit while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00b (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.
Note 2. Only set the ACLRM bit while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting this bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00b (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

PID[1:0] bits (Response PID)

The PID[1:0]bits specify the response type for the next transaction of the selected pipe.

The default PID[1:0] setting is NAK. Change the PID[1:0] setting to BUF to use the associated pipe for USB transfer. Table 29.9 and Table 29.10 show the basic operation (when there are no errors in the transmitted and received packets) of the USBFS depending on the PID[1:0] setting.

After changing the PID[1:0] setting from BUF to NAK through the software during USB communication on the selected pipe, check that the PBUSY bit is 1 to see if USB transfer on the selected pipe has actually entered the NAK state. If the USBFS changes the PID[1:0] bits to NAK, checking the PBUSY bit through the software is not necessary.

The USBFS changes the PIPEnCTR.PID[1:0] setting in the following cases:

- The USBFS sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the selected pipe
- The USBFS sets PID to NAK on detecting a USB bus reset in device controller mode
- The USBFS sets PID to NAK on detecting a reception error, such as a CRC error, three consecutive times in host controller mode
- The USBFS sets PID to STALL (11b) on receiving the STALL handshake in host controller mode

To specify each response type, set the PID[1:0] bits as follows:

- To transition from NAK (00b) to STALL, set 10b
- To transition from BUF (01b) to STALL, set 11b
- To transition from STALL (11b) to NAK, set 10b and then 00b
- To transition from STALL to BUF, set 00b (NAK) and then 01b (BUF)

PBUSY bit (Pipe Busy)

The PBUSY bit indicates whether the selected pipe is being used for the current transaction.

The USBFS changes the PBUSY bit from 0 to 1 on start of the USB transaction for the selected pipe, and changes the PBUSY bit from 1 to 0 on completion of one transaction.

Reading the PBUSY bit by software after PID is set to NAK allows you to check whether changing the pipe setting is possible.

SQMON bit (Sequence Toggle Bit Confirmation)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction of the selected pipe.

The USBFS toggles the SQMON bit on successful completion of the transaction. However, the USBFS does not toggle the SQMON bit when a DATA-PID mismatch occurs during transfer in the receiving direction.

SQSET bit (Sequence Toggle Bit Set)

Setting the SQSET bit to 1 through the software causes the USBFS to set DATA1 as the expected value of the sequence toggle bit for the next transaction on the selected pipe. The USBFS sets the SQSET bit to 0.

SQCLR bit (Sequence Toggle Bit Clear)

Setting the SQCLR bit to 1 through the software causes the USBFS to clear the expected value of the sequence toggle bit for the next transaction on the selected pipe to DATA0. The USBFS sets the SQCLR bit to 0.

ACLRM bit (Auto Buffer Clear Mode)

The ACLRM bit enables or disables auto buffer clear mode for the selected pipe. To completely clear the data in the FIFO buffer allocated to the selected pipe, write 1 and then 0 to the ACLRM bit continuously.

Table 29.13 shows the data cleared by writing 1 and 0 continuously to the ACLRM bit and the cases in which this processing is required.

Table 29.13 Data cleared by the USBFS when ACLRM = 1

Number	Data cleared by setting the ACLRM bit	Situations requiring data clear
1	All data in the FIFO buffer allocated to the selected pipe	When initializing the selected pipe
2	Interval count value when the selected pipe is the isochronous transfer type	When resetting the interval count value
3	Internal flags related to the PIPECFG.BFRE bit	When changing the PIPECFG.BFRE setting
4	Internal flags related to the transaction count	When forcing the transaction count function to terminate

BSTS bit (Buffer Status)

The BSTS bit indicates the FIFO buffer status for the selected pipe.

The meaning of the BSTS bit depends on the PIPECFG.DIR, PIPECFG.BFRE, and DnFIFOSEL.DCLRM settings, as shown in Table 29.12.

29.2.37 PIPEnTRE : PIPEn Transaction Counter Enable Register (n = 1 to 5)

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x090 + 0x4 × (n - 1)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TRENB	TRCLR	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	TRCLR	Transaction Counter Clear 0: Invalid (writing 0 has no effect) 1: Clear counter value	R/W
9	TRENB	Transaction Counter Enable 0: Disable transaction counter 1: Enable transaction counter	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Set each bit in PIPEnTRE while PID is NAK. Before setting these bits after changing the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

TRCLR bit (Transaction Counter Clear)

When the TRCLR bit sets to 1, the USBFS clears the value of the transaction counter associated with the selected pipe and then sets the TRCLR bit to 0.

TRENB bit (Transaction Counter Enable)

The TRENB bit enables or disables the transaction counter.

For receiving pipes, setting the TRENB bit to 1 after setting the total number of the packets to be received in the PIPEnTRN.TRNCNT[15:0] bits through the software allows the USBFS to control hardware on having received the number of packets equal to the TRNCNT[15:0] setting, as follows:

- When the PIPECFG.SHTNAK bit is 1, the USBFS changes the PID bits to NAK for the associated pipe on having received the number of packets equal to the TRNCNT[15:0] setting
- When the PIPECFG.BFRE bit is 1, the USBFS asserts the BRDY interrupt on having received the number of packets equal to the TRNCNT[15:0] setting and then reading the last received data

For transmitting pipes, set the TRENB bit to 0.

When the transaction counter is not used, set this bit to 0. When the transaction counter is used, set the TRNCNT[15:0] bits before setting this bit to 1. Set this bit to 1 before receiving the first packet to be counted by the transaction counter.

29.2.38 PIPEnTRN : PIPEn Transaction Counter Register (n = 1 to 5)

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x092 + 0x4 × (n - 1)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TRNCNT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TRNCNT[15:0]	Transaction Counter When written to, this bit specifies the total packets (number of transactions) to be received by the selected pipe. When read from, when PIPEnTRE.TRENB is 0, this bit indicates the specified number of transactions. When PIPEnTRE.TRENB is 1, this bit indicates the current transaction count.	R/W

Note: S-TYPE-3, P-TYPE-3

The PIPEnTRN registers retain their settings during a USB bus reset.

TRNCNT[15:0] bits (Transaction Counter)

The USBFS increments the value of the TRNCNT[15:0] bits by 1 when all of the following conditions are satisfied on receiving the packet:

- The PIPEnTRE.TRENB bit = 1
- (TRNCNT[15:0] set value ≠ current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the PIPEMAXP.MXPS[9:0] setting

The USBFS clears the value of the TRNCNT[15:0] bits to 0 when any of the following conditions are satisfied:

All of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- (TRNCNT[15:0] set value = current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the PIPEMAXP.MXPS[9:0] setting

Both of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- The USBFS received a short packet

Both of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- The PIPEnTRE.TRCLR bit was set to 1 by software

For transmitting pipes, set the TRNCNT[15:0] bits to 0. When the transaction counter is not used, set the TRNCNT[15:0] bits to 0.

Setting the number of transactions to be transferred to the TRNCNT[15:0] bits is only enabled when the PIPEnTRE.TRENB bit is 0. To set the number of transactions to be transferred, set the TRCLR bit to 1 to clear the current counter value before setting the PIPEnTRE.TRENB bit to 1.

29.2.39 DEVADDn : Device Address n Configuration Register (n = 0 to 5)

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x0D0 + 0x2 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	USBSPD[1:0]		—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
7:6	USBSPD[1:0]	Transfer Speed of Communication Target Device 0 0: Do not use DEVADDn 0 1: Low-speed 1 0: Full-speed 1 1: Setting prohibited	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The DEVADDn register specifies the transfer speed of the peripheral device that is the communication target for pipes 0 to 9.

In host controller mode, set all DEVADDn bits before starting communication to any pipes. Only change the bits in DEVADDn when no valid pipes are using the bit settings. A valid pipe is defined as one that satisfies both of the following conditions:

- The target device of the DEVADDn register is selected in the DEVSEL[3:0] bits
- The PID[1:0] bits are set to BUF for the selected pipe, or the selected pipe is the DCP with the DCPCTR.SUREQ bit set to 1

In device controller mode, set all bits in this register to 0.

USBSPD[1:0] bits (Transfer Speed of Communication Target Device)

TheUSBSPD[1:0] bits specify the USB transfer speed of the target peripheral device. Set these bits to 10b when a full-speed device is connected through the hub. In host controller mode, the USBFS generates packets based on the USBSPD[1:0] setting. In device controller mode, set these bits to 00b.

29.2.40 DPUSR0R : Deep Software Standby USB Transceiver Control/Pin Monitor Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x400

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DVBSTS0	—	DOVCB0	DOVCA0	—	—	DM0	DP0
Value after reset:	0	0	0	0	0	0	0	0	x	0	x	x	0	0	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	FIXPHY0	DRPD0	—	RPUE0	SRPC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SRPC0*1	USB Single-ended Receiver Control 0: Disable input through DP and DM inputs 1: Enable input through DP and DM inputs	R/W
1	RPUE0*1	DP Pull-Up Resistor Control 0: Disable DP pull-up resistor 1: Enable DP pull-up resistor	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	DRPD0*1	D+/D- Pull-Down Resistor Control 0: Disable DP/DM pull-down resistor 1: Enable DP/DM pull-down resistor	R/W
4	FIXPHY0	USB Transceiver Output Fix 0: Fix outputs in Normal mode and on return from Deep Software Standby mode 1 1: Fix outputs on transition to Deep Software Standby mode 1	R/W
15:5	—	These bits are read as 0. The write value should be 0.	R/W
16	DP0	USB D+ Input Indicates D+ input signal on the USBFS side	R
17	DM0	USB D- Input Indicates D– input signal on the USBFS side	R
19:18	—	These bits are read as 0. The write value should be 0.	R/W
20	DOVCA0	USB OVRCURA-DS Input*2 Indicates OVRCURA-DS input signal on the USBFS side	R
21	DOVCB0	USB OVRCURB-DS Input*2 Indicates OVRCURB-DS input signal on the USBFS side	R
22	—	The read value is undefined. The write value should be 0.	R/W
23	DVBSTS0	USB VBUS Input Indicates VBUS input signal on the USBFS side	R
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This register is accessed by the clock of PCLKB/64 to reduce current consumption in Deep Software Standby mode1.
Note 1. Use this bit during operation in Deep Software Standby mode 1. For details, see section 29.3.1.5. Release from Deep Software Standby mode1 because of USB suspend/resume interrupts.
Note 2. OVRCURA or OVRCURB can not be used. Only OVRCURA-DS or OVRCURB-DS can be used in Deep software standby mode1.

SRPC0 bit (USB Single-ended Receiver Control)

The SRPC0 bit controls the D+ and D- inputs of the USB transceiver. In host controller mode, set this bit to 1. In device controller mode, set this bit to 0 when disconnected, set to 1 when suspended. This bit is only valid when the FIXPHY0 bit is 1.

FIXPHY0 bit (USB Transceiver Output Fix)

The FIXPHY0 bit keeps the outputs of the USB transceiver disabled.

29.2.41 DPUSR1R : Deep Software Standby USB Suspend/Resume Interrupt Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x404

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DVBINT0	—	DOVRCRB0	DOVRCRA0	—	—	DMINT0	DPINT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	DVBSE0	—	DOVRCRBE0	DOVRCRAE0	—	—	DMINTE0	DPINTE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DPINTE0	USB DP Interrupt Enable/Clear 0: Disable recovery from Deep Software Standby mode 1 by DP input 1: Enable recovery from Deep Software Standby mode 1 by DP input	R/W
1	DMINTE0	USB DM Interrupt Enable/Clear 0: Disable recovery from Deep Software Standby mode 1 by DM input 1: Enable recovery from Deep Software Standby mode by DM input	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	DOVRCRAE0	USB OVRCURA-DS Interrupt Enable/Clear*1 0: Disable recovery from Deep Software Standby mode 1 by OVRCURA-DS input 1: Enable recovery from Deep Software Standby mode 1 by OVRCURA-DS input	R/W
5	DOVRCRBE0	USB OVRCURB-DS Interrupt Enable/Clear*1 0: Disable recovery from Deep Software Standby mode 1 by OVRCURB-DS input 1: Enable recovery from Deep Software Standby mode 1 by OVRCURB-DS input	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	DVBSE0	USB VBUS Interrupt Enable/Clear 0: Disable recovery from Deep Software Standby mode 1 by VBUS input 1: Enable recovery from Deep Software Standby mode 1 by VBUS input	R/W
15:8	—	These bits are read as 0. The write value should be 0..	R/W
16	DPINT0	USB DP Interrupt Source Recovery 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1 because of DP	R
17	DMINT0	USB DM Interrupt Source Recovery 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1 because of DM input	R
19:18	—	These bits are read as 0. The write value should be 0.	R/W
20	DOVRCRA0	USB OVRCURA-DS Interrupt Source Recovery*1 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1 because of OVRCURA-DS input	R
21	DOVRCRB0	USB OVRCURB-DS Interrupt Source Recovery*1 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1 because of OVRCURB-DS input	R
22	—	This bit is read as 0. The write value should be 0.	R/W
23	DVBINT0	USB VBUS Interrupt Source Recovery 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1 because of VBUS input	R
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This register is accessed by the clock of PCLKB/64 to reduce current consumption in Deep Software Standby mode1.
Note 1. OVRCURA or OVRCURB can not be used. Only OVRCURA-DS or OVRCURB-DS can be used in Deep Software Standby mode1.

DPINTE0 bit (USB DP Interrupt Enable/Clear)

The DPINTE0 bit enables or disables triggering of recovery from Deep Software Standby mode 1 by the DP input of the USBFS. Writing 0 to this bit while the DPINT0 bit is 1 sets the DPINT0 bit to 0.

DMINTE0 bit (USB DM Interrupt Enable/Clear)

The DMINTE0 bit enables or disables triggering of recovery from Deep Software Standby mode 1 by the DM input of the USBFS. Writing 0 to this bit while the DMINT0 bit is 1 clears the DMINT0 bit to 0.

DOVRCRAE0 bit (USB OVRCURA-DS Interrupt Enable/Clear)

The DOVRCRAE0 bit enables or disables triggering of recovery from Deep Software Standby mode 1 by the OVRCURA-DS input of the USBFS. Writing 0 to this bit while the DOVRCRA0 bit is 1 clears the DOVRCRA0 bit to 0.

DOVRCRBE0 bit (USB OVRCURB-DS Interrupt Enable/Clear)

The DOVRCRBE0 bit enables or disables triggering of recovery from Deep Software Standby mode 1 by the OVRCURB-DS input of the USBFS. Writing 0 to this bit while the DOVRCRB0 bit is 1 clears the DOVRCRB0 bit to 0.

DVBSE0 bit (USB VBUS Interrupt Enable/Clear)

The DVBSE0 bit enables or disables triggering of recovery from Deep Software Standby mode 1 by the VBUS input of the USBFS. Writing 0 to this bit while the DVBINT0 bit is 1 clears the DVBINT0 bit to 0.

DPINT0 bit (USB DP Interrupt Source Recovery)

The DPINT0 bit indicates that the system has returned from Deep Software Standby mode 1 because of the DP input of the USBFS. This recovery is only enabled when the DPINTE0 bit is 1. Writing 0 to the DPINTE0 bit while this bit is 1 clears this bit to 0.

DMINT0 bit (USB DM Interrupt Source Recovery)

The DMINT0 bit indicates that the system has returned from Deep Software Standby mode 1 because of the DM input of the USBFS. This recovery is only enabled when the DMINTE0 bit is 1. Writing 0 to the DMINTE0 bit while this bit is 1 clears this bit to 0.

DOVRCRA0 bit (USB OVRCURA-DS Interrupt Source Recovery)

The DOVRCRA0 bit indicates that the system has returned from Deep Software Standby mode 1 because of the OVRCURA-DS input of the USBFS. This recovery is only enabled when the DOVRCRAE0 bit is 1. Writing 0 to the DOVRCRAE0 bit while this bit is 1 clears this bit to 0.

DOVRCRB0 bit (USB OVRCURB-DS Interrupt Source Recovery)

The DOVRCRB0 bit indicates that the system has returned from Deep Software Standby mode 1 because of the OVRCURB-DS input of the USBFS. This recovery is only enabled when the DOVRCRBE0 bit is 1. Writing 0 to the DOVRCRBE0 bit while this bit is 1 clears this bit to 0.

DVBINT0 bit (USB VBUS Interrupt Source Recovery)

The DVBINT0 bit indicates that the system has returned from Deep Software Standby mode 1 because of the VBUS input of the USBFS. This recovery is only enabled when the DVBSE0 bit is 1. Writing 0 to the DVBSE0 bit while this bit is 1 clears this bit to 0.

29.3 Operation

29.3.1 System Control

This section describes register settings required for initializing the USBFS and controlling power consumption.

29.3.1.1 Setting data to the USBFS registers

Setting the SYSCFG.USBE bit to 1 after starting the clock supply (SYSCFG.SCKE bit = 1) enables and starts USBFS operation.

29.3.1.2 Selecting the controller function

The USBFS can operate as either a host or device controller.

Use the SYSCFG.DCFM bit to select one of these USBFS functions. The DCFM bit must be changed in the initial settings immediately after a reset or in the D+ pull-up-disabled state (SYSCFG.DPRPU bit = 0) and D+ and D– pull-down-disabled state (SYSCFG.DRPD bit = 0).

29.3.1.3 Controlling the USB data bus using resistors

The USBFS provides pull-up and pull-down resistors for the D+ and D– lines. Pull these lines up or down by setting the SYSCFG.DPRPU and DRPD bits.

In device controller mode, confirm that connection to the USB host is made, and then set the SYSCFG.DPRPU bit to 1 and pull up the D+ line (in full-speed communication).

When the SYSCFG.DPRPU bit is set to 0 during communication with a PC, the USBFS disables the pull-up resistor of the USB data line, thereby notifying the USB host of disconnection.

In host controller mode, set the SYSCFG.DRPD bit to 1 to pull down the D+ and D– lines.

Table 29.14 USB data bus resistor control

SYSCFG register settings		USB data bus control		
DRPD bit	DPRPU bit	D–	D+	Function
0	0	Open	Open	When resistors not used
0	1	Open	Pull-up	When operating as a device controller at full-speed
1	0	Pull-down	Pull-down	When operating as a host controller
1	1	—	—	Setting prohibited

29.3.1.4 Example external connection circuits

Figure 29.2 shows an example OTG connection in the self-powered system. The USBFS controls the pull-up resistor of the D+ line and the pull-down resistor of D+ and D- lines. Select pull-up and pull-down for the lines in the SYSCFG.DPRPU and SYSCFG.DRPD bits. In device controller mode, the pull-up resistor of USB data line is disabled if SYSCFG.DPRPU bit is set to 0 while communicating with the USB host. The USBFS can use this to notify the USB host of a device disconnect.

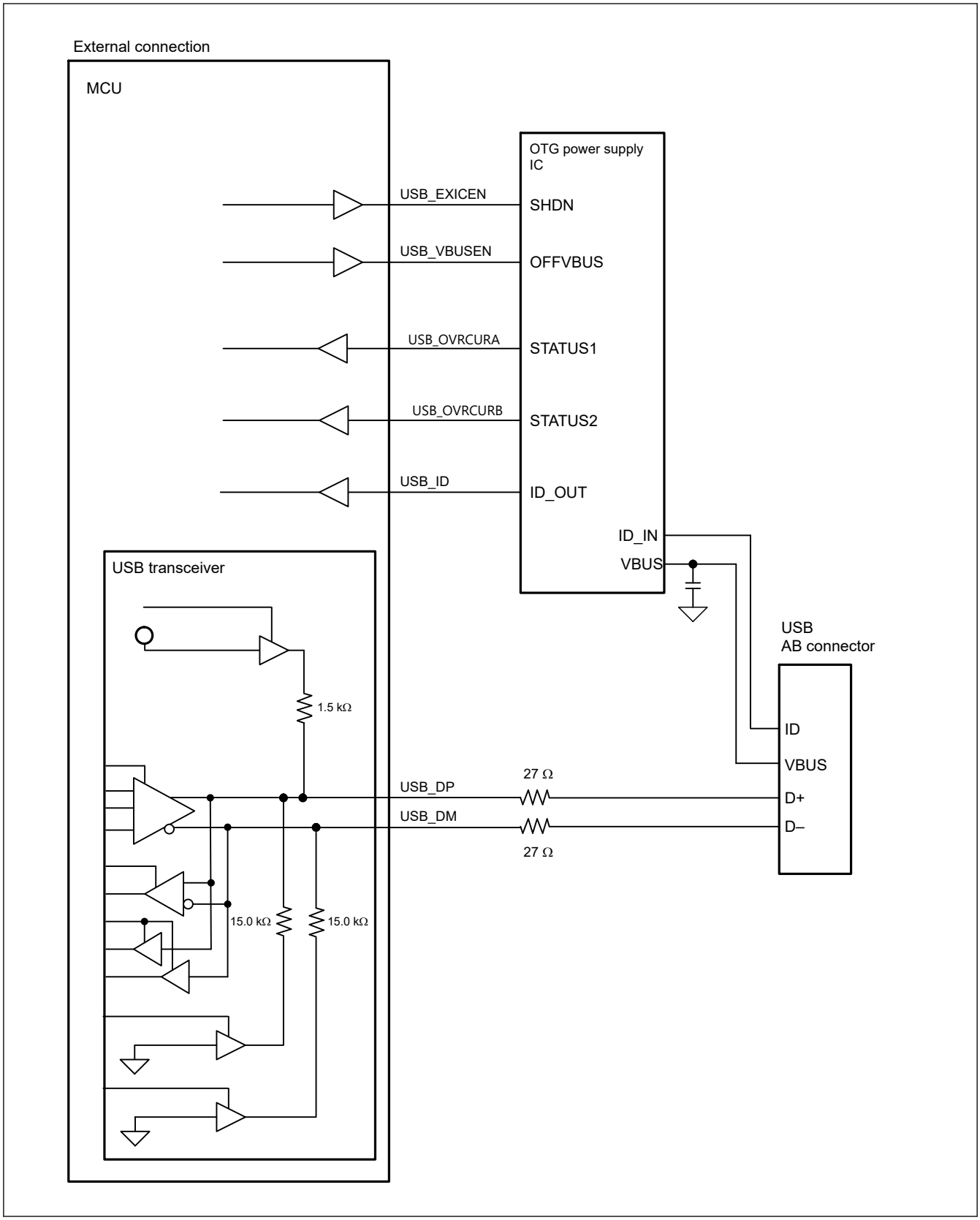


Figure 29.2 Example OTG connection in a self-powered system

Figure 29.3 shows an example device connection in a self-powered system.

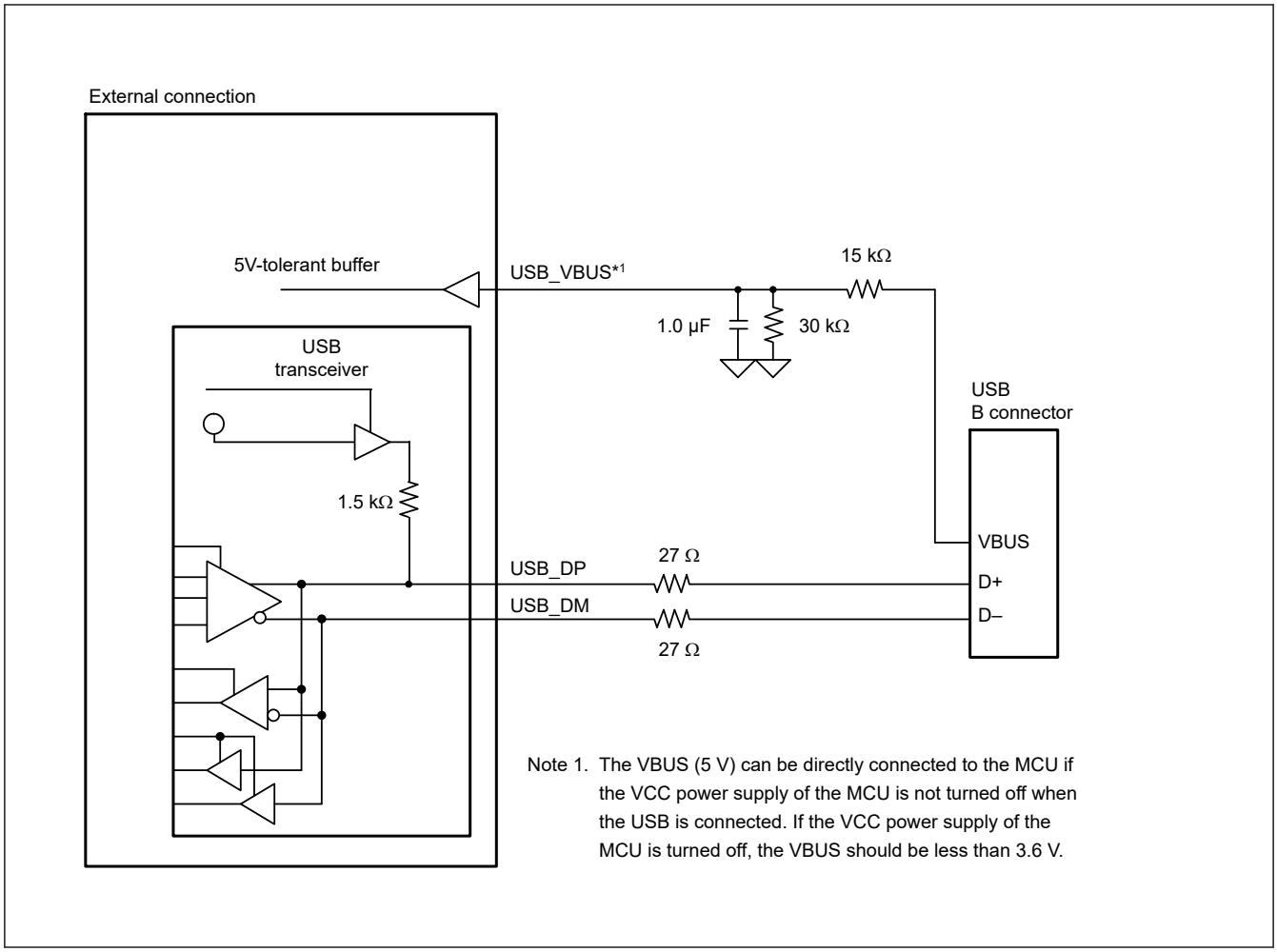


Figure 29.3 Example device connection in a self-powered system

Figure 29.4 shows an example host connection.

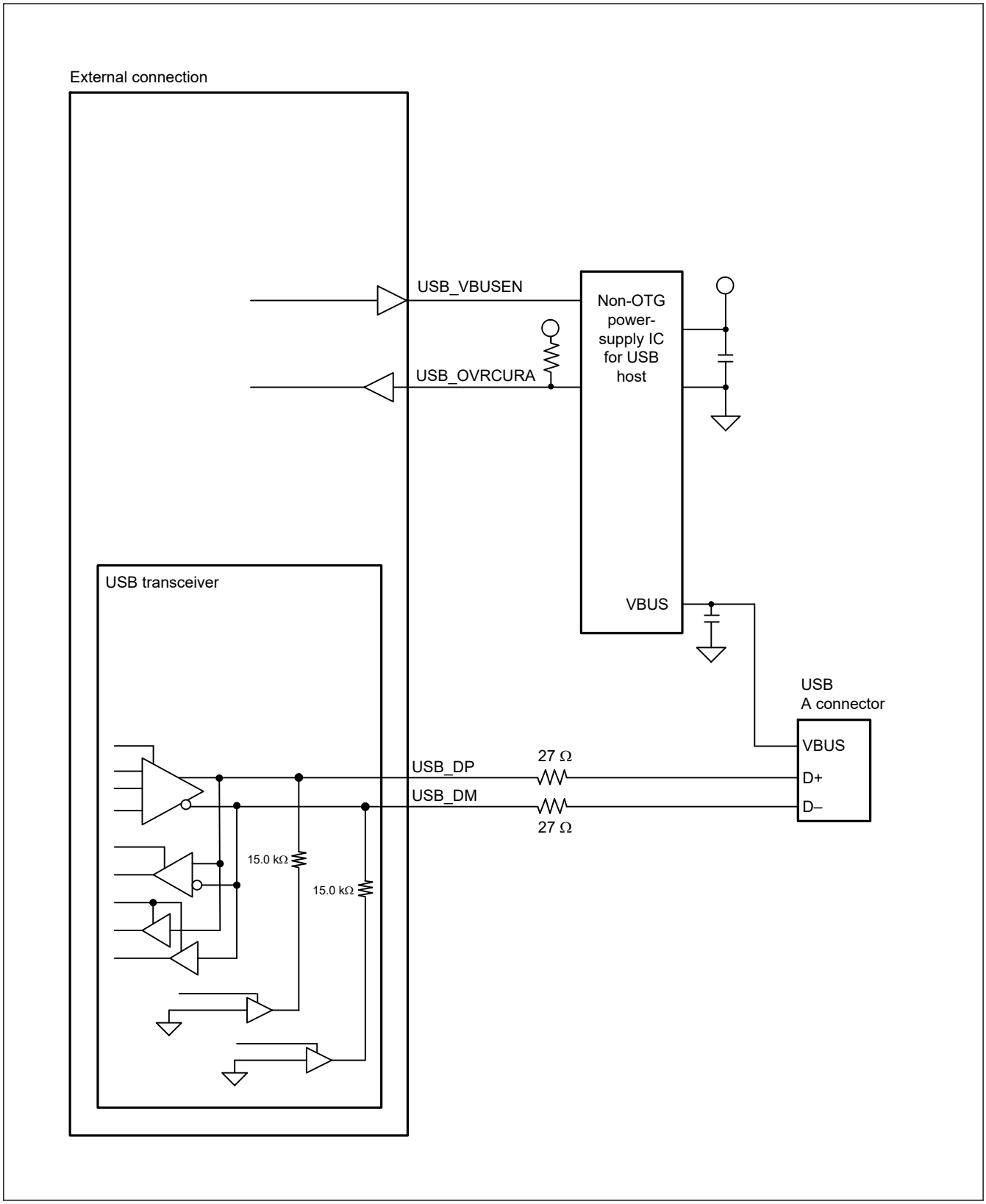


Figure 29.4 Example host connection

Figure 29.5 shows an example device connection in a bus-powered system.

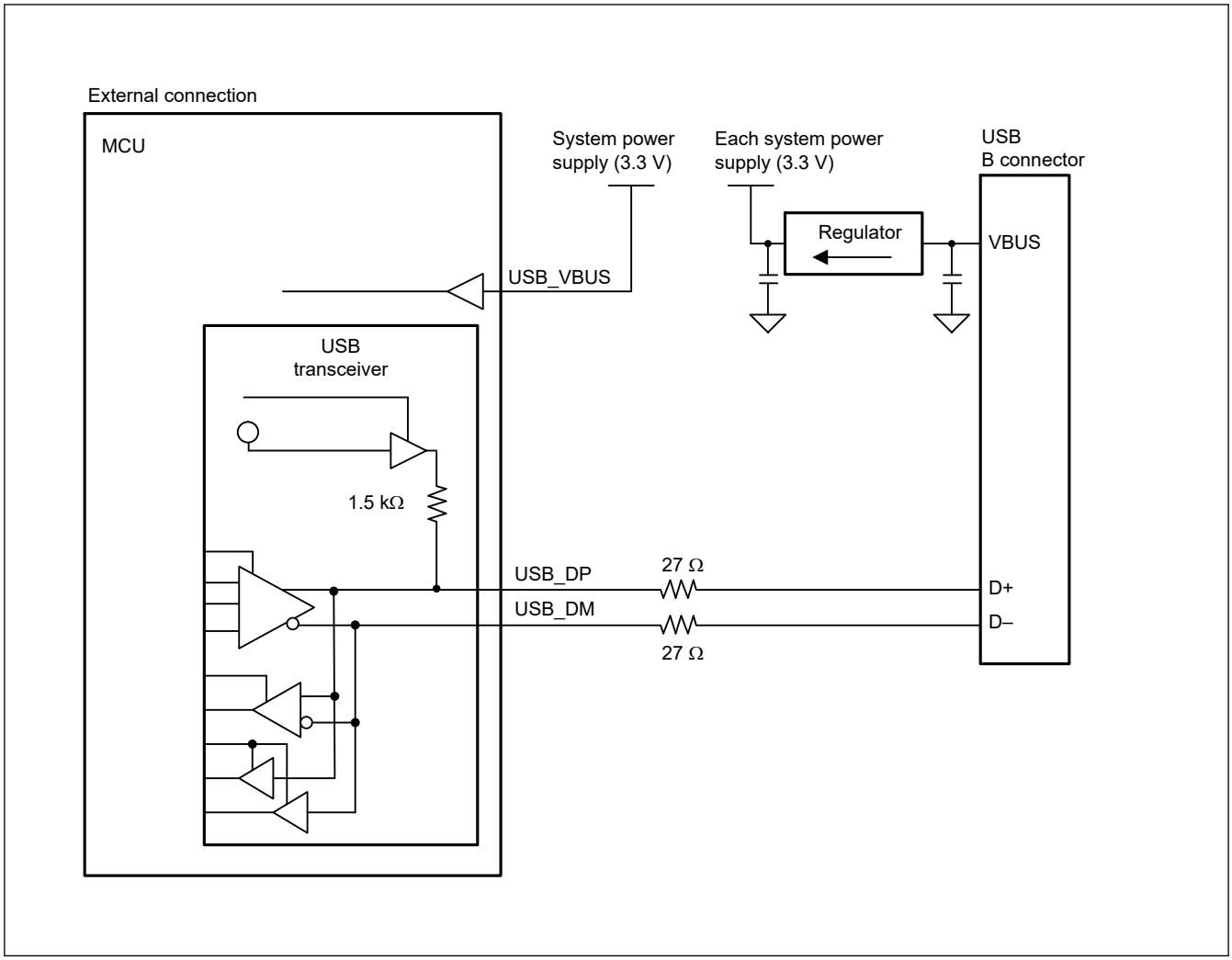


Figure 29.5 Example device connection in a bus-powered state

The examples of external circuits given in this section are simplified circuits, and their operation in every system is not guaranteed.

29.3.1.5 Release from Deep Software Standby mode1 because of USB suspend/resume interrupts

Deep Software Standby mode1 can be canceled by a USB suspend/resume interrupt. USB suspend/resume interrupts are detected by the USB resume detecting unit, which controls and monitors the USB I/O pins to detect the interrupts.

Figure 29.6 shows a schematic diagram of the connection between the USB resume detecting unit and the USB I/O pins.

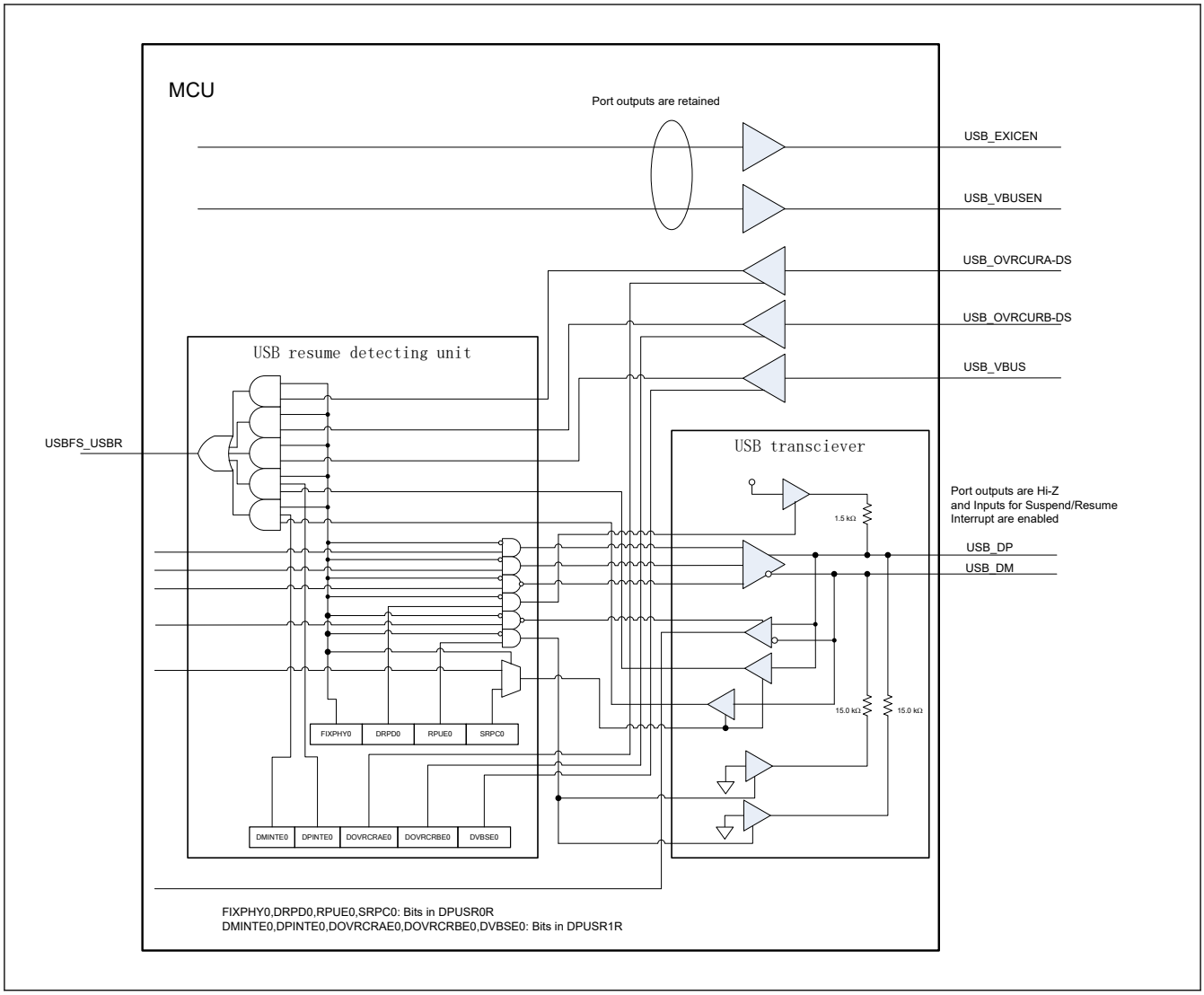


Figure 29.6 Connection between the USB resume detecting unit and the USB I/O pins

Table 29.15 shows the USB suspend and resume interrupt sources and their associated I/O pins.

Table 29.15 USB suspend and resume interrupt sources and their associated I/O pins

USB operating mode	Source	Pin name
Device, OTG	Resume	USB_DP
Host, OTG	Attach or detach	USB_DP, USB_DM
Device	Attach or detach	USB_VBUS
Host	Overcurrent detection	USB_OVRCURA-DS
OTG	Overcurrent detection	USB_OVRCURA-DS, USB_OVRCURB-DS

Figure 29.7 shows the flow for setting the USBFS when entering Deep Software Standby mode 1 from either host or device controller mode. Figure 29.8 shows the flow for setting the USBFS when canceling Deep Software Standby mode 1 from host controller mode. Figure 29.9 shows the flow for setting the USBFS when canceling Deep Software Standby mode 1 from device controller mode.

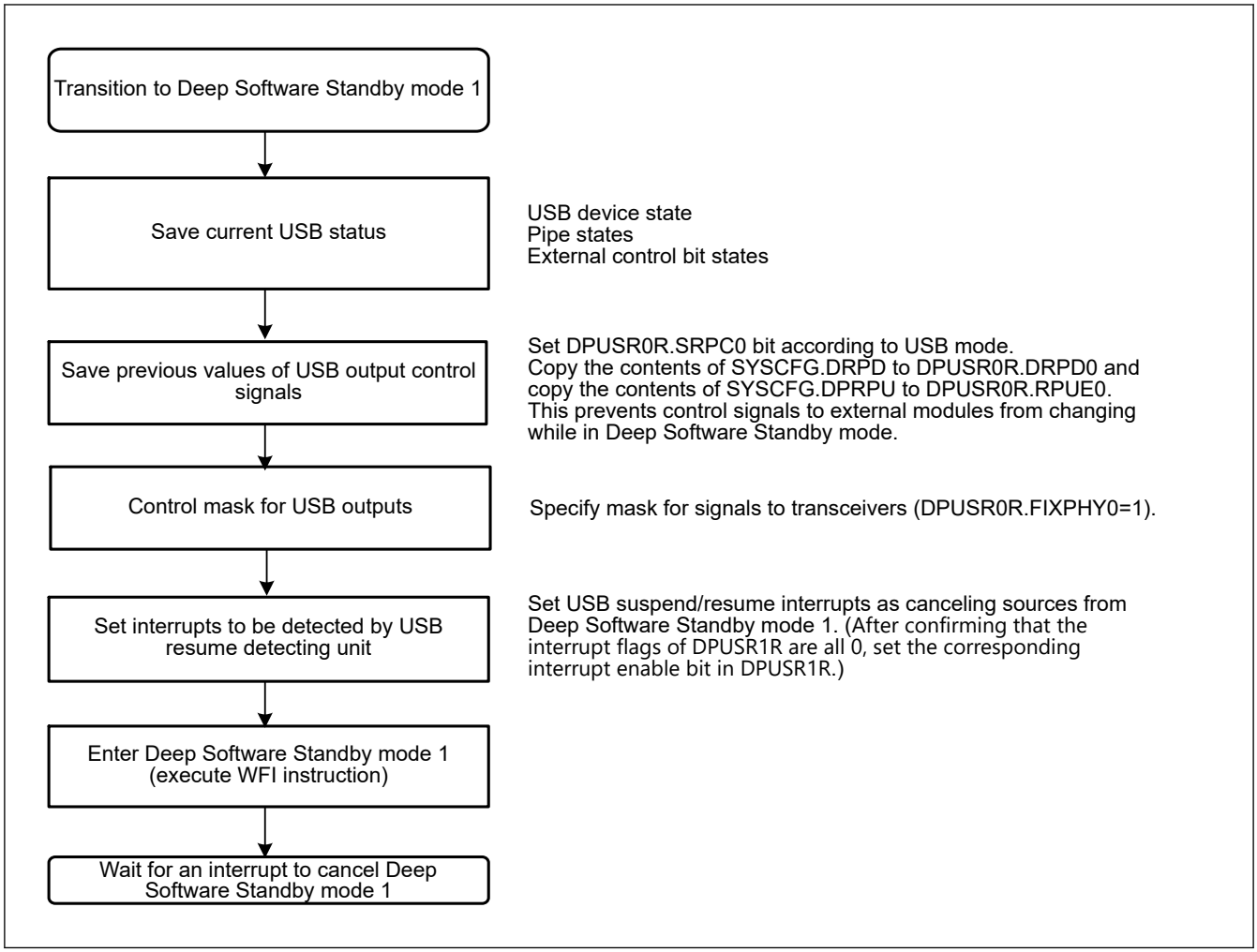


Figure 29.7 USBFS setup flow for transition to Deep Software Standby mode 1 as host or device controller

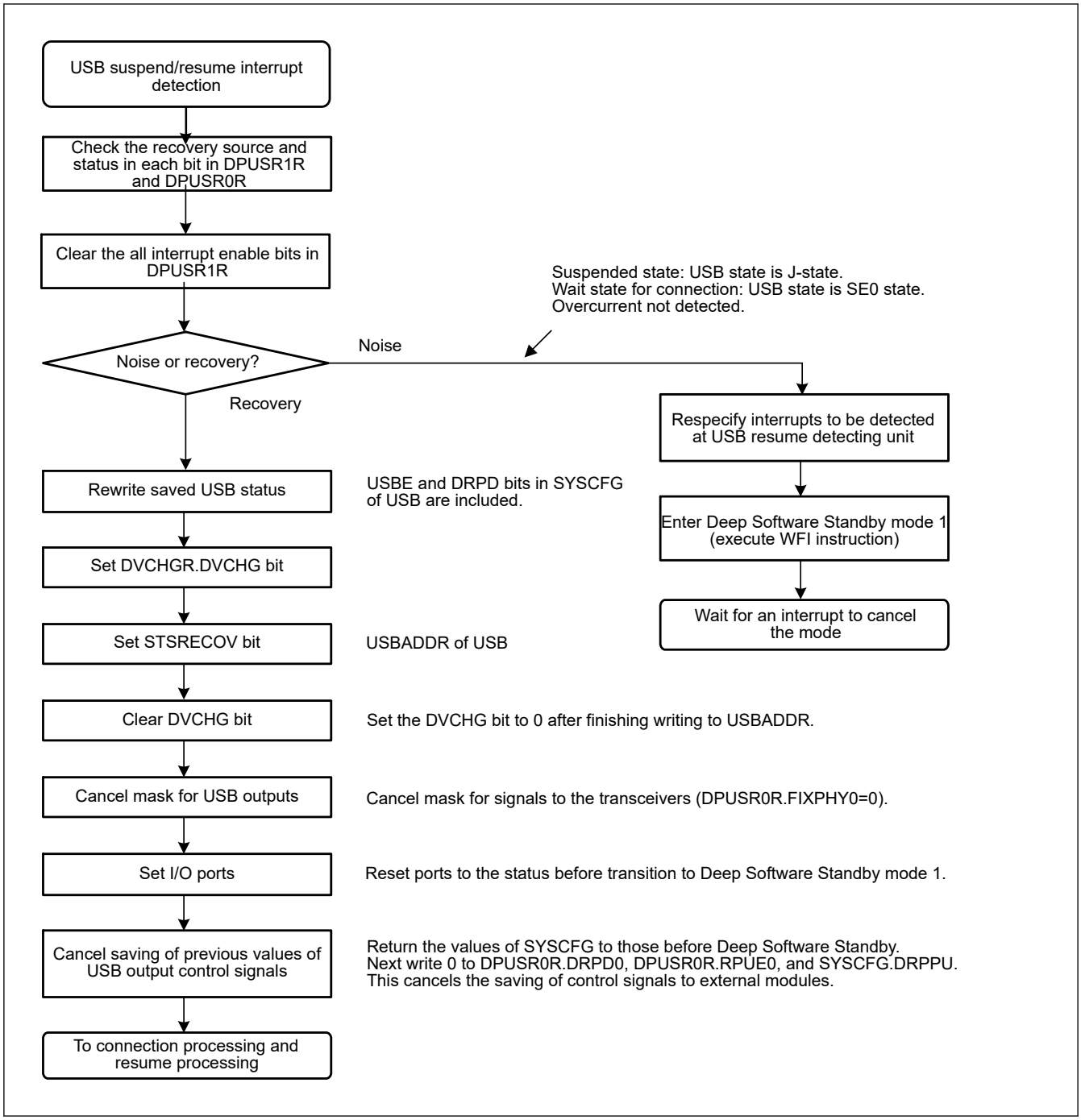


Figure 29.8 USBFS setup flow for canceling Deep Software Standby mode 1 as host controller

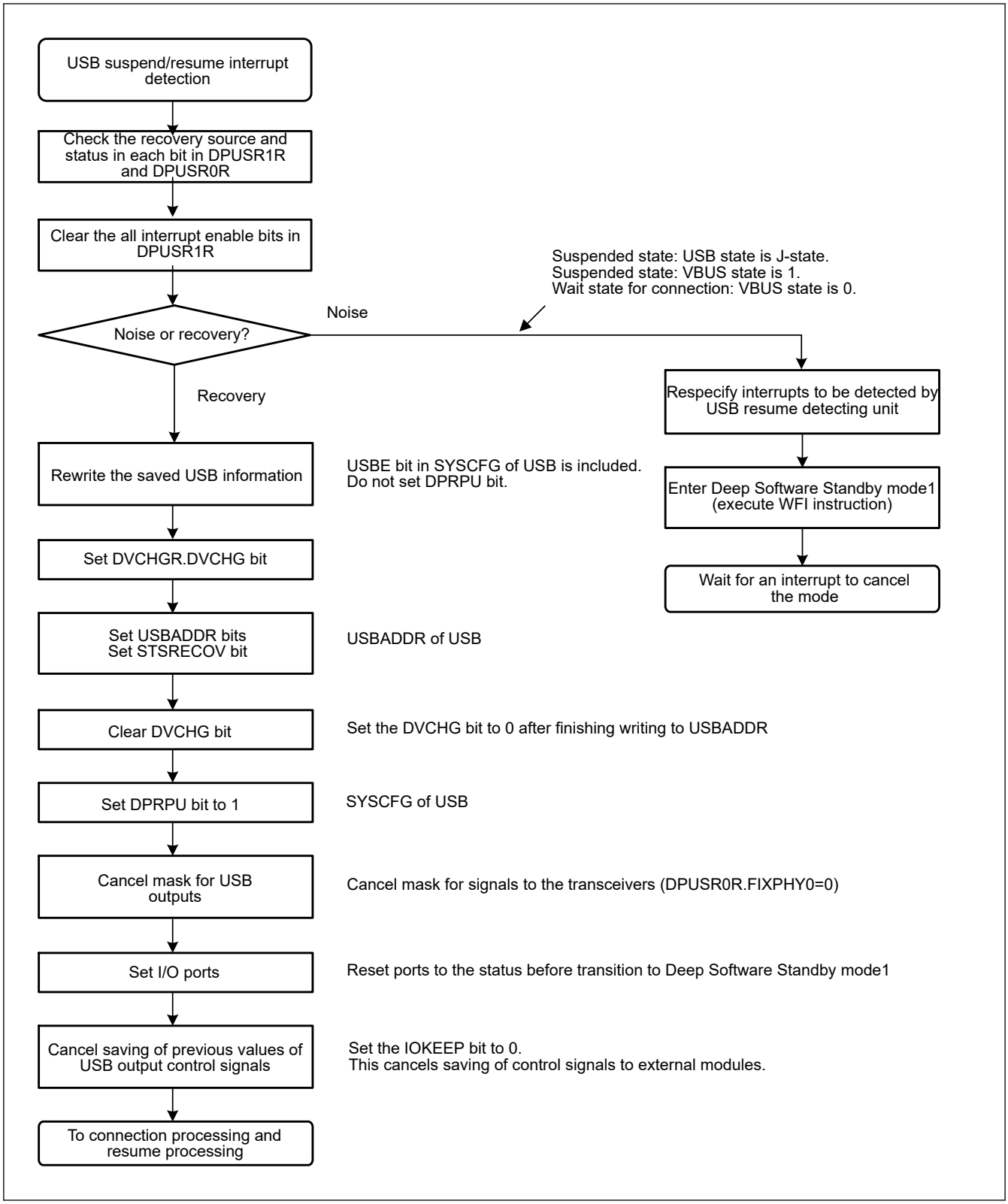


Figure 29.9 USBFS setup flow for canceling Deep Software Standby mode 1 as device controller

29.3.2 Interrupts

Table 29.16 lists the interrupt sources in the USBFS. When an interrupt generation condition is satisfied and the interrupt output is enabled using the associated interrupt enable register, a USBFS interrupt request is issued to the Interrupt Controller Unit (ICU) and an USBFS interrupt is generated.

Table 29.16 Interrupt sources (1 of 2)

Bit to be set to 1	Name	Interrupt source	Applicable controller function	Status flag
VBINT	VBUS interrupt	• A change in the state of the USB_VBUS input pin was detected (low to high or high to low)	Host or device*1	INTSTS0.VBSTS
RESM	Resume interrupt	• A change in the state of the USB bus was detected in the Suspend state (J-state to K-state or J-state to SE0)	Device	—
SOFR	Frame number update interrupt	In host controller mode: • An SOF packet with a different frame number was transmitted In device controller mode: • An SOF packet with a different frame number was received	Host or device	—
DVST	Device state transition interrupt	• One of the following device state transitions was detected: – USB bus reset was detected – Suspend state was detected – SET_ADDRESS request was received – SET_CONFIGURATION request was received	Device	INTSTS0.DVSQ[2:0]
CTRT	Control transfer stage transition interrupt	• A control transfer stage transition was detected because of one of the following: – Setup stage completed – Control write transfer status stage transition occurred – Control read transfer status stage transition occurred – Control transfer completed – Control transfer sequence error occurred	Device	INTSTS0.CTSQ[2:0]
BEMP	Buffer empty interrupt	• The buffer is empty after all FIFO buffer data was transmitted • A packet larger than the maximum packet size was received	Host or device	BEMPSTS.PIPEnBEMP
NRDY	Buffer not ready interrupt	In host controller mode • A STALL response was received from the peripheral device in response to the issued token • The response from the peripheral device in response to the issued token was not received successfully (no response three times consecutively or packet reception error three times consecutively) • An overrun or underrun error occurred during isochronous transfer In device controller mode • NAK was returned for an IN or OUT token while the PID [1:0] bits were set to 01b (BUF) • A CRC error or bit stuffing error occurred during data reception in isochronous transfer • An overrun or underrun occurred during data reception in isochronous transfer	Host or device	NRDYSTS.PIPEnNRDY
BRDY	Buffer ready interrupt	• The buffer is ready (readable or writable state)	Host or device	BRDYSTS.PIPEnBRDY
OVRCR	Overcurrent input change interrupt	• USB_OVRCURA, USB_OVRCURA-DS, USB_OVRCURB or USB_OVRCURB-DS input pin state change was detected (low to high or high to low)	Host	INTSTS1.OVRCR
BCHG	Bus change interrupt	• USB bus state change was detected	Host or device	SYSSTS0.LNST[1:0]
DTCH	Disconnect detection during full-speed operation	Peripheral device disconnect was detected in full-speed operation	Host	DVSTCTR0.RHST[2:0]
ATTCH	Device connect detection interrupt	• J-state or K-state was detected on the USB bus for 2.5 µs continuously This interrupt can be used to check whether peripheral devices are connected	Host	—

Table 29.16 Interrupt sources (2 of 2)

Bit to be set to 1	Name	Interrupt source	Applicable controller function	Status flag
EOFERR	EOF error detection interrupt	• An EOF error was detected for a peripheral device	Host	—
SACK	Setup normal interrupt	• A setup transaction normal response (ACK) was received	Host	—
SIGN	Setup error interrupt	• A setup transaction error (no response or ACK packet corruption) was detected three consecutive times	Host	—

Note 1. Although this interrupt can be generated in host controller mode, it is not usually used in this mode.

Figure 29.10 shows the circuits related to the USBFS interrupts.

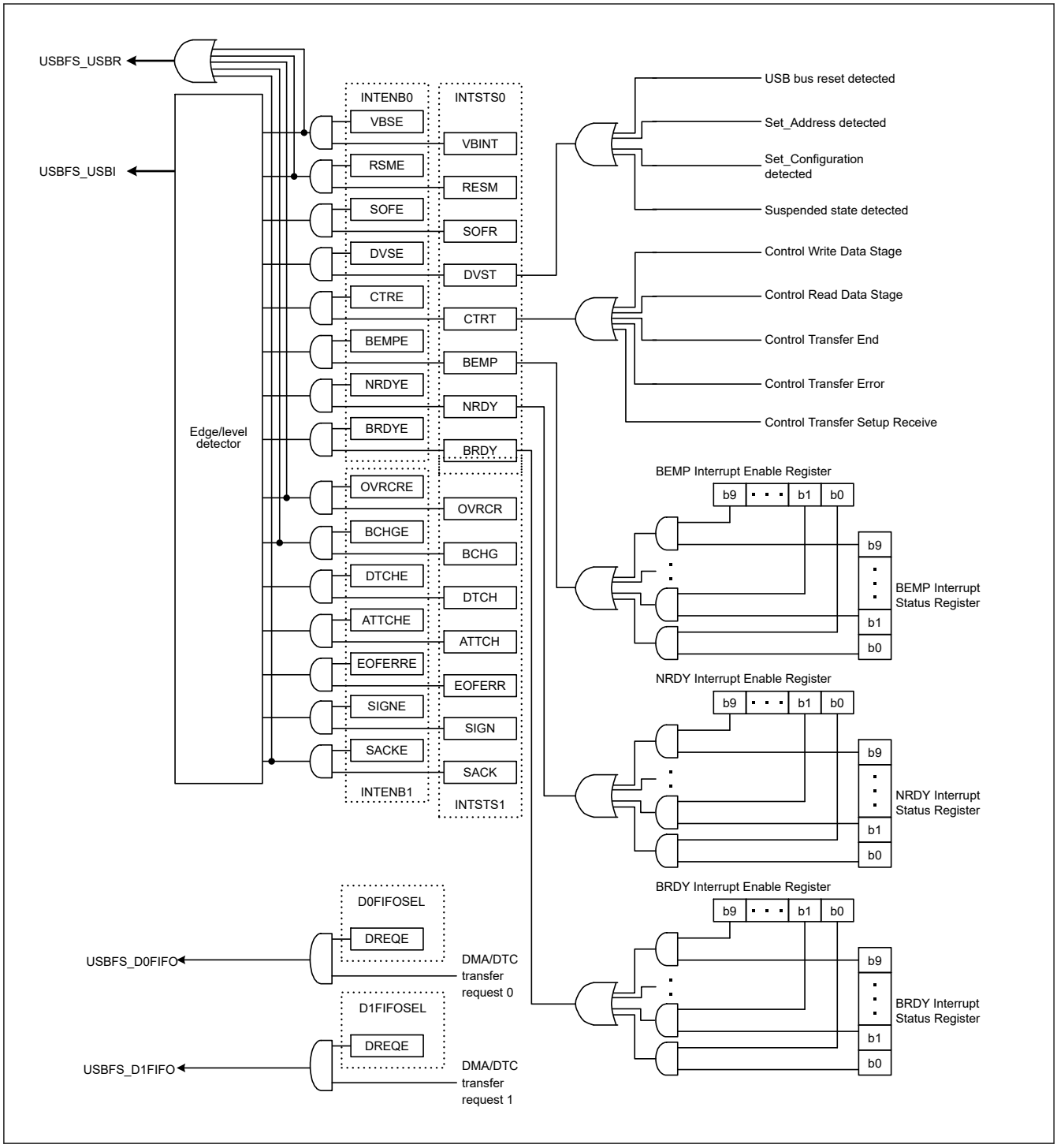


Figure 29.10 USBFS interrupt-related circuits

Table 29.17 shows the interrupts generated by the USBFS.

Table 29.17 USBFS interrupts

Interrupt name	Interrupt status flag	DTC activation	DMAC activation	Priority
USBFS_D0FIFO	DMA transfer request 0	Possible	Possible	High
USBFS_D1FIFO	DMA transfer request 1	Possible	Possible	↑
USBFS_USBI	VBUS interrupt, resume interrupt, frame number update interrupt, device state transition interrupt, control transfer stage transition interrupt, buffer empty interrupt, buffer not ready interrupt, buffer ready interrupt, overcurrent input change interrupt, bus change interrupt, disconnect detection interrupt during full-speed operation, device connect detection interrupt, EOF error detection interrupt, normal setup operation interrupt, and setup error interrupt	Not possible	Not possible	Low
USBFS_USBR*1	VBUS interrupt, resume interrupt, overcurrent input change interrupt and setup error interrupt	Not possible	Not possible	—

Note 1. Software Standby mode can be canceled. And Deep Software Standby mode 1 also can be canceled, but only dedicated pin (OVRCURA-DS, OVRCURB-DS) can be used for overcurrent pins.

29.3.3 Interrupt Descriptions

29.3.3.1 BRDY interrupt

The BRDY interrupt is generated in both host and device controller modes. This section describes the conditions in which the USBFS sets the associated bit in BRDYSTS to 1. Under these conditions, the USBFS generates a BRDY interrupt if the software has set the bit in BRDYENB associated with the given pipe to 1 and the INTENB0.BRDYE bit to 1.

The conditions for generating and clearing the BRDY interrupt depend on the SOFCFG.BRDYM and PIPECFG.BFRE settings for each pipe as follows:

(1) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.

On any of the following conditions, the USBFS generates an internal BRDY interrupt request trigger and sets the BRDYSTS.PIPEnBRDY bit associated with the selected pipe to 1.

For transmitting pipes

- When the DIR bit is changed from 0 to 1 by software
- When packet transmission is complete for a pipe while write-access from the CPU to the FIFO buffer for the pipe is disabled (when the BSTS bit is read as 0)
- When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode
- No request trigger is generated until completion of writing data to the currently-written FIFO buffer even if transmission to the other FIFO buffer is complete
- When the hardware flushes the buffer of the pipe for isochronous transfers
- When 1 is written to the PIPEnCTR.ACLRM bit, which causes the FIFO buffer to transition from the write-disabled to write-enabled state

No request trigger is generated for the DCP, that is, during data transmission for control transfers.

For receiving pipes

- When packet reception is successfully complete, enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the given pipe is disabled (when the BSTS bit is read as 0). No request trigger is generated for transactions in which a DATA-PID mismatch has occurred.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode. No request trigger is generated until completion of reading data from the currently-read FIFO buffer, even if reception by the other FIFO buffer is complete.

In device controller mode, the BRDY interrupt is not generated in the status stage of control transfers. The PIPEBRDY interrupt status of the selected pipe can be set to 0 by writing 0 to the associated PIPEnBRDY bit through software. In this case, the other PIPEBRDY bit should be set to 1.

Clear the BRDY status before accessing the FIFO buffer.

(2) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 1

With these settings, the USBFS generates a BRDY interrupt on completion of reading all data for a single transfer using the receiving pipe, and sets the bit in BRDYSTS associated with the pipe to 1.

On any of the following conditions, the USBFS determines that the last data for a single transfer was received.

- When a short packet including a zero-length packet is received
- When the PIPEn transaction counter register (PIPEnTRN) is used and the number of packets specified in the PIPEnTRN.TRNCNT[15:0] bits are completely received

When the data is completely read after any of these conditions is satisfied, the USBFS determines that all data for a single transfer is completely read.

When a zero-length packet is received while the FIFO buffer is empty, the USBFS determines that all data for a single transfer is completely read when the FRDY bit in the FIFO port control register is 1 and the DTLN[8:0] bits are 0. In this case, to start the next transfer, write 1 to the BCLR bit in the associated port control register through the software. With these settings, the USBFS does not detect a BRDY interrupt for the transmitting pipe.

The PIPEBRDY interrupt status of a pipe can be set to 0 by writing 0 to the associated BRDYSTS.PIPEnBRDY bit through the software. In this case, 1 must be written to the PIPEBRDY bits for the other pipes.

In this mode, do not change the PIPECFG.BFRE bit setting until all data for a single transfer is processed. When it is necessary to change the PIPECFG.BFRE bit before completion of processing, all FIFO buffers for the pipe must be cleared using the PIPEnCTR.ACLRM bit.

(3) When SOFCFG.BRDYM = 1 and PIPECFG.BFRE = 0

With these settings, the BRDYSTS.PIPEnBRDY values are linked to the BSTS bit setting for each pipe. In other words, the BRDY interrupt status bits (PIPEBRDY) are set to 1 or 0 by the USB depending on the FIFO buffer status.

For transmitting pipes

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for write access, and are set to 0 when it is not ready. The BRDY interrupt is not generated for the DCP in the transmitting direction even when it is ready for write access.

For receiving pipes

The BRDY interrupt status bits set to 1 when the FIFO buffer is ready for read access, and set to 0 when all data is read (not ready for read access).

When a zero-length packet is received while the FIFO buffer is empty, the associated bit is set to 1 and the BRDY interrupt is continuously generated until the software writes 1 to BCLR. With this setting, the PIPEnBRDY bit cannot be set to 0 by software.

When the SOFCFG.BRDYM bit is set to 1, set the PIPECFG.BFRE bit for all pipes to 0.

Figure 29.11 shows the timing of BRDY interrupt generation.

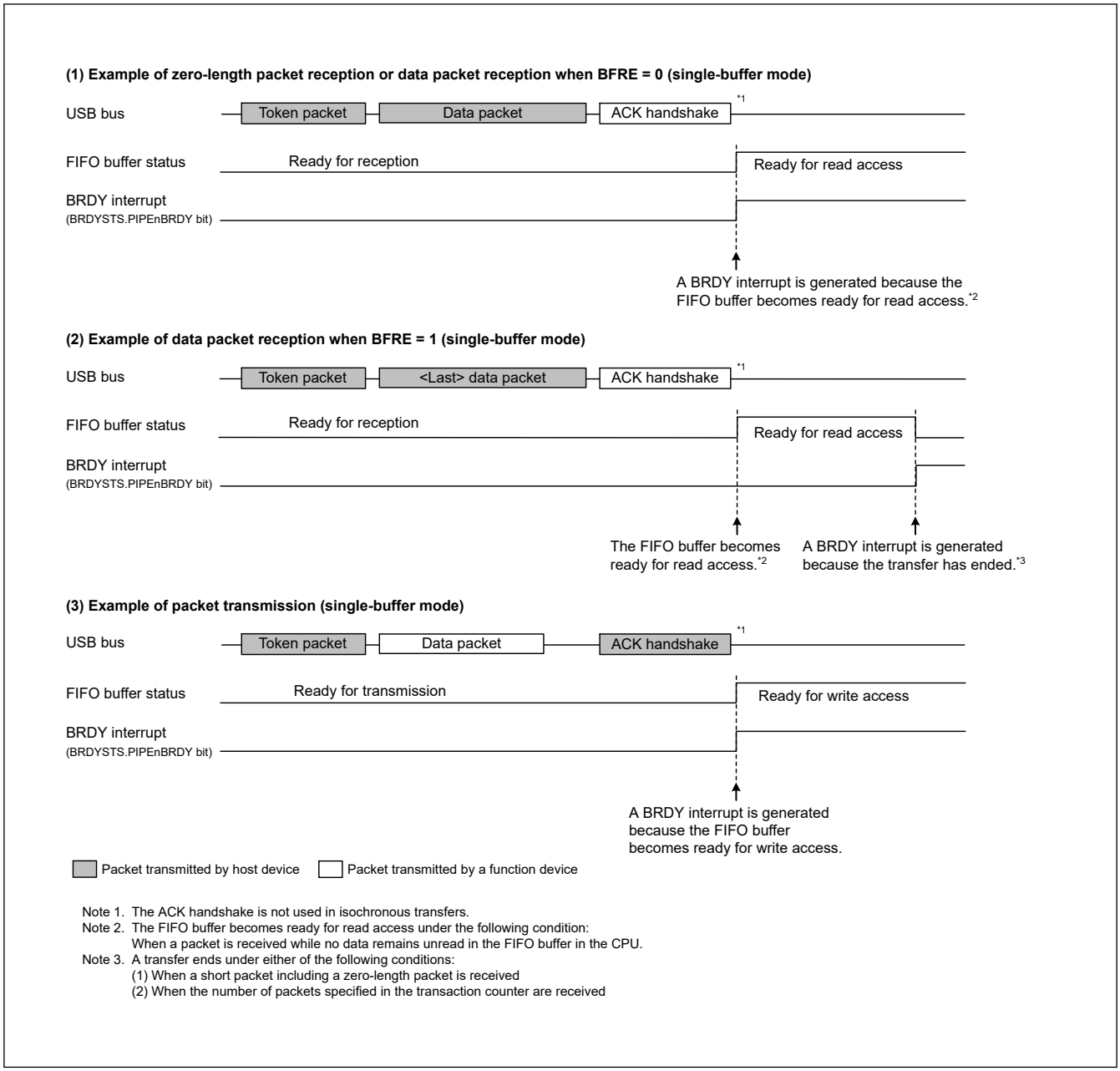


Figure 29.11 Timing of BRDY interrupt generation

The condition for clearing the INTSTS0.BRDY bit depends on the SOFCFG.BRDYM bit setting, as shown in Table 29.18.

Table 29.18 Conditions for clearing the BRDY bit

BRDYM bit	Condition for clearing BRDY bit
0	The USBFS clears the BRDY bit to 0 when all bits in BRDYSTS are set to 0 by software.
1	The USBFS clears the BRDY bit to 0 when the BSTS bits for all pipes have cleared to 0.

29.3.3.2 NRDY interrupt

On generating an internal NRDY interrupt request for the pipe whose PID bits are set to BUF by software, the USBFS sets the associated PIPEnNRDY bit in NRDYSTS to 1. If the associated bit in NRDYENB is set to 1 by software, the USBFS sets the INTSTS0.NRDY bit to 1 and generates a USBFS interrupt.

This section describes the conditions in which the USBFS generates the internal NRDY interrupt request for a given pipe.

The internal NRDY interrupt request is not generated during setup transaction execution in host controller mode. During setup transactions in host controller mode, the SACK or SIGN interrupt is detected.

The internal NRDY interrupt request is not generated during status stage execution of the control transfer in device controller mode.

(1) In host controller mode

For transmitting pipes

On any of the following conditions, the USBFS detects an NRDY interrupt:

- For isochronous transfer pipes, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer. In this case, the USBFS transmits a zero-length packet following the OUT token and sets the associated NRDYSTS.PIPEnNRDY bit and the FRMNUM.OVRN bit to 1.
- During communications other than setup transactions on pipes not used for isochronous transfers, when any combination of the following two cases occur three consecutive times:
  - No response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device
  - An error is detected in the packet from the peripheral device. In this case, the USBFS sets the associated PIPEnNRDY bit to 1 and changes the associated PID[1:0] setting for the pipe to NAK
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device. In this case, the USBFS sets the associated PIPEnNRDY bit to 1 and changes the PID[1:0] setting for the associated pipe to STALL (11b).

For receiving pipes

- For isochronous transfer pipes, when the time to issue an IN token comes but there is no space available in the FIFO buffer. In this case, the USBFS discards the received data for the IN token and sets the PIPEnNRDY bit associated with the pipe and the OVRN bit to 1. When a packet error is detected in the received data for the IN token, the USBFS also sets the FRMNUM.CRCE bit to 1.
- For non-isochronous transfer pipes, when any combination of the following two cases occur three consecutive times:
  - No response is returned from the peripheral device for the IN token issued by the USBFS (when timeout is detected before detection of the DATA packet from the peripheral device)
  - An error is detected in the packet from the peripheral device. In this case, the USBFS sets the associated PIPEnNRDY bit to 1 and changes the associated PID[1:0] setting for the pipe to NAK
- For isochronous transfer pipes, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the DATA packet from the peripheral device) or an error is detected in the packet from the peripheral device. In this case, the USBFS sets the PIPEnNRDY bit associated with the pipe to 1. The PID[1:0] setting for the pipe is not changed.
- For isochronous transfer pipes, when a CRC error or a bit stuffing error is detected in the received data packet. In this case, the USBFS sets the PIPEnNRDY bit associated with the pipe and the CRCE bit to 1.
- When the STALL handshake is received. In this case, the USBFS sets the PIPEnNRDY bit associated with the pipe to 1 and changes the PID[1:0] setting for the associated pipe to STALL.

(2) In device controller mode

For transmitting pipes

- When an IN token is received while there is no data to be transmitted in the FIFO buffer. In this case, the USBFS generates a NRDY interrupt request on reception of the IN token and sets the NRDYSTS.PIPEnNRDY bit to 1. For an isochronous transfer pipe in which an interrupt is generated, the USBFS transmits a zero-length packet and sets the FRMNUM.OVRN bit to 1.

For receiving pipes

- When an OUT token is received but there is no space available in the FIFO buffer. For an isochronous transfer pipe in which an interrupt is generated, the USBFS generates a NRDY interrupt request on reception of the OUT token and sets the PIPEnNRDY bit to 1 and OVRN bit to 1. For a non-isochronous transfer pipe in which an interrupt is generated, the USBFS generates a NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the PIPEnNRDY bit to 1. The NRDY interrupt request is not generated during retransmission

because of a DATA-PID mismatch. In addition, the NRDY interrupt request is not generated if an error occurs in the DATA packet.

- For isochronous transfer pipes, when a token is not received successfully within an interval frame. In this case, the USBFS generates an NRDY interrupt request when the SOF is received, and sets the PIPEnNRDY bit to 1.

Figure 29.12 shows the timing of NRDY interrupt generation in device controller mode.

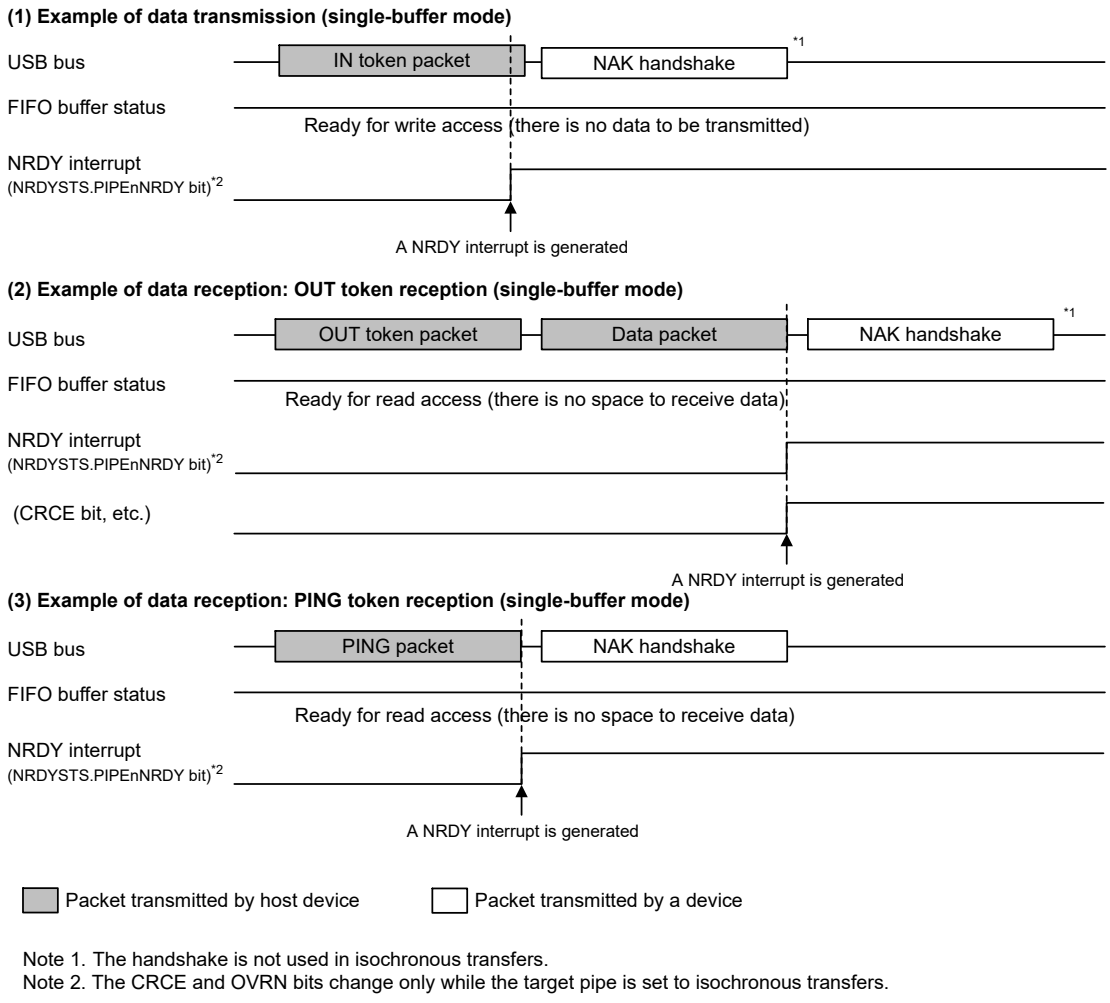


Figure 29.12 Timing of NRDY interrupt generation in device controller mode

29.3.3.3 BEMP interrupt

On detecting a BEMP interrupt for the pipe whose PID bits are set to BUF by software, the USBFS sets the associated BEMPSTS.PIPEnBEMP bit to 1. If the associated bit in BEMPENB is set to 1 by software, the USBFS sets the INTSTS0.BEMP bit to 1 and generates a USBFS interrupt. This section describes the conditions in which the USBFS generates an internal BEMP interrupt request.

(1) For transmitting pipes

When the FIFO buffer of the associated pipe is empty on completion of transmission, including zero-length packet transmission, and in single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for a non-DCP pipe. The internal BEMP interrupt request is not generated in any of the following conditions:

- When the CPU or DMA/DTC has already started writing data to the FIFO buffer of the CPU on completion of transmitting data from one FIFO buffer in double buffer mode
- When the buffer is cleared (emptied) by setting the PIPEnCTR.ACLRM or the BCLR bit to 1 in the port control register
- When an IN transfer (zero-length packet transmission) is performed during the control transfer status stage in device controller mode

(2) For receiving pipes

When a successfully-received data packet size exceeds the specified maximum packet size. In this case, the USBFS generates a BEMP interrupt request, sets the associated BEMPSTS.PIPEnBEMP bit to 1, discards the received data, and changes the associated PID[1:0] setting for the pipe to STALL (11b). The USBFS returns no response in host controller mode, and returns STALL response in device controller mode.

The internal BEMP interrupt request is not generated in any of the following conditions:

- When a CRC error or a bit stuffing error is detected in the received data
- When a setup transaction is being performed:
  - Writing 0 to the BEMPSTS.PIPEnBEMP bit clears the status
  - Writing 1 to the BEMPSTS.PIPEnBEMP bit has no effect

Figure 29.13 shows the timing of BEMP interrupt generation in device controller mode.

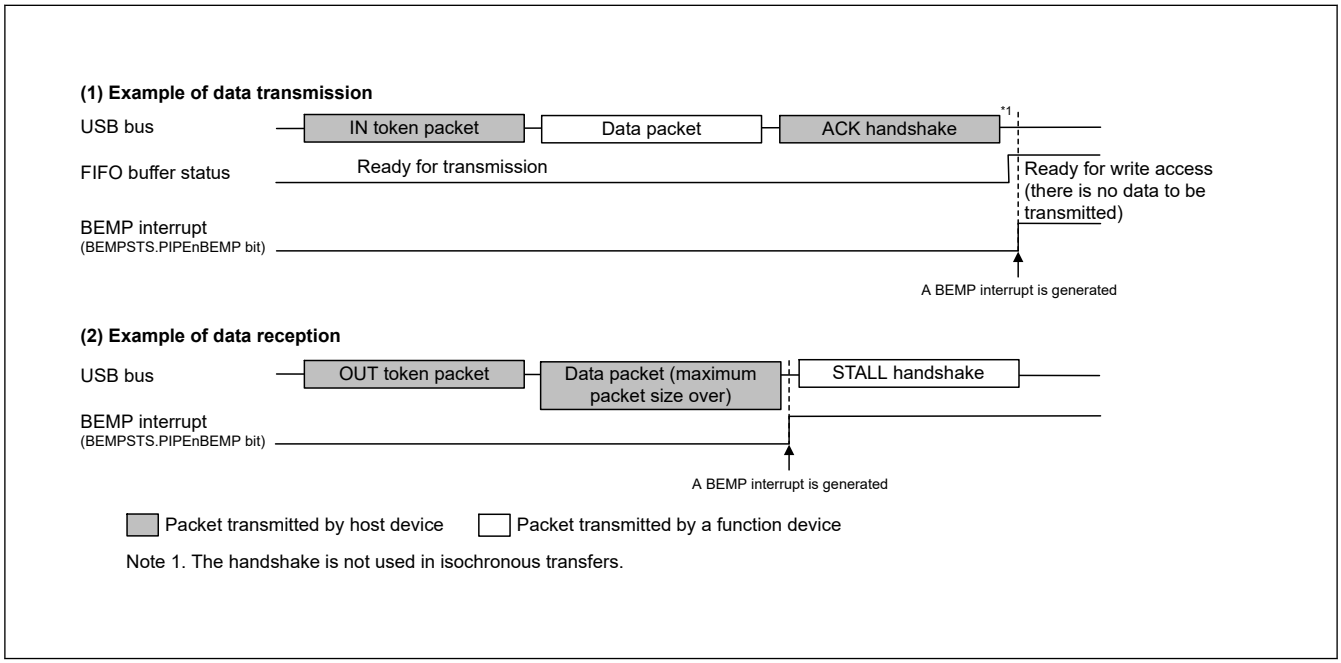


Figure 29.13 Timing of BEMP interrupt generation in device controller mode

29.3.3.4 Device state transition interrupt (device controller mode)

Figure 29.14 shows a diagram of the USBFS device state transitions. The USBFS controls device states and generates device state transition interrupts. However, recovery from the Suspend state (resume signal detection) is detected by means of the resume interrupt. Device state transition interrupts can be enabled or disabled independently in INTENB0. Devices whose states have changed can be checked in the INTSTS0.DVSQ[2:0] bits.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

The USBFS controls device states, and device state transition interrupts can be generated, only in device controller mode.

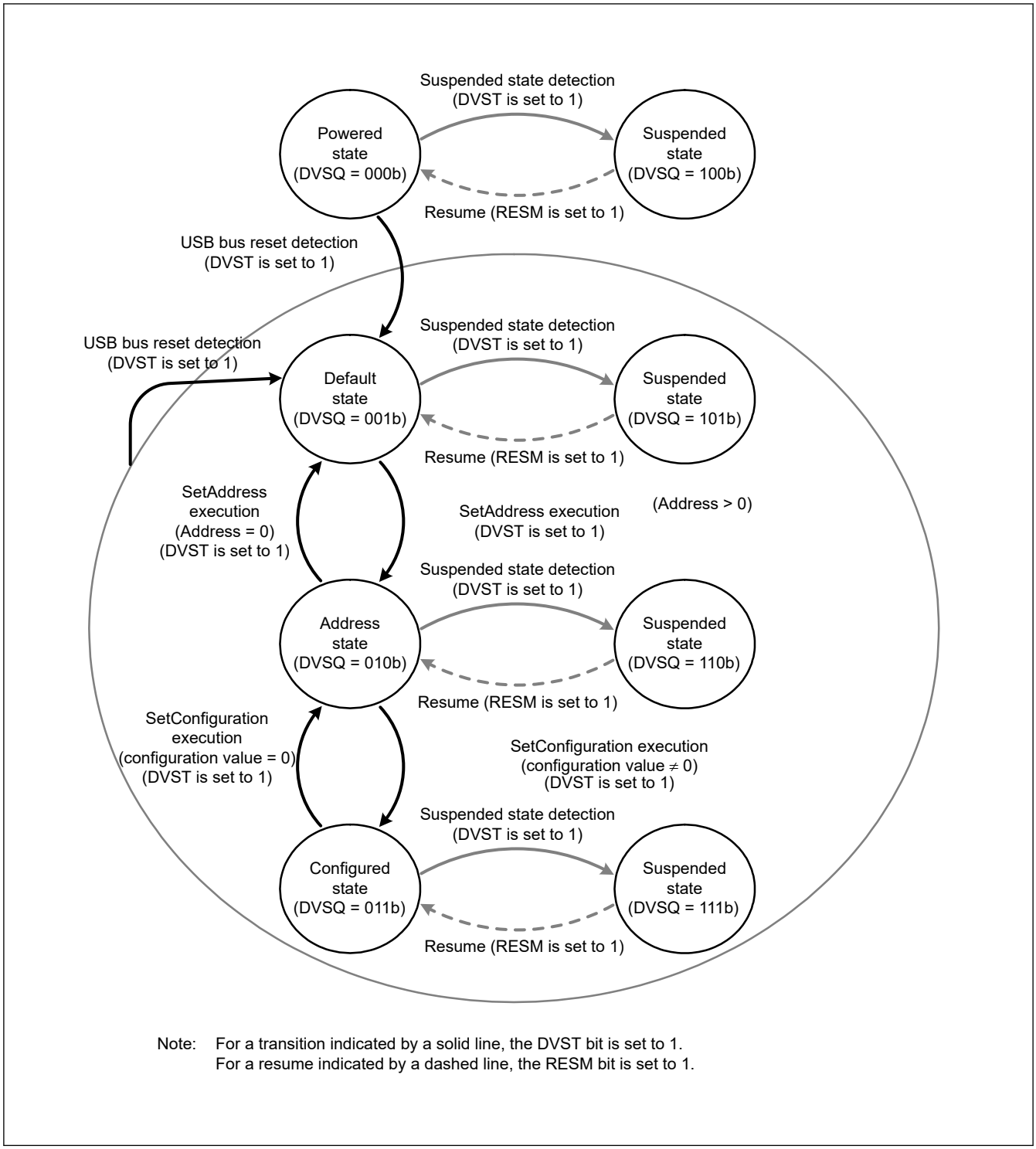


Figure 29.14 Device state transitions

29.3.3.5 Control transfer stage transition interrupt (device controller mode)

Figure 29.15 shows a diagram of the control transfer stage transitions of the USBFS. The USBFS controls the control transfer sequence and generates control transfer stage transition interrupts. Control transfer stage transition interrupts can be enabled or disabled independently in INTENB0. Transfer stages that have transitioned can be checked in the INTSTS0.CTSQ[2:0] bits.

Control transfer stage transition interrupts are generated only in device controller mode. This section describes control transfer sequence errors. If an error occurs, the DCPCTR.PID[1:0] bits are set to 1xb (STALL response).

(1) Control read transfer errors

- An OUT token is received but no data is transferred in response to the IN token at the data stage
- An IN token is received at the status stage
- A data packet with DATAPID = DATA0 is received at the status stage

(2) Control write transfer errors

- An IN token is received but no ACK is returned in response to the OUT token at the data stage
- A data packet with DATAPID = DATA0 is received as the first data packet at the data stage
- An OUT token is received at the status stage

(3) Control write no data transfer errors

- An OUT token is received at the status stage

At the control write transfer data stage, if the receive data length exceeds the wLength value of the USB request, it is not recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (INTSTS0.CTRT = 1), the CTSQ[2:0] = 110b value is saved until the CTRT bit is set to 0, clearing the interrupt status. While CTSQ[2:0] = 110b is being saved, no CTRT interrupt for ending the setup stage is generated, even if a new USB request is received. The USBFS saves the setup stage completion status, and it generates a CTRT interrupt after the interrupt status is cleared by software.

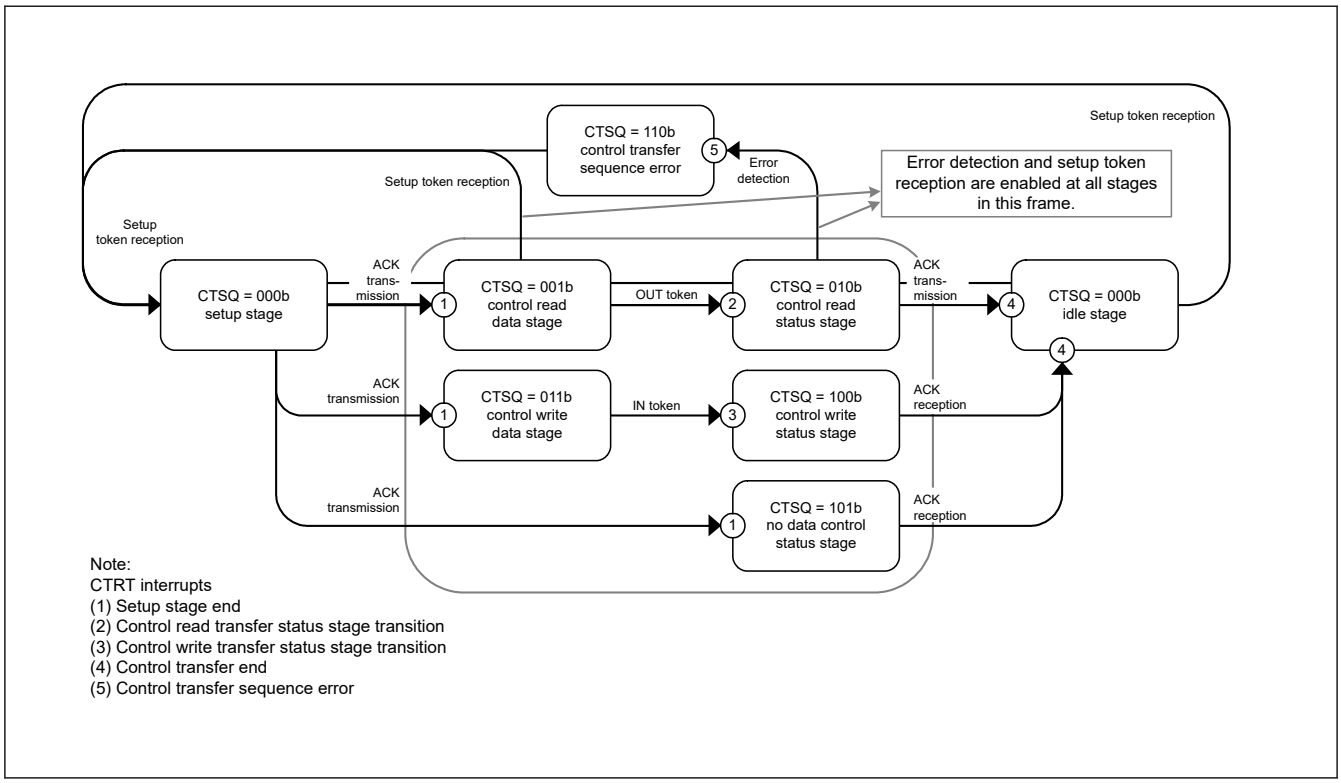


Figure 29.15 Control transfer stage transitions

29.3.3.6 Frame update interrupt

In host controller mode, an interrupt is generated when the frame number is updated.

In device controller mode, an SOFR interrupt is generated when the frame number is updated. The USBFS updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

29.3.3.7 VBUS interrupt

When the USB_VBUS pin level changes, a VBUS interrupt is generated. The level of the USB_VBUS pin can be checked with the INTSTS0.VBSTS bit. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. If the system is activated with the host controller connected, the first VBUS interrupt is not generated, because there is no change in the USB_VBUS pin level.

29.3.3.8 Resume interrupt

In device controller mode, a resume interrupt is generated when the device state is the Suspend state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the Suspend state is detected by means of the resume interrupt.

In host controller mode, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

29.3.3.9 OVRCR interrupt

An OVRCR interrupt is generated when the USB_OVRCURA, USB_OVRCURA-DS, USB_OVRCURB or USB_OVRCURB-DS pin level has changed. The levels of the USB_OVRCURA, USB_OVRCURA-DS, USB_OVRCURB and USB_OVRCURB-DS pins can be checked in the SYSSTS0.OVCMON[1:0] flags. The external power supply IC can check whether overcurrent is detected using the OVRCR interrupt.

For OTG connections, the OVRCR interrupt allows you to check whether a change is detected in the VBUS comparator.

29.3.3.10 BCHG interrupt

A BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether a peripheral device is connected and can also be used to detect a remote wakeup in host controller mode. The BCHG interrupt is generated in both host and device controller modes.

29.3.3.11 DTCH interrupt

A DTCH interrupt occurs when a USB bus disconnect is detected in host controller mode. The USBFS detects bus disconnects in compliance with the USB 2.0 specification.

On interrupt detection, all pipes in which communications are being carried out for the relevant port must be terminated by software. The pipes enter the wait state for a bus connection to the port, waiting for an ATTCH interrupt to occur. Regardless of the value set in the associated interrupt enable bit, the USBFS hardware:

- Sets the DVSTCTR0.UACT bit for the port in which the DTCH interrupt is detected to 0
- Puts the port in which the DTCH interrupt occurred into the idle state

29.3.3.12 SACK interrupt

A SACK interrupt is generated when an ACK response for the transmitted setup packet is received from the peripheral device in host controller mode. The SACK interrupt can be used to confirm that the setup transaction is successfully complete.

29.3.3.13 SIGN interrupt

A SIGN interrupt is generated when an ACK response for the transmitted setup packet is not correctly received from the peripheral device three consecutive times in host controller mode. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

29.3.3.14 ATTCH interrupt

An ATTCH interrupt is generated when J-state or K-state of the full-speed signal level is detected on the USB port for 2.5 µs in host controller mode. To be more specific, an ATTCH interrupt is detected on any of the following conditions:

- When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5 µs
- When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 µs

29.3.3.15 EOFERR interrupt

An EOFERR interrupt occurs when the USBFS detects that communication is not complete at the EOF2 timing defined in the USB 2.0 specification.

On interrupt detection, all pipes in which communications are being carried out for the relevant port must be terminated by software, and the port must be re-enumerated. Regardless of the value set in the associated interrupt enable bit, the USBFS hardware:

- Sets the DVSTCTR0.UACT bit for the port in which the EOFERR interrupt is detected to 0
- Puts the port in which the EOFERR interrupt is generated into the idle state

29.3.4 Pipe Control

Table 29.19 lists the pipe settings for the USBFS. USB data transfer is performed through logical pipes that the software associates with endpoints. The USBFS provides 10 pipes that are used for data transfer. Set up the pipes based on your system specifications.

Table 29.19 Pipe settings

Register name	Bit name	Setting	Notes
DCPCFG PIPECFG	TYPE	Transfer type	Pipes 1 to 9: Settable
	BFRE	BRDY interrupt mode	Pipes 1 to 5: Settable
	DBLB	Double buffer select	Pipes 1 to 5: Settable
	DIR	Transfer direction select	IN or OUT settable
	EPNUM	Endpoint number	Pipes 1 to 9: Settable A value other than 0000b must be set when the pipe is used.
	SHTNAK	Selects disabled state for pipe when transfer ends	Pipes 1 and 2: Settable only for bulk transfers Pipes 3 to 5: Settable
DCPMAXP PIPEMAXP	DEVSEL	Device select	Referenced only in host controller mode.
	MXPS	Maximum packet size	Compliant with the USB 2.0 specification.
PIPEPERI	IFIS	Buffer flush	Pipes 1 and 2: Settable only for isochronous transfers Pipes 3 to 9: Setting disabled
	IITV	Interval counter	Pipes 1 and 2: Settable only for isochronous transfers Pipes 3 to 5: Setting disabled Pipes 6 to 9: Settable only in host controller mode
DCPCTR PIPEnCTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit.
	INBUFM	IN buffer monitor	Available only for pipes 1 to 5.
	SUREQ	Setup request	Settable only for the DCP and controlled in host controller mode
	SUREQCLR	SUREQ clear	Settable only for the DCP and controlled in host controller mode
	ATREPM	Auto response mode	Pipes 1 to 5: Settable only in device controller mode
	ACLRM	Auto buffer clear	Pipes 1 to 9: Settable
	SQCLR	Sequence clear	Clears the data toggle bit
	SQSET	Sequence set	Sets the data toggle bit
	SQMON	Sequence monitor	Monitors the data toggle bit
	PBUSY	Pipe busy status	—
	PID	Response PID	See section 29.3.4.6. Response PID.
PIPEnTRE	TRENB	Transaction counter enable	Pipes 1 to 5: Settable
	TRCLR	Current transaction counter clear	Pipes 1 to 5: Settable
PIPEnTRN	TRNCNT	Transaction counter	Pipes 1 to 5: Settable

29.3.4.1 Pipe control register switching procedures

The following bits in the pipe control registers can be changed only when USB communication is prohibited (PID = NAK).

Do not change the following registers and bits when USB communication is enabled (PID = BUF):

- Bits in DCPCFG and DCPMAXP
- SQCLR and SQSET bits in DCPCTR
- Bits in PIPECFG, PIPEMAXP, and PIPEPERI
- ATREPM, ACLRM, SQCLR, and SQSET bits in PIPEnCTR
- Bits in PIPEnTRE and PIPEnTRN

To set these bits when USB communication is enabled (PID = BUF):

1. A request to change the bits in the pipe control register occurs.
2. Set the PID[1:0] bits associated with the pipe to NAK.
3. Wait until the associated PBUSY bit clears to 0.
4. Set the bits in the pipe control register.

The following bits in the pipe control registers can be changed only when the selected pipe information is not set in the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Do not set the following registers when the CURPIPE[3:0] bits are set:

- Bits in DCPCFG and DCPMAXP
- Bits in PIPECFG, PIPEMAXP and PIPEPERI

To change pipe information, you must set the CURPIPE[3:0] bits in the port select registers to a pipe other than the one to be changed. For the DCP, the buffer must be cleared using the BCLR bit in the Port Control Register after the pipe information is changed.

29.3.4.2 Transfer types

The PIPECFG.TYPE[1:0] bits specify the following transfer types for each pipe:

- DCP: No setting is necessary (fixed at control transfer)
- Pipes 1 and 2: Set to bulk or isochronous transfer
- Pipes 3 to 5: Set to bulk transfer
- Pipes 6 to 9: Set to interrupt transfer

29.3.4.3 Endpoint number

The PIPECFG.EPNUM[3:0] bits are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to 15.

- DCP: No setting is necessary (fixed at endpoint 0)
- Pipes 1 to 9: Select and set the endpoint numbers from 1 to 15 so that the combination of the PIPECFG.DIR and EPNUM[3:0] bits is unique

29.3.4.4 Maximum packet size setting

Specify the maximum packet size for each pipe in the DCPMAXP.MXPS[6:0] and PIPEMAXP.MXPS[9:0] bits. The DCP and pipes 1 to 5 can be set to any of the maximum pipe sizes defined in the USB 2.0 specification. For pipes 6 to 9, the maximum packet size is 64 bytes. Set the maximum packet size as follows before starting a transfer (PID = BUF):

- DCP: Set to 8, 16, 32, or 64
- Pipes 1 to 5: Set to 8, 16, 32, or 64 for bulk transfers
- Pipes 1 and 2: Set between 1 and 256 for isochronous transfers
- Pipes 6 to 9: Set between 1 and 64

29.3.4.5 Transaction counter for pipes 1 to 5 in the receiving direction

When the specified number of transactions is complete in the data packet receiving direction, the USBFS recognizes that the transfer ended. Two transaction counters are provided: one is the PIPEnTRN register, which specifies the number of transactions to be executed, and the other is the current counter, which internally counts the number of executed transactions. If the PIPECFG.SHTNAK bit is set to 1, when the current counter value matches the specified number of transactions, the associated PIPEnCTR.PID[1:0] bits are set to NAK and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the PIPEnTRE.TRCLR bit. The data read from PIPEnTRN differs depending on the PIPEnTRE.TRENB setting as follows:

- The TRENB bit = 0: Specified transaction counter value can be read
- The TRENB bit = 1: Current counter value indicating the internally counted number of executed transactions can be read

The following constraints apply when working with the TRCLR bit:

- If the transactions are being counted and PID = BUF, the current counter cannot be cleared
- If there is any data left in the buffer, the current counter cannot be cleared

29.3.4.6 Response PID

Specify the response PID for each pipe in the PID[1:0] bits in DCPCTR and PIPEnCTR. This section describes the USBFS operation with different response PID settings.

(1) Software response PID settings in host controller mode

Select the response PID to specify the execution of transactions as follows:

- NAK setting: Using pipes is disabled and no transactions are executed
- BUF setting: Transactions are executed based on the FIFO buffer state:
  - OUT direction: An OUT token is issued if the FIFO buffer contains transmit data.
  - IN direction: An IN token is issued if the FIFO buffer is not full and can receive data.
- STALL setting: Using pipes is disabled and no transactions are executed

Note: Use the DCPCTR.SUREQ bit to execute setup transactions for the DCP.

(2) Software response PID settings in device controller mode

Select the response PID to respond as follows to transactions from the host:

- NAK setting: A NAK response is returned to all generated transactions
- BUF setting: A response is returned to transactions based on the FIFO buffer
- STALL setting: A STALL response is returned to all generated transactions

Note: For setup transactions, an ACK response is always returned, regardless of the PID[1:0] bits setting, and the USB request is stored in the register.

Sections (3) and (4) describe situations in which the USBFS writes to the PID[1:0] bits because of specific transaction results.

(3) Hardware response PID settings in host controller mode

- NAK setting: PID = NAK is set in the following cases, and issuing of tokens is automatically stopped:
  - When a non-isochronous transfer is performed and an NRDY interrupt is generated

    (For details, see section 29.3.3.2. NRDY interrupt.)
  - If a short packet is received when the PIPECFG.SHTNAK bit is set to 1 for bulk transfers
  - If transaction counting ends when the SHTNAK bit is set to 1 for bulk transfers
- BUF setting: The USBFS does not write this setting.

- STALL setting: PID = STALL is set in the following cases, and issuing of tokens is automatically stopped:
  - When STALL is received in response to a transmitted token
  - When a received data packet exceeds the maximum packet size

(4) Hardware response PID settings in device controller mode

- NAK setting: PID = NAK is set in the following cases, and a NAK response is returned to transactions:
  - When the setup token is received normally (DCP only)
  - If transaction counting ends or a short packet is received when the PIPECFG.SHTNAK bit is set to 1 for bulk transfers
- BUF setting: There is no BUF writing by the USBFS.
- STALL setting: PID = STALL is set in the following cases, and a STALL response is returned to transactions:
  - When a received data packet exceeds the maximum packet size
  - When a control transfer sequence error is detected (DCP only)

29.3.4.7 Data PID sequence bit

The USBFS automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON bit in DCPCTR and PIPEnCTR. When data is transmitted, the sequence bit toggles on ACK handshake reception. When data is received, the sequence bit toggles on ACK handshake transmission. The SQCLR bit in DCPCTR and the SQSET bit in PIPEnCTR can be used to change the data PID sequence bit.

In device controller mode when control transfers are used, the USBFS automatically sets the sequence bit for stage transitions. DATA1 is returned when the setup stage ends. The sequence bit is not referenced and PID = DATA1 is returned in the status stage. Therefore, no software settings are required. However, in host controller mode when control transfers are used, the sequence bit must be set by software for the stage transitions.

For ClearFeature requests for transmission or reception, the data PID sequence bit must be set by software in both host and device controller modes.

29.3.4.8 Response PID = NAK function

The USBFS provides a function for disabling pipe operation (PID response = NAK) when the final data packet of a transaction is received. The USBFS automatically distinguishes this based on reception of a short packet or the transaction counter. Enable this function by setting the PIPECFG.SHTNAK bit to 1.

When the double buffer mode is being used for the FIFO buffer, using this function enables reception of data packets in transfer units. If pipe operation is disabled, the software must enable the pipe again (PID response = BUF).

The response PID = NAK function can be used only for bulk transfers.

29.3.4.9 Auto response mode

For bulk transfer pipes (1 to 5), when the PIPEnCTR.ATREPM bit is set to 1, a transition is made to auto response mode. During an OUT transfer (PIPECFG.DIR = 0), OUT-NAK mode is invoked, and during an IN transfer (DIR = 1), null auto response mode is invoked.

29.3.4.10 OUT-NAK mode

For bulk OUT transfer pipes, NAK is returned in response to an OUT token, and an NRDY interrupt is output when the PIPEnCTR.ATREPM bit is set to 1. To transition from normal mode to OUT-NAK mode, specify OUT-NAK mode while pipe operation is disabled (PID[1:0] = 00b for NAK response). Next enable pipe operation (PID[1:0] = 01b for BUF response), on which OUT-NAK mode becomes valid. If an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is returned to the host.

To transition from OUT-NAK mode to normal mode, cancel OUT-NAK mode while pipe operation is disabled (NAK). Next enable pipe operation (BUF). In normal mode, reception of OUT data is enabled.

29.3.4.11 Null auto response mode

For bulk IN transfer pipes, zero-length packets are continuously transmitted when the PIPEnCTR.ATREPM bit is set to 1.

To transition from normal mode to null auto response mode, specify null auto response mode while pipe operation is disabled (response PID = NAK). Next enable pipe operation (response PID = BUF) on which null auto response mode becomes valid. Before setting null auto response mode, check that PIPEnCTR.INBUFM = 0, because the mode can be set only when the buffer is empty. If the INBUFM bit is 1, empty the buffer using the PIPEnCTR.ACLRM bit. Do not write data from the FIFO port while a transition to null auto response mode is being made.

To transition from null auto response mode to normal mode, keep pipe operation disabled (response PID = NAK) for the period of the zero-length packet transmission (about 10 µs) before canceling the null auto response mode. In normal mode, data can be written from the FIFO port, so packet transmission to the host is enabled by enabling pipe operation (response PID = BUF).

29.3.5 FIFO Buffer

The USBFS provides a FIFO buffer for data transfers, and it manages the memory area used for each pipe. The FIFO buffer has two states depending on whether the access right is assigned to the system (CPU side) or the USBFS (SIE side).

(1) Buffer status

Table 29.20 and Table 29.21 show the buffer status in the USBFS. The FIFO buffer status can be confirmed using the DCPCTR.BSTS and PIPEnCTR.INBUFM bits. The transfer direction for the FIFO buffer can be specified in either the PIPECFG.DIR or CFIFOSEL.ISEL bit (when DCP is selected).

The INBUFM bit is valid for pipes 1 to 5 in the transmitting direction.

When a transmitting pipe uses double buffering, the software can read the BSTS bit to monitor the FIFO buffer status on the CPU side and the INBUFM bit to monitor the FIFO buffer status on the SIE side. When write access to the FIFO port by the CPU or DMA/DTC is slow and the buffer empty status cannot be determined using the BEMP interrupt, the software can use the INBUFM bit to confirm the end of transmission.

Table 29.20 Buffer status indicated in the BSTS bit

ISEL or DIR	BSTS	FIFO buffer status
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is disabled.
0 (receiving direction)	1	There is received data, or a zero-length packet is received. Reading from the FIFO port is allowed. When a zero-length packet is received, reading is not possible and the buffer must be cleared.
1 (transmitting direction)	0	Transmission has not completed. Writing to the FIFO port is disabled.
1 (transmitting direction)	1	Transmission is complete. CPU write is allowed.

Table 29.21 Buffer status indicated in the INBUFM bit

DIR	INBUFM	FIFO buffer status
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	Transmission is complete. There is no data waiting to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted.

29.3.6 FIFO Buffer Clearing

Table 29.22 shows the methods for clearing the FIFO buffer. The FIFO buffer can be cleared using BCLR bit in the port control register, DnFIFOSEL.DCLRM, or the PIPEnCTR.ACLRM bit.

Single or double buffering can be selected for pipes 1 to 5 in the PIPECFG.DBLB bit.

Table 29.22 Buffer clearing methods

FIFO buffer clearing mode	Clearing FIFO buffer on the CPU side	Mode for automatically clearing the FIFO buffer after reading the specified pipe data	Auto buffer clear mode for discarding all received packets
Register used	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Bit used	BCLR	DCLRM	ACLRM
Clearing condition	Cleared by writing 1	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

(1) Auto buffer clear mode function

The USBFS discards all received data packets if the PIPEnCTR.ACLRM bit is set to 1. If a correct data packet is received, the ACK response is returned to the host controller. The auto buffer clear mode function can only be set in the FIFO buffer reading direction.

Setting the ACLRM bit to 1 and then to 0 clears the FIFO buffer of the selected pipe regardless of the access direction. An access cycle of at least 100 ns is required for the internal hardware sequence processing between ACLRM = 1 and ACLRM = 0.

29.3.7 FIFO Port Functions

Table 29.23 shows the settings for the FIFO port functions. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, set the BVAL flag in the port control register to end writing. To send a zero-length packet, use the BCLR bit to clear the buffer, and then set the BVAL flag to end writing.

In reading, reception of new packets is automatically enabled when all data is read. Data cannot be read when a zero-length packet is received (DTLN[8:0] = 0), so the buffer must be cleared with the BCLR bit. The length of the receive data can be confirmed in the DTLN[8:0] bits in the port control register.

Table 29.23 FIFO port function settings

Register name	Bit name	Description
CFIFOSEL, DnFIFOSEL (n = 0, 1)	RCNT	Selects DTLN[11:0] read mode
	REW	FIFO buffer rewind (re-read, rewrite)
	DCLRM	Automatically clears receive data for a specified pipe after the data is read (only for DnFIFO)
	DREQE	Enables DMA/DTC transfers (only for DnFIFO)
	MBW	FIFO port access bit width
	BIGEND	Selects FIFO port endian
	ISEL	FIFO port access direction (only for DCP)
	CURPIPE	Selects the current pipe
CFIFOCTR, DnFIFOCTR (n = 0, 1)	BVAL	Ends writing to the FIFO buffer
	BCLR	Clears the FIFO buffer on the CPU side
	DTLN	Checks the length of receive data

(1) FIFO port selection

Table 29.24 shows the pipes that can be selected with the different FIFO ports. The pipe to be accessed must be selected in the CURPIPE[3:0] bits in the port select register. After the pipe is selected, the software must check whether the written value can be read correctly from the CURPIPE[3:0] bits. (If the previous pipe number is read, it indicates that the USBFS is modifying the pipe.) Next, the software checks that the FRDY bit in the port control register is 1.

In addition, the software must specify the bus width to be accessed in the MBW bit in the port select register. The FIFO buffer access direction conforms to the PIPECFG.DIR setting. For the DCP only, the ISEL bit in the port select register determines the direction.

Table 29.24 FIFO port access by pipe

Pipe	Access method	Ports that can be used
DCP	CPU access	CFIFO port register
Pipes 1 to 9	CPU access	• CFIFO port register • D0FIFO/D1FIFO port register
	DMA/DTC access	D0FIFO/D1FIFO port register

(2) REW bit

It is possible to temporarily stop access to a pipe currently being accessed, access a different pipe, and then continue processing for the first pipe again. The REW bit in the port select register is used for this processing.

If a pipe is selected in the CURPIPE[3:0] bits in the port select register with the REW bit set to 1, the pointer used for reading from and writing to the FIFO buffer is reset, and reading or writing can be carried out from the first byte. If a pipe is selected with 0 set for the REW bit, data can be read and written in continuation from the previous selection, without the pointer being reset.

To access the FIFO port, the software must check that the FRDY bit in the port control register is 1 after selecting a pipe.

29.3.8 DMA Transfers (D0FIFO and D1FIFO Ports)

(1) Overview of DMA transfers

For pipes 1 to 9, the FIFO port can be accessed using the DMAC. When buffer access for a pipe targeted for DMA transfer is enabled, a DMA transfer request is issued.

Select the unit of transfer to the FIFO port in the DnFIFOSEL.MBW bit, and select the pipe targeted for the DMA transfer in the DnFIFOSEL.CURPIPE[3:0] bits. Do not change the selected pipe during the DMA transfer.

(2) DnFIFO auto clear mode (D0FIFO and D1FIFO port reading direction)

If 1 is set in the DnFIFOSEL.DCLRM bit, the USBFS automatically clears the FIFO buffer of the selected pipe when reading of data from the FIFO buffer is complete.

Table 29.25 shows the packet reception and FIFO buffer clearing processing by software for each of the settings.
As shown in the table, the buffer clearing conditions depend on the value set in the PIPECFG.BFRE bit. Using the DnFIFOSEL.DCLRM bit eliminates the need for the buffer to be cleared by software in any situation that requires buffer clearing. This enables DMA transfers without involving software.

The DnFIFO auto clear mode can only be set in the FIFO buffer reading direction.

Table 29.25 Packet reception and FIFO buffer clearing processing by software

Buffer status when packet is received	Register setting			
	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	No clearing required	No clearing required	No clearing required	No clearing required
Zero-length packet reception	Clearing required	Clearing required	No clearing required	No clearing required
Normal short packet reception	No clearing required	Clearing required	No clearing required	No clearing required
Transaction count end	No clearing required	Clearing required	No clearing required	No clearing required

29.3.9 Control Transfers Using the DCP

The DCP is used for data transfers in the control transfer data stage. The FIFO buffer of the DCP is a 64-byte single buffer with a fixed area for both control reads and control writes. The FIFO buffer can be accessed only through the CFIFO port.

29.3.9.1 Control transfers in host controller mode

(1) Setup stage

The USQREQ, USBVAL, USBINDX, and USBLENG registers are used to transmit USB requests for setup transactions. Writing the setup packet data to the registers and then writing 1 to the DCPCTR.SUREQ bit transmits the specified data for the setup transaction. On completion of the transaction, the SUREQ bit clears to 0. Do not change these USB request registers while SUREQ = 1.

When an attached function device is detected, the software must issue the first setup transaction for the device using this sequence with the DCPMAXP.DEVSEL[3:0] bits cleared to 0 and the DEVADD0.USBSPD[1:0] bits set appropriately.

When an attached function device is shifted to the Address state, the software must issue setup transactions using this sequence with the assigned USB address set in the DEVSEL[3:0] bits and the bits in DEVADDn corresponding to the specified USB address set appropriately. For example, when PIPEMAXP.DEVSEL[3:0] = 0010b, make appropriate settings in DEVADD2. When PIPEMAXP.DEVSEL[3:0] = 0101b, make appropriate settings in DEVADD5.

When the setup transaction data is sent, an interrupt request is generated based on the response from the peripheral device (SIGN or SACK bit in INTSTS1). This interrupt request allows the software to check the setup transaction result.

A DATA0 data packet (USB request) for the setup transaction is always transmitted regardless of the status of the DCPCTR.SQMON bit.

(2) Data stage

The data stage is used to transfer data using the DCP FIFO buffer.

Before accessing the DCP FIFO buffer, specify the access direction in the CFIFOSEL.ISEL bit. Specify the transfer direction in the DCPCFG.DIR bit.

For the first data packet of the data stage, the data PID must be transferred as DATA1. Set data PID = DATA1 in the DCPCTR.SQSET bit and set the PID bits = BUF. Completion of data transfer is detected using the BRDY or BEMP interrupt.

For control write transfers, when the number of data bytes to be sent is an integer multiple of the maximum packet size, the software must send a zero-length packet at the end.

(3) Status stage

The status stage is used for zero-length packet data transfers in the reverse direction of the data stage. As in the data stage, data is transferred using the DCP FIFO buffer. Transactions are executed using the same procedure as the data stage.

Data packets in the status stage must be transmitted and received with the data PID set to DATA1 using the DCPCTR.SQSET bit.

When a zero-length packet is received, check the receive-data length in the CFIFOCTR.DTLN[8:0] bits after a BRDY interrupt is generated, and then clear the FIFO buffer using the BCLR bit.

29.3.9.2 Control transfers in device controller mode

(1) Setup stage

The USBFS sends an ACK response to a normal setup packet for the USBFS. The USBFS operates in the setup stage as follows:

On receiving a new setup packet, the USBFS sets the following bits:

- Sets the INTSTS0.VALID bit to 1
- Sets the DCPCTR.PID[1:0] bits to NAK
- Sets the DCPCTR.CCPL bit to 0

When the USBFS receives a data packet following a setup packet, it stores the USB request parameters in USBREQ, USBVAL, USBINDX, and USBLENG.

Before performing the response processing for a control transfer, set the VALID flag to 0. When the VALID bit = 1, PID = BUF cannot be set, and the data stage cannot be terminated.

Using the VALID bit function, the USBFS can suspend a request being processed when it receives a new USB request during a control transfer and return a response to the latest request.

In addition, the USBFS automatically detects the direction bit (bmRequestType bit [8]) and the request data length (wLength) in the received USB request. It distinguishes between control read transfers, control write transfers, and no-data control transfers, and it controls stage transitions. For an incorrect sequence, a sequence error occurs in the control transfer stage transition interrupt, and the interrupt is reported to the software. For a diagram of the stage control by the USBFS, see Figure 29.15.

(2) Data stage

The DCP must be used to execute data transfers for received USB requests. Before accessing the DCP FIFO buffer, specify the access direction in the CFIFOSEL.ISEL bit.

If the transfer data is larger than the size of the DCP FIFO buffer, execute the data transfer using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

(3) Status stage

Control transfers are terminated by setting the DCPCTR.CCPL bit to 1 while the DCPCTR.PID[1:0] bits are set to BUF.

After this setting is made, the USBFS automatically executes the status stage based on the data transfer direction determined at the setup stage. The procedure is as follows:

- For control read transfers

  The USBFS receives a zero-length packet from the USB host and transmits an ACK response.
- For control write transfers and no-data control transfers

  The USBFS transmits a zero-length packet and receives an ACK response from the USB host.

(4) Control transfer auto response function

The USBFS automatically responds to a correct SET_ADDRESS request. If any of the following errors occurs in the SET_ADDRESS request, a response from the software is necessary.

- bmRequestType is not 0x00: Any transfer other than a control write transfer
- wIndex is not 0x00: Request error
- wLength is not 0x00: Any transfer other than a no-data control transfer
- wValue is larger than 0x7F: Request error
- INTSTS0.DVSQ[2:0] are 011b (Configured state): Control transfer of a device state error

For all requests other than the SET_ADDRESS request, a response is required from the corresponding software.

29.3.10 Bulk Transfers (Pipes 1 to 5)

The FIFO buffer usage (single/double buffer setting) is configurable for bulk transfers. The USBFS provides the following functions for bulk transfers:

- BRDY interrupt function (PIPECFG.BFRE bit), see section 29.3.3.1. BRDY interrupt
- Transaction count function (PIPEnTRE.TRENB, TRCLR, and PIPEnTRN.TRNCNT[15:0] bits), see section 29.3.4.5. Transaction counter for pipes 1 to 5 in the receiving direction
- Response PID = NAK function (PIPECFG.SHTNAK bit), see section 29.3.4.8. Response PID = NAK function
- Auto response mode (PIPEnCTR.ATREPM bit), see section 29.3.4.9. Auto response mode

29.3.11 Interrupt Transfers (Pipes 6 to 9)

In device controller mode, the USBFS performs interrupt transfers based on the timing dictated by the host controller.

In host controller mode, the software can set the timing for issuing tokens using the interval counter.

29.3.11.1 Interval counter for interrupt transfers in host controller mode

Specify the transaction interval for interrupt transfers in the PIPEPERI.IITV[2:0] bits. The USBFS issues interrupt transfer tokens based on this interval.

(1) Counter initialization

The USBFS initializes the interval counter under the following conditions:

- Power-on reset

  This initializes the IITV[2:0] bits.
- FIFO buffer initialization using the PIPEnCTR.ACLRM bit:

  This does not initialize the IITV[2:0] bits, but does initialize the count value. Setting the PIPEnCTR.ACLRM bit to 0 starts counting from the value set in IITV[2:0].

The interval counter is not initialized in the following case:

- USB bus reset or USB suspended

  The IITV[2:0] bits are not initialized. Setting 1 to the DVSTCTR0.UACT bit starts counting from the value saved before entering the USB bus reset state or USB suspend state.

(2) Operation when tokens cannot be transmitted or received even on token generation

No token is generated in the following cases even at token generation time. In these cases, the USBFS tries to execute the transaction in the next interval.

- When the PID is set to NAK or STALL
- When the FIFO buffer is full at token transmit time in the receiving (IN) direction
- When there is no data to be transmitted in the FIFO buffer at token transmit time in the transmitting (OUT) direction

29.3.12 Isochronous Transfers (Pipes 1 and 2)

The USBFS provides the following functions for isochronous transfers:

- Notification of isochronous transfer error
- Interval counter specified in the PIPEPERI.IITV[2:0] bits
- Isochronous IN transfer data setup control (IDLY function)
- Isochronous IN transfer buffer flush function specified in the PIPEPERI.IFIS bit

29.3.12.1 Error detection in isochronous transfers

The USBFS provides a function for detecting the errors described in this section, so that when errors occur in isochronous transfers, they can be controlled by software. Table 29.26 and Table 29.27 show the priority order for errors detected by the USBFS and the associated interrupts.

PID errors

- The PID value of the received packet is invalid.

CRC errors and bit stuffing errors

- A CRC error is found in a received packet or the bit stuffing is illegal.

Maximum packet size exceeded

- The data size of the received packet exceeds the specified maximum packet size.

Overrun and underrun errors

In host controller mode:

- The FIFO buffer is full at token transmit time in the IN (receiving) direction
- There is no data to be sent in the FIFO buffer at token transmit time in the OUT (transmitting) direction

In device controller mode:

- There is no data to be sent in the FIFO buffer at token receive time in the IN (transmitting) direction
- The FIFO buffer is full at token receive time in the OUT (receiving) direction

Interval errors

In device controller mode, the following cases are treated as an interval error:

- Failure to receive an IN token in the interval frame during an isochronous IN transfer
- Failure to receive an OUT token in the interval frame during an isochronous OUT transfer

Table 29.26 Error detection for token transmission and reception

Detection priority	Error	Generated interrupt and status
1	PID error	No interrupts are generated in either host or device controller mode (ignored as a corrupted packet)
2	CRC or bit stuffing error	No interrupts are generated in either host or device controller mode (ignored as a corrupted packet)
3	Overrun or underrun error	An NRDY interrupt is generated to set the FRMNUM.OVRN bit to 1 in both host and device controller modes. In device controller mode, a zero-length packet is transmitted in response to an IN token. No data packets are received in response to OUT token.
4	Interval error	An NRDY interrupt is generated in device controller mode. No interrupt is generated in host controller mode.

Table 29.27 Error detection for data packet reception

Detection priority	Error	Generated interrupt and status
1	PID error	No interrupts are generated (ignored as a corrupted packet)
2	CRC or bit stuffing error	An NRDY interrupt is generated and the FRMNUM.CRCE bit sets to 1 in both host and device controller modes
3	Maximum packet size exceeded error	A BEMP interrupt is generated and the PID[1:0] bits set to STALL in both host and device controller modes

29.3.12.2 DATA-PID

In device controller mode, the USBFS responds to a received PID as follows:

(1) IN direction

- DATA0: Transmitted as data packet PID
- DATA1: Not transmitted
- DATA2: Not transmitted
- mDATA: Not transmitted

(2) OUT direction

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Packets ignored
- mDATA: Packets ignored

29.3.12.3 Interval counter

The isochronous transfer interval can be set in the PIPEPERI.IITV[2:0] bits. In device controller mode, the interval counter enables the functions as shown in Table 29.28. In host controller mode, the USBFS generates the token issuance timing, and the interval counter operation is the same as that for interrupt transfers.

Table 29.28 Interval counter functions in device controller mode

Transfer direction	Function	Conditions for detection
IN	Transmit buffer flush	Failure to receive an IN token successfully in the interval frame during an isochronous IN transfer.
OUT	Notification of no reception of token	Failure to receive an OUT token successfully in the interval frame during an isochronous OUT transfer.

The interval count is performed when an SOF is received or for interpolated SOFs, so the isochronism can be maintained even if an SOF is corrupt. The frame interval can be set to 2^{IITV} frames.

(1) Counter initialization in device controller mode

The USBFS initializes the interval counter under the following conditions:

- Power-on reset

  This initializes the PIPEPERI.IITV[2:0] bits.
- FIFO buffer initialization using the ACLRM bit

  This does not initialize the IITV[2:0] bits, but does initialize the count value.

After the interval counter is initialized, the interval count starts under one of the following conditions when a packet is transferred successfully:

- An SOF is received after data is transmitted in response to an IN token when PID = BUF
- An SOF is received after data is received in response to an OUT token when PID = BUF

The interval counter is not initialized in the following conditions:

- When the PID[1:0] bits are set to NAK or STALL

  This does not stop the interval timer. The USBFS attempts the transaction in the next interval.
- When the USB bus is reset or USBFS is suspended

  This does not initialize the IITV[2:0] bits. When an SOF is received, the interval counter starts counting from the value set before SOF was received.

(2) Interval counting and transfer control in host controller mode

The USBFS controls the interval between token issuance operations based on the PIPEPERI.IITV[2:0] bits settings. Specifically, the USBFS issues a token for a selected pipe once every 2^{IITV} frames.

The USBFS starts counting the token issuance interval at the frame following the frame in which the PID[1:0] bits are set to BUF by software.

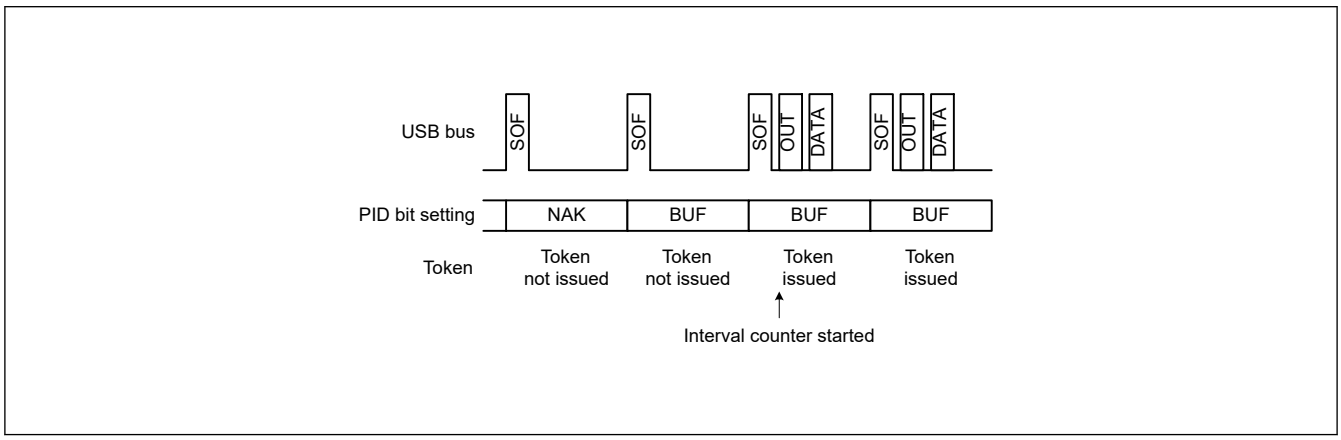


Figure 29.16 Token issuance when IITV = 0

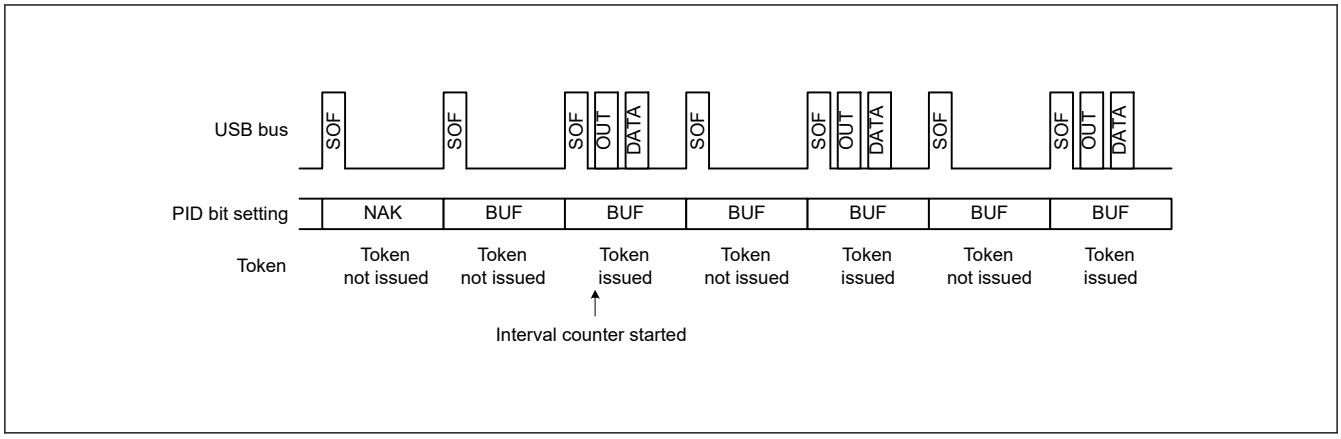


Figure 29.17 Token issuance when IITV = 1

When the selected pipe is set for isochronous transfers, the USBFS performs the following operation in addition to controlling the token issuance interval. The USBFS issues a token even when the NRDY interrupt generation condition is satisfied.

When the selected pipe is for isochronous IN transfers

The USBFS generates an NRDY interrupt when the USBFS issues an IN token but does not successfully receive a packet from a peripheral device (no response or packet error).

The USBFS sets the FRMNUM.OVRN bit to 1, generating an NRDY interrupt, when the time to issue an IN token occurs while the USBFS cannot receive data because the FIFO buffer is full, because the CPU or DMAC/DTC is too slow in reading data from the FIFO buffer.

When the selected pipe is for isochronous OUT transfers

The USBFS sets the OVRN bit to 1, generating an NRDY interrupt and transmitting a zero-length packet, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer, or because the CPU or DMAC/DTC is too slow in writing data to the FIFO buffer.

The token issuance interval is reset on any of the following conditions:

- When the USBFS is reset through a reset pin

  This initializes the IITV[2:0] bits.
- When the PIPEnCTR.ACLRM bit is set to 1 by software

(3) Interval counting and transfer control in device controller mode

When the selected pipe is for isochronous OUT transfers

The USBFS generates an NRDY interrupt when it fails to receive a data packet within the interval set in the PIPEPERI.IITV[2:0] bits.

The USBFS also generates an NRDY interrupt when it fails to receive data because of a CRC error or other errors contained in the data packet or because the FIFO buffer is full.

The NRDY interrupt is generated on SOF packet reception. Even if the SOF packet is corrupted, internal interpolation allows the interrupt to be generated when the SOF packet is received. However, when the IITV bits are set to a value other than 0, the USBFS generates an NRDY interrupt on receiving an SOF packet for every interval after interval counting starts.

When the PID[1:0] bits are set to NAK by software after starting the interval timer, the USBFS does not generate an NRDY interrupt on receiving an SOF packet.

The timing for starting interval counting depend on the IITV[2:0] setting as follows:

- When the IITV[2:0] bits = 0:

  Interval counting starts at the next frame after the software changes the PID[1:0] bits of the selected pipe to BUF.
- When the IITV[2:0] bits ≠ 0:

  Interval counting starts on completion of successful reception of the first data packet after the PID[1:0] bits for the selected pipe are changed to BUF.

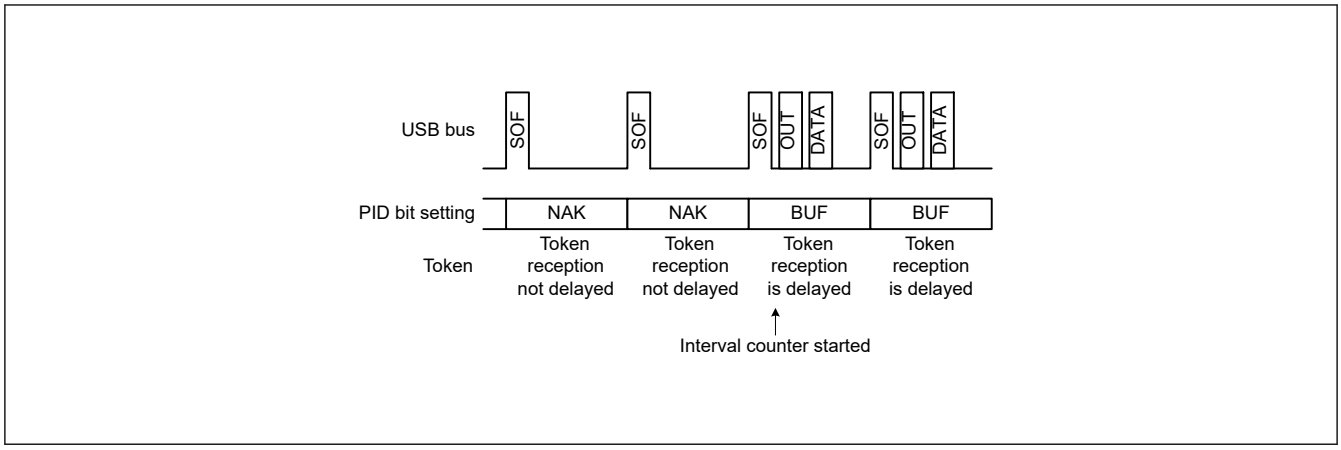


Figure 29.18 Relationship between frames and expected token reception when IITV[2:0] = 0

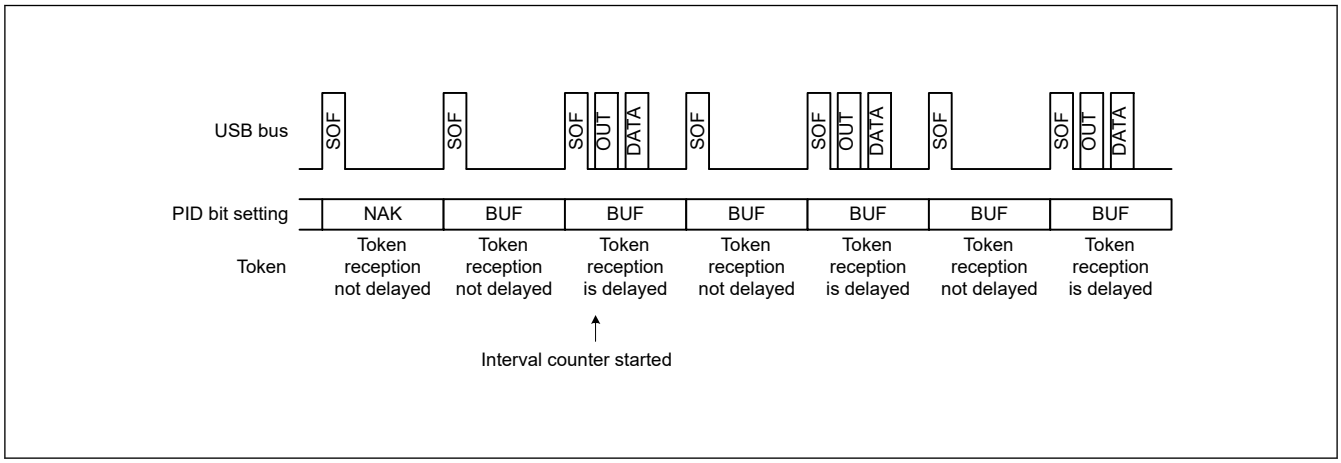


Figure 29.19 Relationship between frames and expected token reception when IITV[2:0] ≠ 0

When the selected pipe is for isochronous IN transfers

The PIPEPERI.IFIS bit must be 1 for this use case. When IFIS = 0, the USBFS transmits a data packet in response to a received IN token regardless of the PIPEPERI.IITV[2:0] setting.

When IFIS is 1 and there is data to be transmitted in the FIFO buffer, the USBFS clears the FIFO buffer when it fails to receive an IN token in the frame at the interval set in the IITV[2:0] bits.

The USBFS also clears the FIFO buffer when it fails to receive an IN token successfully because of a bus error, such as a CRC error, contained in the IN token.

The FIFO buffer is cleared on SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the FIFO buffer to be cleared when the SOF packet is received.

The timing to start interval counting depends on the IITV[2:0] setting, as with OUT transfers.

The interval is counted on any of the following conditions in device controller mode:

- When a hardware reset is applied to the USBFS (which also sets the IITV[2:0] bits to 000b)
- When the PIPEnCTR.ACLRM bit is set to 1 by software
- When the USBFS detects a USB bus reset

(4) Transmit data setup for isochronous transfers in device controller mode

With isochronous data transmission using the USBFS in device controller mode, after data is written to the FIFO buffer, a data packet can be transmitted in the first frame after the SOF packet is detected. This isochronous transfer transmit data setup function can identify the frame that started transmission.

When the double buffering is used, transmission is only enabled for the buffer where data writing was completed first, even after the data write to both buffers is complete. Accordingly, even if multiple IN tokens are received, only the one packet of FIFO buffer data is transmitted.

When the FIFO buffer is ready to transmit data when an IN token is received, the data is transferred and a normal response is returned. However, if the FIFO buffer cannot transmit data, a zero-length packet is transmitted and an underrun error occurs.

Figure 29.20 shows an example transmission using the isochronous transfer transmission data setup function when IITV = 0 (every frame) is set.

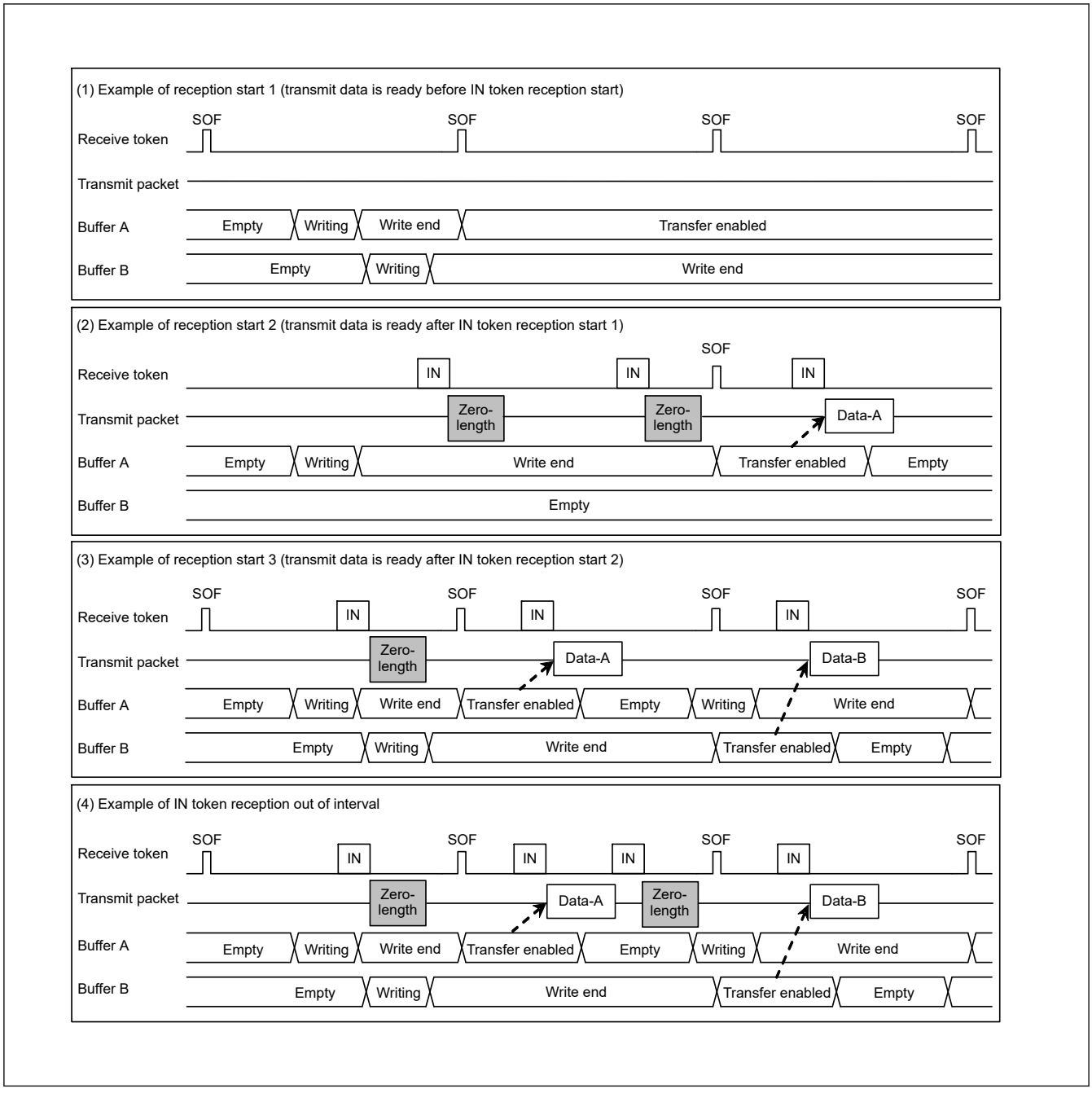


Figure 29.20 Example data setup operation

(5) Transmit buffer flush for isochronous transfers in device controller mode

In device controller mode during isochronous data transmission, if the USBFS receives an SOF packet for the next frame without receiving an IN token in the interval frame, it operates as if the IN token is corrupt and clears the buffer that is enabled for transmission, putting that buffer in the writing enabled state.

When double buffering is used and writing to both buffers is complete, the cleared FIFO buffer is assumed to be the one where the data was transmitted in the interval frame, and transmission is enabled for the FIFO buffer that was not cleared on SOF packet reception.

The timing of the buffer flush function depends on the PIPEPERI.IITV[2:0] setting as follows:

- When IITV = 0:

  The buffer flush operation starts from the first frame after the pipe is enabled.

- When IITV ≠ 0:

  The buffer flush operation starts after the first normal transaction.

Figure 29.21 shows an example buffer flush. When an unanticipated token is received before the interval frame, the USBFS sends the write data or a zero-length packet as an underrun error, depending on the data setup status.

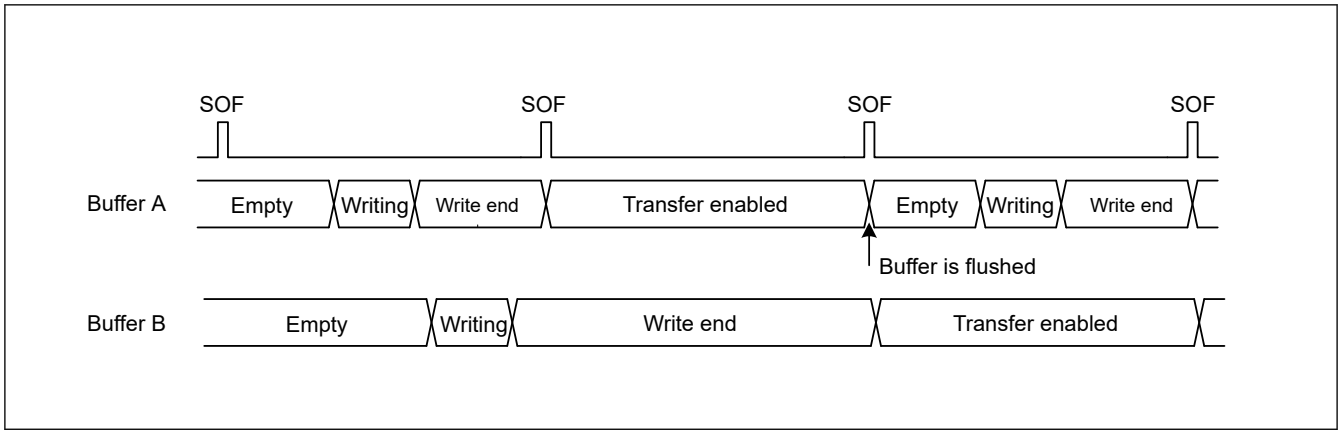


Figure 29.21 Example buffer flush operation

Figure 29.22 shows an example interval error occurrence. There are five types of interval errors, as shown in the figure. An interval error occurs at timing (A), and the buffer flush function is activated.

If an interval error occurs during an IN transfer, the buffer flush function is activated. If it occurs during an OUT transfer, an NRDY interrupt is generated. Use the FRMNUM.OVRN bit to distinguish between this and NRDY interrupts triggered by received packet errors and overrun errors.

For tokens that are shaded in the figure, responses are returned based on the FIFO buffer status.

- IN direction:
  - If the buffer is ready to transfer data, the data is transferred and a normal response is returned
  - If the buffer is not ready to transfer data, a zero-length packet is transmitted and an underrun error occurs
- OUT direction:
  - If the buffer is ready to receive data, the data is received and a normal response is returned
  - If the buffer is not ready to receive data, the received data is discarded and an overrun error occurs

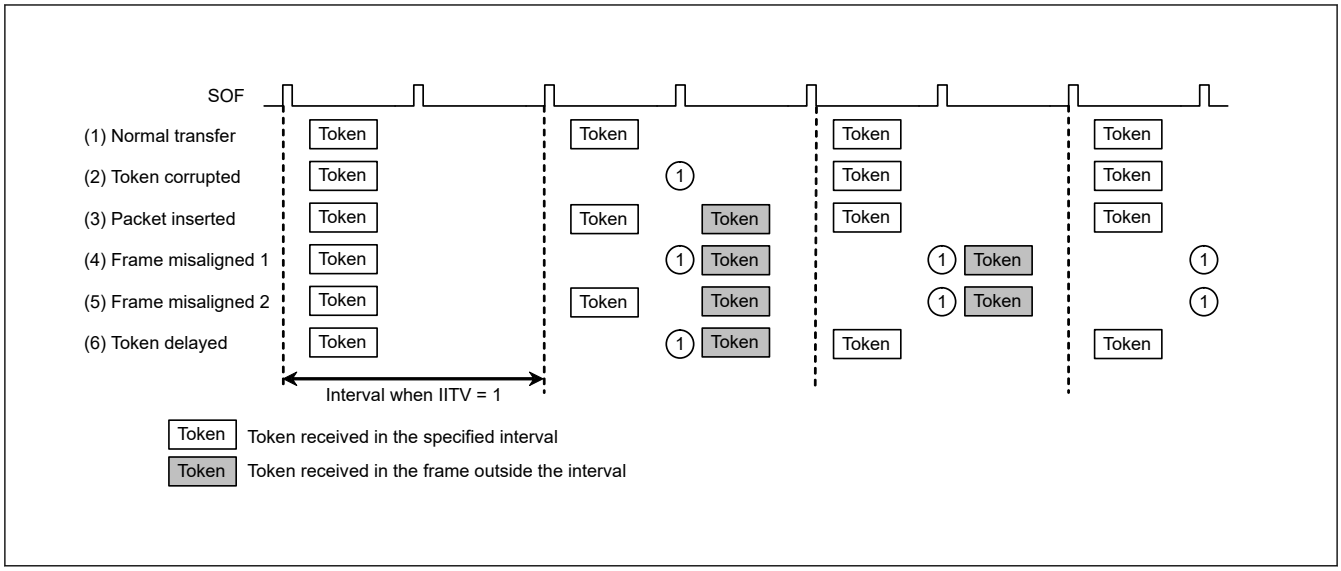


Figure 29.22 Example interval error occurrence when IITV = 1

29.3.13 SOF Interpolation Function

In device controller mode, if packet reception is disabled at intervals of 1 ms because the SOF packet is corrupted or missing, the USBFS interpolates the SOF. SOF interpolation begins when the USBE and SCKE bits in SYSCFG are set to 1 and an SOF packet is received.

The interpolation function is initialized under the following conditions:

- MCU reset
- USB bus reset
- Suspend state detection

The SOF interpolation operates as follows:

- The interpolation function is not activated until an SOF packet is received
- When the first SOF packet is received, interpolation is performed by counting 1 ms on the 48-MHz internal clock
- When the second and subsequent SOF packets are received, interpolation is performed at the previous reception interval
- Interpolation is not performed in the Suspend state or on reception of a USB bus reset

The USBFS supports the following functions controlled by SOF packet reception. These functions operate normally with SOF interpolation if the SOF packet is missing:

- Updating of the frame number
- SOFR interrupt timing
- Isochronous transfer interval count

If an SOF packet is missing during full-speed operation, the FRMNUM.FRNM[10:0] bits are not updated.

29.3.14 Pipe Schedule

29.3.14.1 Conditions for generating transactions

In host controller mode and when the DVSTCTR0.UACT bit is set to 1, the USBFS generates transactions under the conditions shown in Table 29.29.

Table 29.29 Conditions for generating transactions

Transaction	Conditions for generation				
	DIR	PID	IITV0	Buffer state	SUREQ
Setup	—*1	—*1	—*1	—*1	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	Invalid	Receive area exists	—*1
	OUT	BUF	Invalid	Transmit data exists	—*1
Interrupt transfer	IN	BUF	Valid	Receive area exists	—*1
	OUT	BUF	Valid	Transmit data exists	—*1
Isochronous transfer	IN	BUF	Valid	*2	—*1
	OUT	BUF	Valid	*3	—*1

Note 1. An em dash (—) in the table indicates that the condition is unrelated to the generating of tokens. “Valid” indicates that, for interrupt transfers and isochronous transfers, a transaction is generated only in transfer frames that are based on the interval counter. “Invalid” indicates that a transaction is generated regardless of the interval counter.
Note 2. This indicates that a transaction is generated regardless of whether there is a receive area. If there is no receive area, however, the received data is discarded.
Note 3. This indicates that a transaction is generated regardless of whether there is any data to be transmitted. If there is no data to be transmitted, however, a zero-length packet is transmitted.

29.3.14.2 Transfer schedule

This section describes the transfer scheduling within a frame of the USBFS. After the USBFS sends an SOF, the transfer is carried out in the following sequence:

1. Execution of periodic transfers:

   A pipe is searched for in the order of pipe 1 → pipe 2 → pipe 6 → pipe 7 → pipe 8 → pipe 9, and then if there is a pipe for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.

2. Setup transactions for control transfers:

   The DCP is checked, and if a setup transaction is possible, it is sent.

3. Execution of bulk transfers, control transfer data stages, and control transfer status stages:

   A pipe is searched for in the order of DCP → pipe 1 → pipe 2 → pipe 3 → pipe 4 → pipe 5, and then if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated.

   When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

29.3.14.3 Enabling USB communication

Setting the DVSTCTR0.UACT bit to 1 initiates an SOF transmission, and transaction generation is enabled. Setting the UACT bit to 0 stops SOF transmission and the Suspend state is invoked. If the UACT setting is changed from 1 to 0, processing stops after the next SOF is sent.

29.4 Usage Notes

29.4.1 Settings for the Module-Stop State

USBFS operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The USBFS is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

29.4.2 Clearing the Interrupt Status Register on Canceling Software Standby Mode

Because the input buffer is always enabled in Software Standby mode, an unexpected interrupt might occur under the following conditions:

- When the interrupt is enabled in Normal mode
- When the interrupt is disabled in Software Standby mode
- When the input level of the pin that cancels software standby mode is changed in Software Standby mode

These conditions might cause the associated interrupt flag in the Interrupt Status Register to set unexpectedly. After the MCU cancels the Software Standby mode, the unexpected interrupt might be sent to the interrupt controller. To avoid this, always clear the INTSTS0 and INTSTS1 registers in the canceling sequence.

29.4.3 Clearing the Interrupt Status Register after Setting Up the Port Function

The input buffer is disabled before the PmnPFS.PSEL and PmnPFS.PMR port is set up, so the internal signal is fixed high or low. The input buffer is enabled after the port is set so that the external pin state is propagated to the MCU. An unexpected interrupt might occur at this time, causing the VBINT and OVRCR bits in INTSTS0 and INTSTS1, or other interrupt status flags to set to 1. To avoid a malfunction, always clear the INTSTS0 and INTSTS1 registers after setting up the port.

29.4.4 Setting Up the Port Function before setting USB function

USB_DP, USB_DM pins are compatible with P814, P815 I/O ports. So before setting USBFS registers to enable USBFS function, set register bit P814PFS.PMR and P815PFS.PMR to 1.

30. USB 2.0 High-Speed Module (USBHS)

30.1 Overview

The MCU provides a USB 2.0 High-Speed Module (USBHS) that operates as a host or a device controller compliant with the Universal Serial Bus (USB) Specification revision 2.0. The host controller supports USB 2.0 high-speed, full-speed, and low-speed transfers, and the device controller supports USB 2.0 high-speed and full-speed transfers. The USBHS has an internal USB transceiver and supports all of the transfer types defined in the USB 2.0 specification.

The USBHS has FIFO buffer for data transfers, providing a maximum of 10 pipes. Any endpoint number can be assigned to pipes 1 to 9, based on the peripheral devices or the communication requirements for your system.

Table 30.1 lists the USBHS specifications, Figure 30.1 shows a block diagram, and Table 30.2 lists the I/O pins.

Table 30.1 USBHS specifications

Parameter	Specifications
Features	• USB Device Controller (UDC) and USB 2.0 transceiver supporting host controller, device controller, and On-The-Go (OTG) functions • Software can switch between host and device controller modes • Self-power and bus power mode can be used
	Host controller features: • High-speed transfer (480 Mbps), full-speed transfer (12 Mbps), and low-speed transfer (1.5 Mbps) • Automatic scheduling for SOF and packet transmissions • Programmable intervals for isochronous and interrupt transfers • Communications with multiple peripheral devices connected through a single hub
	Device controller features: • High-speed transfer (480 Mbps) and full-speed transfer (12 Mbps) • Control transfer stage control function • Device state control function • Auto response function for SET_ADDRESS request • SOF complementation
Supported transfer types	• Control transfer • Bulk transfer • Interrupt transfer • Isochronous transfer
Pipe configuration	• FIFO buffer of up to 8.5 KB for USB communications • Up to 10 pipes selectable, including the default control pipe • Programmable pipe configurations • Pipes 1 to 9 assignable to any endpoint number
	Transfer conditions specifiable for each pipe: • Pipe 0: Control transfer with 64-byte single buffer • Pipes 1 and 2: Bulk isochronous transfer continuous transfer mode with programmable buffer size up to 2 KB and optional double buffer • Pipes 3 to 5: Bulk transfer continuous transfer mode with programmable buffer size up to 2 KB and optional double buffer • Pipes 6 to 9: Interrupt transfer with 64-byte single buffer
Other features	• Force-end transfer function using transaction count • Function that changes the BRDY interrupt event notification timing • Automatic clearing of the FIFO buffer after data for the pipe specified in the DnFIFO port (n = 0, 1) is read • NAK setting function for response PID generated on transfer end • On-chip pull-up and pull-down resistors for D+ and D- • Support for Link Power Management (LPM) ECN, including a new Sleep state (the L1 state) • Compliance with Battery Charging Class Specification Revision 1.2 • For power reduction, selectable classic-only mode (CL-only mode) in which operation is only USB 1.1-compliant
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security and Privilege attribution can be set

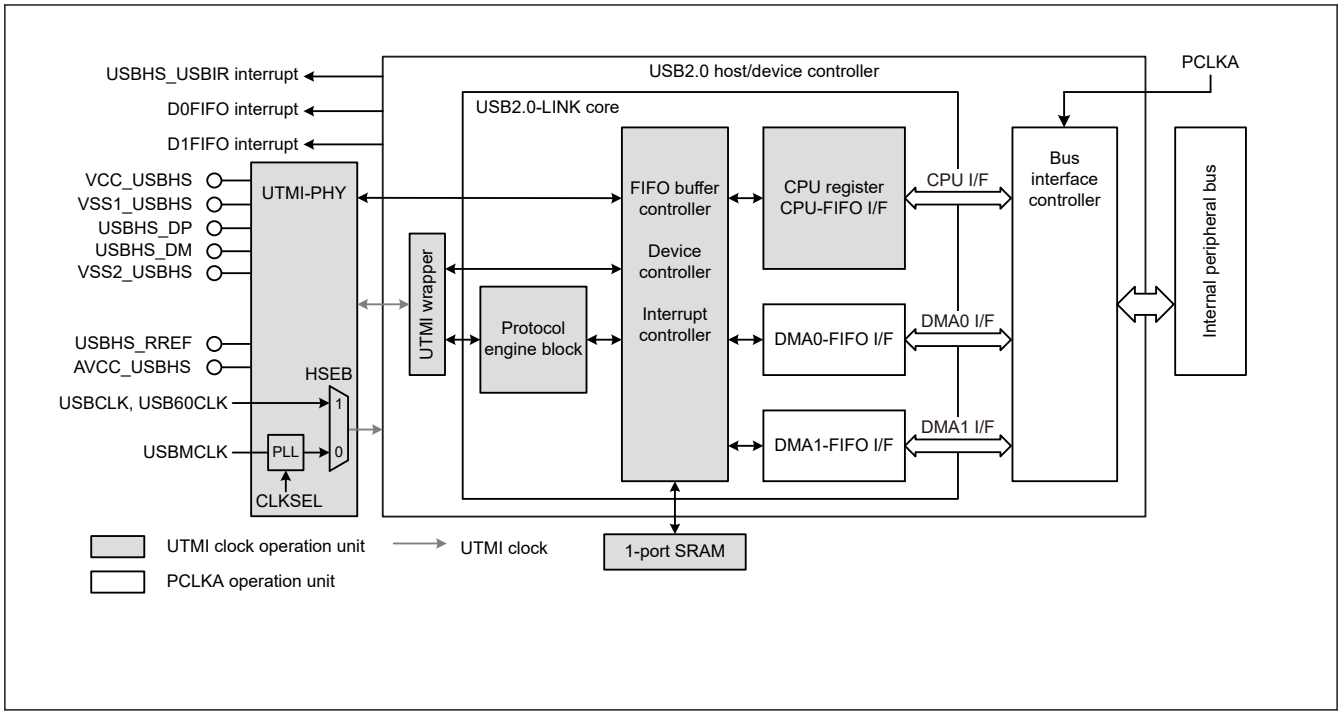


Figure 30.1 USBHS block diagram

Table 30.2 USBHS I/O pins

Pin name	I/O	Function
VCC_USBHS	Input	Power supply pin for the USBHS
VSS1_USBHS VSS2_USBHS	Input	Ground pin for the USBHS
AVCC_USBHS	Input	Analog power supply pin for the USBHS
USBHS_RREF	I/O	Reference current source pin for the USBHS Must be connected to the VSS2_USBHS pin through a 2.2-kΩ (±1%) resistor.
USBHS_DP	I/O	Input/output pin for the D+ data line of the USB bus
USBHS_DM	I/O	Input/output pin for the D- data line of the USB bus
USBHS_EXICEN	Output	Must be connected to the OTG power supply IC
USBHS_ID	Input	Must be connected to the OTG power supply IC
USBHS_VBUSEN	Output	VBUS power supply enable pin for the USBHS
USBHS_OVRCURA, USBHS_OVRCURB, USBHS_OVRCURA-DS, USBHS_OVRCURB-DS	Input	Overcurrent pin for the USBHS USBHS_OVRCURA or USBHS_OVRCURB pins can be used in Software standby mode or in normal mode. USBHS_OVRCURA-DS, USBHS_OVRCURB-DS are dedicated pins that can generate interrupt in Software standby mode or in normal mode as well as for cancel Deep Software Standby mode 1.
USBHS_VBUS	Input	USB cable connection monitor input pin

30.2 Register Descriptions

30.2.1 SYSCFG : System Configuration Control Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x000

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CNEN	HSE	DCFM	DRPD	DPRPU	—	—	—	USBE
Value after reset:	x	x	x	x	x	x	x	0	0	0	1	0	x	x	x	0

Bit	Symbol	Function	R/W
0	USBE	USBHS Operation Enable 0: Disable 1: Enable	R/W
3:1	—	The read values are undefined. The write value should be 0.	R/W
4	DPRPU	D+ Line Resistor Control 0: Disable line pull-up 1: Enable line pull-up	R/W
5	DRPD	D+/D- Line Resistor Control 0: Disable line pull-down 1: Enable line pull-down	R/W
6	DCFM	Controller Operation Select 0: Select device controller mode 1: Select host controller mode	R/W
7	HSE	High-Speed Operation Enable 0: Disable Device controller mode: full-speed Host controller mode: full-speed or low-speed 1: Enable The controller detects the communication speed	R/W
8	CNEN	Single-ended Receiver Enable 0: Disable 1: Enable	R/W
15:9	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Writing to the SYSCFG register can proceed while the PHY clock is stopped. However, written values are only reflected in the SYSCFG register after the PHY clock is oscillating again.

USBE bit (USBHS Operation Enable)

The USBE bit enables or disables operation of USBHS.

Changing the USBE bit from 1 to 0 initializes the bits listed in Table 30.3. Only change this bit after specifying the input clock in the PHYSET.CLKSEL[1:0] bits and confirming that the PLLSTA.PLLLOCK flag is 1. In CL-only mode, change the USBE bit after setting the PHYSET.HSEB bit to 1. At that time, the USBCLK must be set to 48 MHz and USB60CLK must be set to 60 MHz. For the clock settings, see section 30.3.3. Supplying the Clock.

In host controller mode, always set this bit to 1 after setting the DRPD bit to 1, eliminating SYSSTS0.LNST[1:0] bit chattering, and confirming that the USB bus state is stable.

Table 30.3 Bits initialized by writing SYSCFG.USBE = 0

Selected function	Register	Bit	Remarks
Device controller (DCFM = 0)	SYSSTS0	LNST[1:0]	Value is saved in host controller mode
	DVSTCTR0	RHST[2:0]	—
	PL1CTRL1	DVSQ[3:0]	Value is saved in host controller mode
	USBADDR	USBADDR[6:0]	Value is saved in host controller mode
	USBREQ	• BREQUEST[7:0] • BMREQUESTTYPE[7:0]	Value is saved in host controller mode
	USBVAL	WVALUE[15:0]	Value is saved in host controller mode
	USBINDX	WINDEX[15:0]	Value is saved in host controller mode
	USBLENG	WLENTUH[15:0]	Value is saved in host controller mode
Host controller (DCFM = 1)	DVSTCTR0	RHST[2:0]	—
	FRMNUM	FRNM[10:0]	Value is saved in device controller mode
	UFRMNUM	UFRNM[2:0]	Value is saved in device controller mode

DPRPU bit (D+ Line Resistor Control)

The DPRPU bit enables or disables pulling up the D+ line in device controller mode.

When the DPRPU bit is set to 1 in device controller mode, the USBHS pulls up the D+ line to notify the USB host that it attached. Changing the DPRPU bit from 1 to 0 releases the pull-up, thereby notifying the USB host that it detached.

Set this bit to 1 in device controller mode and to 0 in host controller mode.

DRPD bit (D+/D- Line Resistor Control)

The DRPD bit enables or disables pulling down D+ and D- lines in host controller mode.

Set this bit to 1 in host controller mode. Set it to 0 when OTG is not used in device controller mode.

DCFM bit (Controller Operation Select)

The DCFM bit selects the host or device function of the USBHS.

Only change this bit when the DPRPU and DRPD bits are both 0.

HSE bit (High-Speed Operation Enable)

The HSE bit enables or disables high-speed operation.

When this bit is 1, the USBHS operates in high-speed or full-speed based on the results of the reset handshake.

In host controller mode, setting this bit to 0 allows the USBHS to operate in low-speed or full-speed. If the DVSTCTR0.RHST[2:0] flags indicate that a low-speed device has attached, set the HSE bit to 0.

In host controller mode, setting this bit to 1 allows the USBHS to operate in high-speed or full-speed based on the results of the reset handshake. Change the HSE bit after detection of an attach event (ATTCH interrupt) and before the USB bus reset (when DVSTCTR0.USBRST = 1), or after detection of a detach event.

In device controller mode, setting this bit to 0 allows the USBHS to operate in full-speed. Setting the bit to 1 allows the USBHS to perform the reset handshake and then operate in high-speed or full-speed, based on the results.

In device controller mode, only change this bit when the DPRPU bit is 0.

CNEN bit (Single-ended Receiver Enable)

Setting the CNEN bit to 1 enables single-ended receiver operation and selects monitoring of the D+ and D- line states in the SYSSTS0.LNST[1:0] flags. Use this bit to prevent through-current damage that might otherwise be caused during single-ended receiver operation, where the terminals are floating while the USBHS is detached.

In host controller mode, set this bit to 1 after confirming that the PHY clock is being supplied. In device controller mode, set this bit to 1 when the VBUS is detected because of a VBUS interrupt, and set it to 0 when the VBUS line is removed.

30.2.2 BUSWAIT : CPU Bus Wait Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x002

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	BWAIT[3:0]			
Value after reset:	x	x	0	0	0	0	0	0	x	x	0	0	1	1	1	1

Bit	Symbol	Function	R/W
3:0	BWAIT[3:0]	CPU Bus Access Wait Specification 0x0: 0 waits (2 access cycles) ⋮ 0x2: 2 waits (4 access cycles) ⋮ 0x4: 4 waits (6 access cycles) ⋮ 0xF: 15 waits (17 access cycles) (initial value)	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
7:6	—	The read values are undefined. The write value should be 0.	R/W
13:8	—	These bits are read as 0. The write value should be 0.	R/W
15:14	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

BWAIT[3:0] bits (CPU Bus Access Wait Specification)

The BWAIT[3:0] bits specify the wait time for access to the USBHS registers.

When accessing the registers at offset addresses in range 0x004 to 0x15F for writing and range 0x004 to 0x167 for reading, set the cycle time for consecutive access to at least 41 ns. The initial value is 0xF (17 cycles), but Renesas recommends that you satisfy this condition by setting the best wait time for the frequency of PCLKA in your application.

This setting is the same as the wait time for accesses to the FIFO port register. The maximum speed of access to the FIFO port is as follows:

- MBW[1:0] = 10b (32-bit width): Maximum 60 MB/s
- MBW[1:0] = 01b (16-bit width): Maximum 30 MB/s
- MBW[1:0] = 00b (8-bit width): Maximum 15 MB/s

30.2.3 SYSSTS0 : System Configuration Status Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x004

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVCMON[1:0]		—	—	—	—	—	—	—	HTACT	SOFEA	—	—	IDMON	LNST[1:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	0	0	x	x	x	x	x

Bit	Symbol	Function	R/W
1:0	LNST[1:0]	USB Data Line Status Monitor Flag Indicates the status of the USB data lines. See Table 30.4.	R
2	IDMON	USBHS_ID Pin Monitor Flag 0: USBHS_ID pin is low 1: USBHS_ID pin is high	R

Bit	Symbol	Function	R/W
4:3	—	The read values are undefined.	R
5	SOFEA	SOF Active Monitor Flag While Host Controller Operation Is Selected 0: SOF output stopped 1: SOF output operating	R
6	HTACT	Host Sequencer Status Monitor Flag 0: Host sequencer stopped 1: Host sequencer operating	R
13:7	—	The read values are undefined.	R
15:14	OVCMON[1:0]	External USBHS_OVRCURA or USBHS_OVRCURA-DS, and USBHS_OVRCURB or USBHS_OVRCURB-DS Input Pin Monitor Flag OVCMON[1] indicates the USBHS_OVRCURA or USBHS_OVRCURA-DS pin status. OVCMON[0] indicates the USBHS_OVRCURB or USBHS_OVRCURB-DS pin status.	R

Note: S-TYPE-3, P-TYPE-3

LNST[1:0] flags (USB Data Line Status Monitor Flag)

The LNST[1:0] flags indicate the state of the USB data lines, D+ and D-. For details, see Table 30.4.

In device controller mode, read the LNST[1:0] flags after setting the SYSCFG.CNEN and SYSCFG.USBE bits to 1. In host controller mode, read them after setting the SYSCFG.DRPD bit to 1.

When you are checking hardware contacts for the battery charging function in device controller mode, read the LNST[1:0] flags after setting the SYSCFG.DRPD, SYSCFG.CNEN, and BCCTRL.IDPSRCE bits to 1. For details, see section 30.3.15. Battery charging detection processing.

Table 30.4 Status of USB data bus lines (D+ and D-)

LNST[1]	LNST[0]	Low-speed operation (host controller mode only)	Full-speed operation	High-speed operation	Chirp operation
0	0	SE0	SE0	Squelch	Squelch
0	1	K-State	J-State	Unsquelch	Chirp J
1	0	J-State	K-State	Invalid	Chirp K
1	1	SE1	SE1	Invalid	Invalid

Chirp: The reset handshake protocol is being executed when high-speed operation is enabled (HSE bit is 1).

Squelch: SE0 or idle state

Unsquelch: High-speed J-state or high-speed K-state

Chirp J: Chirp J-State

Chirp K: Chirp K-State

SOFEA flag (SOF Active Monitor Flag While Host Controller Operation Is Selected)

The SOFEA flag is used in host controller mode to check whether the output of the last SOF is complete when the USBHS is suspended because of a 0 setting to the DVSTCTR0.UACT bit.

In host controller mode, check that both the HTACT and SOFEA flags are 0 before setting the SYSCFG.USBE bit to 0 to stop the USBHS or setting the LPSTS.SUSPENDM bit to 0 to stop the clock signal supply during communication.

HTACT flag (Host Sequencer Status Monitor Flag)

The HTACT flag clears to 0 when the host sequencer of the USBHS is completely stopped.

In host controller mode, check that the HTACT flag is 0 before setting the DVSTCTR0.UACT bit to 0 to place the USBHS in the Suspend state or setting the LPSTS.SUSPENDM bit to 0 to stop the clock signal supply during communication.

OVCMON[1:0] flags (External USBHS_OVRCURA or USBHS_OVRCURA-DS, and USBHS_OVRCURB or USBHS_OVRCURB-DS Input Pin Monitor Flag)

The OVCMON[1:0] flags indicate the status of the overcurrent signals from an external power supply IC.

30.2.4 PLLSTA : PLL Status Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x006

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLLOCK
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	PLLLOCK	PLL Lock Flag 0: PLL not locked 1: PLL locked	R
15:1	—	The read values are undefined.	R

Note: S-TYPE-3, P-TYPE-3

PLLLOCK flag (PLL Lock Flag)

The PLLLOCK flag indicates whether the USB-PHY internal PLL is locked. When not using CL-only mode, make sure that the PLL is locked before starting USB communication.

30.2.5 DVSTCTR0 : Device State Control Register 0

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x008

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	HNPBTOA	EXICEN	VBUSEN	WKUP	RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
Value after reset:	x	x	x	x	0	0	0	0	0	0	0	0	x	0	0	0

Bit	Symbol	Function	R/W
2:0	RHST[2:0]	USB Bus Reset Status Flag 0 0 0: Communication speed indeterminate (powered state or no connection) 0 0 1: Host controller mode Low-speed connection 0 1 0: Host controller mode Full-speed connection Device controller mode USB bus reset in progress or full-speed connection 0 1 1: Host controller mode High-speed connection Device controller mode USB bus reset in progress or high-speed connection Others: Host controller mode USB bus reset in progress Device controller mode Setting prohibited	R
3	—	The read value is undefined. The write value should be 0.	R/W
4	UACT	USB Bus Operation Enable for the Host Controller Operation 0: Disable downstream port (disable SOF or micro-SOF transmission) 1: Enable downstream port (enable SOF or micro-SOF transmission)	R/W
5	RESUME	Resume Signal Output for the Host Controller Operation 0: Do not output resume signal 1: Output resume signal	R/W

Bit	Symbol	Function	R/W
6	USBRST	USB Bus Reset Output for the Host Controller Operation 0: Do not output USB bus reset signal 1: Output USB bus reset signal	R/W
7	RWUPE	Remote Wakeup Detection Enable for the Host Controller Operation 0: Disable downstream port remote wakeup 1: Enable downstream port remote wakeup	R/W
8	WKUP	Remote Wakeup Output for the Device Controller Operation 0: Do not output remote wakeup signal 1: Output remote wakeup signal	R/W
9	VBUSEN	USBHS_VBUSEN Output Pin Control 0: Output low on external USBHS_VBUSEN pin 1: Output high on external USBHS_VBUSEN pin	R/W
10	EXICEN	USBHS_EXICEN Output Pin Control 0: Output low on external USBHS_EXICEN pin 1: Output high on external USBHS_EXICEN pin	R/W
11	HNPBTOA	Host Negotiation Protocol (HNP) Control Use this bit when switching from device B to device A in OTG mode. If the HNPBTOA bit is 1, the internal function control remains in the Suspend state until the HNP processing ends even if SYSCFG.DPRPU = 0 or SYSCFG.DCFM = 1 is set.	R/W
15:12	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

RHST[2:0] flags (USB Bus Reset Status Flag)

The RHST[2:0] flags indicate the USB bus reset status.

In host controller mode, writing 1 to the USBRST bit causes the RHST[2:0] flags to set to 100b. When 0 is written to the USBRST bit and the USBHS ends the SE0 state, the RHST[2:0] flags update to a new value.

In device controller mode, if the USBHS detects a USB bus reset, the RHST[2:0] flags set to 010b if an attach event occurs while the DPRPU bit is 1, and a DVST interrupt is generated.

UACT bit (USB Bus Operation Enable for the Host Controller Operation)

When set to 1 in host controller mode, the UACT bit enables USB bus operation by controlling SOF packet transmission to the USB bus in addition to data and reception. The USBHS starts SOF packet output within one frame period after the this bit is set to 1. If UACT is set to 0, the USBHS enters the idle state after the SOF packet output.

The USBHS sets the bit to 0 on any of the following conditions:

- A DTCH interrupt is detected during communication (while UACT = 1)
- An EOFERR interrupt is detected during communication (while UACT = 1)

Always write 1 to the UACT bit at the end of the USB bus reset processing (on a 0 write to the USBRST bit) or at the end of resume processing from the Suspend state (on a 0 write to the RESUME bit).

The USBHS clears the UACT bit to 0 if it receives an ACK response to an LPM token while the HL1CTRL1.L1REQ bit is set to 1. The USBHS sets the UACT bit to 1 when it finishes resume processing from the L1 state.

In device controller mode, always set this bit to 0.

RESUME bit (Resume Signal Output for the Host Controller Operation)

The RESUME bit controls the resume signal output in host controller mode.When this bit is set to 1, the USBHS drives the USB port to the K-state and outputs the resume signal. The USBHS sets the bit to 1 on detection of a remote wakeup signal while the RWUPE bit is 1 and in the USB suspend state. The USBHS continues outputting the K-state while the RESUME bit is 1, until the bit is cleared to 0 by software. The RESUME bit must be 1 (= resume period) for the time defined in the USB 2.0 specification. Only set this bit to 1 while the interface is in the Suspend state. Write 1 to the UACT bit simultaneously with the end of the resume processing (0 write to the RESUME bit).

Setting the RESUME bit to 1 during transition to the L1 state allows the USBHS to drive the USB port to the K-state and output the resume signal. The USBHS clears the RESUME bit to 0 at the end of the resume period, the value set in the HL1CTRL2.HIRD[3:0] bits.

Always set this bit to 0 in device controller mode.

USBRST bit (USB Bus Reset Output for the Host Controller Operation)

The USBRST bit controls the output of the USB bus signal in host controller mode. When this bit set to 1, the USBHS drives the USB port to the SE0 state to reset the USB bus. The USBHS continues outputting SE0 while the USBRST bit is 1, until the bit is cleared to 0 by software. The USBRST bit must be 1 (= USB bus reset period) for the time defined in the USB 2.0 specification. Writing 1 to the USBRST bit during communication (UACT bit = 1) or during resume processing (RESUME bit = 1) prevents the USBHS from starting USB bus reset processing until both the UACT and RESUME bits clear to 0. Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (0 write to the USBRST bit).

Always set this bit to 0 in device controller mode.

RWUPE bit (Remote Wakeup Detection Enable for the Host Controller Operation)

The RWUPE bit enables or disables remote wakeup signals (resume signals) from downstream peripheral devices in host controller mode. When this bit is set to 1, the USBHS detects a remote wakeup signal (K-state for 2.5 µs) from a downstream peripheral device, and it performs resume processing, driving the K-state. When the RWUPE bit is set to 0, the USBHS ignores remote wakeup signals (K-states) from peripheral devices connected to the USB port.

Do not stop the PHY clock while the RWUPE bit is 1, even in the Suspend state (the LPSTS.SUSPENDM bit must be set to 1). Also, do not reset the USB bus (setting USBRST to 1) from the Suspend state. This is prohibited in the USB 2.0 specification.

The RWUPE bit is also used to enable or disable detection of a remote wakeup signal during transition to the L1 state.

Always set this bit to 0 in device controller mode.

WKUP bit (Remote Wakeup Output for the Device Controller Operation)

The WKUP bit enables or disables remote wakeup signals (resume signals) to the USB bus in device controller mode.

The USBHS controls the output timing of the remote wakeup signals. When this bit is set to 1, the USBHS clears it to 0 after outputting the K-state for 10 ms. The USB 2.0 specification dictates that the USB bus idle state must be maintained for 5 ms or longer before a remote wakeup signal is sent. If the USBHS writes 1 to the WKUP bit immediately after detecting the Suspend state, the K-state is output after 2 ms.

Only write 1 to the WKUP bit when the device is in the Suspend state (the PL1CTRL1.DVSQ[3:0] flags are 01xxb) and the USB host enables the remote wakeup signal. Do not stop the PHY clock while this bit is 1, even in the Suspend state (the LPSTS.SUSPENDM bit must be set to 1).

If the WKUP bit is set to 1 during transition to the L1 state, the USBHS outputs the K-state for 50 µs and then clears the bit to 0. Before writing 1 to the bit during the L1 state, check that the PL1CTRL1.DVSQ[3:0] flags are 10xxb.

Always set this bit to 0 in host controller mode.

HNPBTOA bit (Host Negotiation Protocol (HNP) Control)

The HNPBTOA bit is used when switching from device B to device A while in OTG mode.

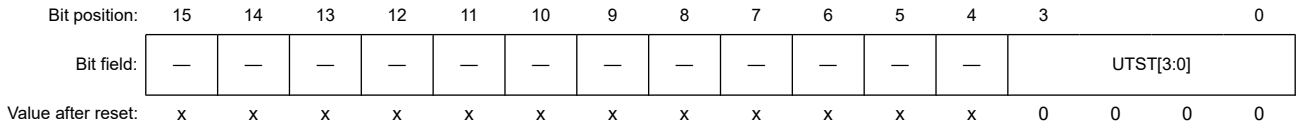
If the HNPBTOA bit is 1, the internal function control maintains the Suspend state until HNP processing ends, even if the SYSCFG.DPRPU bit is set to 0 or the SYSCFG.DCFM bit is set to 1. Resume interrupts (RESM) are not generated even if a falling edge of D+ is detected.

The HNP processing ends when a host attach event is detected, because of a pull-up by the initiating party, or the HNPBTOA bit is cleared to 0 by software because the HNP processing times out.

30.2.6 TESTMODE : USB Test Mode Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x00C



Bit	Symbol	Function	R/W
3:0	UTST[3:0]	Test Mode These bits output the USB test signals. See Table 30.5	R/W
15:4	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

UTST[3:0] bits (Test Mode)

Writing values to the UTST[3:0] bits allows the USBHS to output USB test signals in high-speed operation mode. Table 30.5 shows the test mode operation settings.

Table 30.5 Test mode operation settings

Test mode	UTST[3:0] bit setting	
	In device controller mode	In host controller mode
Normal operation	0x0	0x0
Test_J	0x1	0x9
Test_K	0x2	0xA
Test_SE0_NAK	0x3	0xB
Test_Packet	0x4	0xC
Test_Force_Enable	—	0xD
Reserved	0x5 to 0x7	0xE to 0xF

Host controller mode

In host controller mode, these bits can be set after setting the SYSCFG.DRPD bit to 1. After the UTST[3:0] bits are set, the USBHS outputs waveforms to the USB port by setting the DVSTCTR0.UACT bit to 1. The USBHS also performs high-speed termination for the USB port by setting these bits in host controller mode.

To set the UTST[3:0] bits in host controller mode:

1. Reset the hardware.
2. Start supplying the PHY clock, and then set the LPSTS.SUSPENDM bit to 1.
3. Set the SYSCFG.DCFM and SYSCFG.DRPD bits to 1. (Setting the SYSCFG.HSE bit to 1 is not required.)
4. Set the SYSCFG.USBE bit to 1.
5. Set the UTST[3:0] bits based on the test requirements.
6. Set the DVSTCTR0.UACT bit to 1.

Assuming the initial steps (1) to (6) are already complete, to change the UTST[3:0] bits in host controller mode:

1. Set the DVSTCTR0.UACT and SYSCFG.USBE bits to 0.
2. Set the SYSCFG.USBE bit to 1.
3. Set the UTST[3:0] bits based on the test requirements.
4. Set the DVSTCTR0.UACT bit to 1.

When the UTST[3:0] bits are set to 1011b (Test_SE0_NAK), the USBHS does not output SOF packets to ports for which the DVSTCTR0.UACT bit is set to 1.

When the UTST[3:0] bits are set to 1101b (Test_Force_Enable), the USBHS outputs SOF packets to ports for which the DVSTCTR0.UACT bit is set to 1. In this test mode, the USBHS does not control the hardware related to attach detection, even if it detects a high-speed detach event (DTCH interrupt).

Before setting the UTST[3:0] bits, set the PID[1:0] bits of all pipe control registers to 00b (NAK response). To return to normal USB communication after setting a test mode, issue a hardware reset.

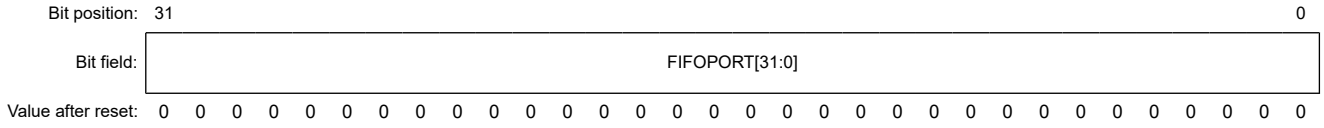
Device controller mode

In device controller mode, set these bits using a SetFeature request from the USB host during high-speed communication. The USBHS does not enter the Suspend state while these bits are 0001b to 0100b. To return to normal USB communication after setting a test mode, issue a hardware reset.

30.2.7 CFIFO, DnFIFO : FIFO Port Register (n = 0, 1)

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x014 (CFIFO/CFIFOL/CFIFOLL)
0x016 (CFIFOH)
0x017 (CFIFOHH)
0x018 + 0x04 × n (DnFIFO/DnFIFOL/DnFIFOLL)
0x01A + 0x04 × n (DnFIFOH)
0x01B + 0x04 × n (DnFIFOHH)



Bit	Symbol	Function	R/W
31:0	FIFOPORT[31:0]*1	Read receive data from the FIFO buffer or write transmit data to the FIFO buffer by accessing these bits.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The valid bits depend on the MBW[1:0] and BIGEND settings in the associated port selection register.

Three FIFO ports are provided:

- CFIFO
- D0FIFO
- D1FIFO

Each FIFO port is configured with:

- A port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer and writing of data to the FIFO buffer
- A port selection register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that selects the pipe assigned to the FIFO port
- A port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR)

Each FIFO port has the following constraints:

- Access to the FIFO buffer for DCP control transfers is through the CFIFO port
- Access to the FIFO buffer for DMAC or DTC transfers is through the D0FIFO or D1FIFO port
- The D0FIFO and D1FIFO ports can also be accessed by the CPU
- When using functions specific to the FIFO port, such as the DMAC or DTC transfer function, you cannot change the pipe number selected in the CURPIPE[3:0] bits of the Port Selection Register
- Registers configuring one FIFO port do not affect other FIFO ports
- The same pipe must not be assigned to two or more FIFO ports

- There are two FIFO buffer states, one giving access rights to the CPU and the other to the serial interface engine (SIE). When the SIE has access rights, the FIFO buffer cannot be accessed by the CPU

FIFOPORT bits (FIFOPORT[31:0])

When the FIFOPORT bits is accessed, the USBHS reads the received data from the FIFO buffer or writes the transmission data to the FIFO buffer. The FIFO port register can be accessed only when the FRDY flag in the associated port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in the FIFO port register depend on the MBW[1:0] and BIGEND settings in the port selection register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL). See Table 30.6 to Table 30.8.

Table 30.6 Endian operation in 32-bit access (MBW[1:0] = 10b)

BIGEND	CFIFO, D0FIFO, D1FIFO b31 to b24	CFIFO, D0FIFO, D1FIFO b23 to b16	CFIFO, D0FIFO, D1FIFO b15 to b8	CFIFO, D0FIFO, D1FIFO b7 to b0	Remarks
0	Located at N+3	Located at N+2	Located at N+1	Located at N+0	Transmit data is sent from the address N+0. Receive data is stored from the address N+0.
1	Located at N+0	Located at N+1	Located at N+2	Located at N+3	Transmission data is sent from the address N+3. Receive data is stored from the address N+3.

Table 30.7 Endian operation in 16-bit access (MBW[1:0] = 01b)

BIGEND	CFIFOL, D0FIFOL, D1FIFOL b15 to b8	CFIFOL, D0FIFOL, D1FIFOL b7 to b0	CFIFOH, D0FIFOH, D1FIFOH b15 to b8	CFIFOH, D0FIFOH, D1FIFOH b7 to b0	Remarks
0	Access prohibited*1		Located at N+1	Located at N+0	Transmit data is sent from the address N+0. Receive data is stored from the address N+0.
1	Located at N+0	Located at N+1	Access prohibited*1		Transmit data is sent from the address N+1. Receive data is stored from the address N+1.

Note 1. Writing to or reading from these areas is prohibited.

Table 30.8 Endian operation in 8-bit access (MBW[1:0] = 00b)

BIGEND	CFIFOLL, D1FIFOLL, D0FIFOLL	CFIFOHH, D1FIFOHH, D0FIFOHH
0	Access prohibited*1	Located at N+0
1	Located at N+0	Access prohibited*1

Note 1. Writing to or reading from these locations is prohibited.

30.2.8 CFIFOSEL : CFIFO Port Selection Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x020

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RCNT	REW	—	—	MBW[1:0]		—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]			
Value after reset:	0	0	x	x	0	0	x	0	x	x	0	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CURPIPE[3:0]	FIFO Port Access Pipe Specification 0x0: DCP (default control pipe) 0x1: Pipe 1 0x2: Pipe 2 0x3: Pipe 3 0x4: Pipe 4 0x5: Pipe 5 0x6: Pipe 6 0x7: Pipe 7 0x8: Pipe 8 0x9: Pipe 9 Others: Setting prohibited	R/W
4	—	The read value is undefined. The write value should be 0.	R/W
5	ISEL	FIFO Port Access Direction when DCP Is Selected 0: Select reading from the FIFO buffer 1: Select writing to the FIFO buffer	R/W
7:6	—	The read values are undefined. The write value should be 0.	R/W
8	BIGEND	FIFO Port Endian Control 0: Little endian 1: Big endian	R/W
9	—	The read value is undefined. The write value should be 0.	R/W
11:10	MBW[1:0]	CFIFO Port Access Bit Width 0 0: 8-bit width 0 1: 16-bit width 1 0: 32-bit width 1 1: Setting prohibited	R/W
13:12	—	The read values are undefined. The write value should be 0.	R/W
14	REW	Buffer Pointer Rewind 0: Do not rewind buffer pointer (Writing 0 has no effect.) 1: Rewind buffer pointer	W
15	RCNT	Read Count Mode 0: Clear DTLN[11:0] flags in the FIFO port control register to 0x000 when all receive data is read from CFIFO 1: Decrement DTLN[11:0] flags each time receive data is read from CFIFO	R/W

Note: S-TYPE-3, P-TYPE-3

Do not specify the same pipe number in the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.

Do not change the pipe number while DMAC or DTC transfer is enabled.

CURPIPE[3:0] bits (FIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number used to read or write data through the CFIFO port. After writing to these bits, read them to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

During FIFO buffer access, the pipe specification is maintained until the access is complete, even if the software attempts to change the CURPIPE[3:0] setting. Access continues after the current value is written back to the CURPIPE[3:0] bits.

ISEL bit (FIFO Port Access Direction when DCP Is Selected)

After writing a new value to the ISEL bit while the DCP is the selected pipe, read this bit to check that the written value agrees with the read value before proceeding to the next process. Set the ISEL and CURPIPE[3:0] bits simultaneously.

BIGEND bit (FIFO Port Endian Control)

Use the BIGEND bit to set the byte endian order of the CFIFO port.

MBW[1:0] bits (CFIFO Port Access Bit Width)

The MBW[1:0] bits specify the bit width for accessing the CFIFO port.

When the selected pipe is receiving, after a write to these bits starts a data read from the FIFO buffer, do not change the MBW[1:0] bits until all of the data is read. When reading the FIFO buffer, read with the access size set in MBW.

When the selected pipe is transmitting, set the CURPIPE[3:0] and MBW[1:0] bits simultaneously. The bit width cannot be changed from 8-bit to 16- or 32-bit, or from 16-bit to 32-bit while data is being written to the FIFO buffer.

An odd number of bytes can also be written through byte-access control even when 16- or 32-bit width is selected.

REW bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.

When the selected pipe is receiving, setting this bit to 1 while the FIFO buffer is being read allows re-reading of the FIFO buffer from the first data. In double-buffering when reading is already in progress, this setting enables reading either FIFO buffer from the first entry.

Do not set this bit to 1 while simultaneously changing the CURPIPE[3:0] bits. Before setting the bit to 1, always check that the FRDY flag is 1.

To rewrite to the FIFO buffer from the first data for the transmitting pipe, use the BCLR bit.

RCNT bit (Read Count Mode)

When the RCNT bit set to 0, the USBHS clears the CFIFOCTR.DTLN[11:0] flags to 0 on finishing reading all of the received data in the FIFO buffer assigned to the pipe specified in the CURPIPE[3:0] bits, or after reading a single plane in double buffer mode.

With this bit set to 1, the USBHS decrements the value in the CFIFOCTR.DTLN[11:0] flags each time it reads data received from the FIFO buffer assigned to the pipe specified in the CURPIPE[3:0] bits.

30.2.9 DnFIFOSEL : DnFIFO Port Selection Register (n = 0, 1)

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x028 + 0x4 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RCNT	REW	DCLRM	DREQE	MBW[1:0]		—	BIGEND	—	—	—	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	x	0	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CURPIPE[3:0]	FIFO Port Access Pipe Specification 0x0: No pipe specification 0x1: Pipe 1 0x2: Pipe 2 0x3: Pipe 3 0x4: Pipe 4 0x5: Pipe 5 0x6: Pipe 6 0x7: Pipe 7 0x8: Pipe 8 0x9: Pipe 9 Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
7:4	—	The read values are undefined. The write value should be 0.	R/W
8	BIGEND	FIFO Port Endian Control 0: Little endian 1: Big endian	R/W
9	—	The read value is undefined. The write value should be 0.	R/W
11:10	MBW[1:0]	FIFO Port Access Bit Width 0 0: 8-bit width 0 1: 16-bit width 1 0: 32-bit width 1 1: Setting prohibited	R/W
12	DREQE	DMAC/DTC Transfer Request Enable 0: Disable DMAC/DTC transfer request 1: Enable DMAC/DTC transfer request	R/W
13	DCLRM	Auto FIFO Buffer Clear Mode after Specified Pipe is Read 0: Disable auto buffer clear mode 1: Enable auto buffer clear mode	R/W
14	REW	Buffer Pointer Rewind 0: Do not rewind buffer pointer (writing 0 has no effect) 1: Rewind buffer pointer	W
15	RCNT	Read Count Mode 0: Clear DTLN[11:0] flags in the FIFO port control register to 0x000 when all receive data is read from DnFIFO (after read of a single plane in double buffer mode) 1: Decrement DTLN[11:0] flags each time receive data is read from DnFIFO	R/W

Note: S-TYPE-3, P-TYPE-3

Do not specify the same pipe number in the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

Do not change the pipe number while DMAC or DTC transfer is enabled.

CURPIPE[3:0] bits (FIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number used to read or write data through the DnFIFO port. After writing to these bits, read them to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

During FIFO buffer access, the pipe specification is maintained until the access is complete, even if the software attempts to change the CURPIPE[3:0] setting. Access continues after the current value is written back to the CURPIPE[3:0] bits.

BIGEND bit (FIFO Port Endian Control)

Use the BIGEND bit to set the byte endian order of the D0FIFO or D1FIFO port.

MBW[1:0] bits (FIFO Port Access Bit Width)

The MBW[1:0] bits specify the bit width for accessing the DnFIFO port.

When the selected pipe is receiving, after a write to these bits starts a data read from the FIFO buffer, do not change the MBW[1:0] bits until all of the data is read. When reading the FIFO buffer, read with the access size set in MBW.

When the selected pipe is transmitting, set the CURPIPE[3:0] and MBW[1:0] bits simultaneously. The bit width cannot be changed from 8-bit to 16- or 32-bit, or from 16-bit to 32-bit while data is being written to the FIFO buffer.

An odd number of bytes can also be written through byte-access control even when 16- or 32-bit width is selected.

DREQE bit (DMAC/DTC Transfer Request Enable)

The DREQE bit enables or disables issuing of DMAC or DTC transfer requests. Only change the settings of DREQE bit when the CURPIPE[3:0] bits are 0x0.

To enable DMAC or DTC transfer requests, set this bit to 1 after setting the CURPIPE[3:0] bits to 0x0, and then set the CURPIPE[3:0] bits to the PIPE number for the transfer.

DCLRM bit (Auto FIFO Buffer Clear Mode after Specified Pipe is Read)

The DCLRM bit enables or disables automatic FIFO buffer clearing after data in the selected pipe is read.

When this bit is set to 1, on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or when reading of a received short packet is complete while the PIPECFG.BFRE bit is 1, the USBHS sets the BCLR bit in the FIFO port control register to 1.

When using the USBHS with the SOFCFG.BRDYM bit set to 1, set this bit to 0.

REW bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.

When the selected pipe is receiving, setting this bit to 1 while the FIFO buffer is being read allows re-reading of the FIFO buffer from the first data. In double-buffering when reading is already in progress, this setting enables reading either FIFO buffer from the first entry.

Do not set this bit to 1 while simultaneously changing the CURPIPE[3:0] bits. Before setting the bit to 1, always check that the FRDY flag is 1.

To rewrite to the FIFO buffer from the first data for the transmitting pipe, use the BCLR bit.

RCNT bit (Read Count Mode)

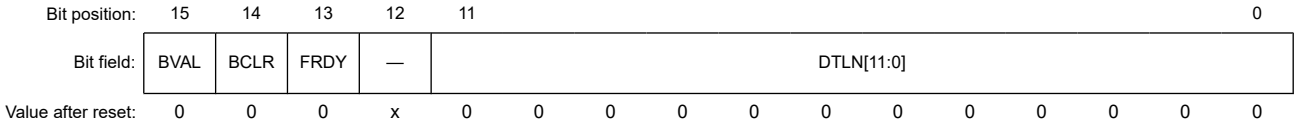
When the RCNT bit set to 0, the USBHS clears the DnFIFOCTR.DTLN[11:0] flags (n = 0, 1) to 0 on finishing reading all of the received data in the FIFO buffer assigned to the pipe specified in the CURPIPE[3:0] bits, or after reading a single plane in double buffer mode.

With this bit set to 1, the USBHS decrements the value in the CFIFOCTR.DTLN[11:0] flags each time it reads data received from the FIFO buffer assigned to the pipe specified in the CURPIPE[3:0] bits. When accessing DnFIFO with the PIPECFG.BFRE bit set to 1, set the RCNT bit to 0.

30.2.10 CFIFOCTR, DnFIFOCTR : FIFO Port Control Register (n = 0, 1)

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x022 (CFIFOCTR)
0x02A + 0x4 × n (DnFIFOCTR)



Bit	Symbol	Function	R/W
11:0	DTLN[11:0]	Receive Data Length Flag The meaning of the values differs depending on the RCNT bit setting in the port selection register. For details, see the description of the DTLN[11:0] bits.	R
12	—	The read value is undefined. The write value should be 0.	R/W
13	FRDY	FIFO Port Ready Flag 0: FIFO port access disabled 1: FIFO port access enabled	R
14	BCLR	CPU Buffer Clear 0: No operation (writing 0 has no effect) 1: Clear FIFO buffer on the CPU side	W
15	BVAL	FIFO Buffer Valid Flag Set this bit to 1 when data is completely written to the FIFO buffer on the CPU side for the selected pipe (CURPIPE[3:0] setting). 0: Invalid (writing 0 has no effect) 1: Writing ended	R/W

Note: S-TYPE-3, P-TYPE-3

The CFIFOCTR, D0FIFOCTR, and D1FIFOCTR registers correspond to the CFIFO, D0FIFO, and D1FIFO buffers.

DTLN[11:0] flags (Receive Data Length Flag)

The DTLN[11:0] flags indicate the length of the receive data.

While the FIFO buffer is being read, the DTLN[11:0] bits indicate different values depending on the DnFIFOSEL.RCNT bit (n = 0, 1), as follows:

- RCNT = 0:
  The USBHS sets the DTLN[11:0] flags to indicate the length of the receive data until the CPU or DMAC/DTC has read all of the received data in the FIFO buffer (or until it has read a single plane in double buffer mode).
  While the PIPECFG.BFRE bit is 1, the USBHS retains the length of the receive data until the BCLR bit is set to 1, even after all the data is read.
- RCNT = 1:
  The USBHS decrements the value indicated in the DTLN[11:0] flags each time the CPU or DMAC/DTC reads the receive data from the FIFO buffer. (The value is decremented by 1 when MBW[1:0] = 00b, by 2 when MBW[1:0] = 01b, and by 4 when MBW[1:0] = 10b.)
  The USBHS sets these flags to 0 when all the data is read from the FIFO buffer. In double buffer mode, if data is received in one FIFO buffer plane before all of the data is read from the other plane, the USBHS sets these bits to indicate the length of the receive data in the latter plane when all of the data is read from the former plane.
  When the RCNT bit is 1, reading the DTLN[11:0] flags while the FIFO buffer is being read returns the latest value within 150 ns after the FIFO port read cycle.

FRDY flag (FIFO Port Ready Flag)

The FRDY flag indicates whether the FIFO port can be accessed by the CPU or DMAC/DTC.

In the following cases, the USBHS sets the FRDY flag to 1 but data cannot be read through the FIFO port because there is no data to be read:

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty
- A short packet is received and the data is completely read while the PIPECFG.BFRE bit is 1

In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

BCLR bit (CPU Buffer Clear)

Set the BCLR bit to 1 to clear the FIFO buffer on the CPU for the selected pipe.

When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USBHS clears only one plane of the FIFO buffer even when both planes are read-enabled.

When the DCP is the selected pipe, setting the BCLR bit to 1 allows the USBHS to clear both sets of FIFO buffers regardless of whether the CPU or SIE has access rights. To clear the buffer when the SIE has access rights, set the DCPCTR.PID[1:0] bits to 00b (NAK response) before setting the BCLR bit to 1.

When the selected pipe is not the DCP, only write 1 to the BCLR bit while the FRDY flag in the FIFO port control register is 1 (set by the USBHS).

BVAL bit (FIFO Buffer Valid Flag)

Set the BVAL bit to 1 when data is completely written to the FIFO buffer on the CPU for the pipe selected in CURPIPE[3:0].

When the selected pipe is transmitting, set this bit to 1 in the following cases:

- To transmit a short packet, set this bit to 1 after data is written
- To transmit a zero-length packet, set this bit to 1 before data is written to the FIFO buffer
- Set this bit to 1 after the specified number of data bytes is written for the pipe in continuous transfer mode, where the number is a natural integer multiple of the maximum packet size and less than the buffer size

The USBHS then switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

When the selected pipe is in use for transmission, simultaneously setting the BVAL flag and the BCLR bit to 1 causes the USBHS to clear the data that is already written and enables transmission of a zero-length packet. When data of the maximum packet size is written for the pipe in non-continuous transfer mode, the USBHS sets this bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

Only write 1 to the BVAL flag while the FRDY bit is 1 (set by the USBHS). When the selected pipe is receiving, do not set the BVAL flag to 1.

30.2.11 INTENB0 : Interrupt Enable Register 0

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x030

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	—	The read values are undefined. The write value should be 0.	R/W
8	BRDYE	Buffer Ready Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
9	NRDYE	Buffer Not Ready Response Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
10	BEMPE	Buffer Empty Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
11	CTRE	Control Transfer Stage Transition Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
12	DVSE	Device State Transition Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
13	SOFE	Frame Number Update Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
14	RSME	Resume Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
15	VBSE	VBUS Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W

Note: S-TYPE-3, P-TYPE-3
Note: The RSME, DVSE, and CTRE bits can only be set to 1 in device controller mode. Do not set these bits to 1 in host controller mode.

When a status flag in the INTSTS0 register sets to 1 and the associated interrupt request enable bit setting in the INTENB0 register is 1, the USBHS issues a USBHS interrupt request.

Regardless of the INTENB0 register setting, the status flag in the INTSTS0 register sets to 1 in response to a state change that satisfies the associated condition.

When an interrupt request enable bit in the INTENB0 register is switched from 0 to 1 while the associated status flag in the INTSTS0 register is set to 1, a USBHS interrupt is requested.

30.2.12 INTENB1 : Interrupt Enable Register 1

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x032

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVRCRE	BCHGE	—	DTCHE	ATTCHE	—	L1RSMENDE	LPMENDE	—	EOFERRE	SIGNE	SACKE	—	—	—	PDDETINTE
Value after reset:	0	0	x	0	0	x	0	0	x	0	0	0	x	x	x	0

Bit	Symbol	Function	R/W
0	PDDETINTE	PDDETINT Detection Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
3:1	—	The read values are undefined. The write value should be 0.	R/W
4	SACKE	Setup Transaction Normal Response Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
5	SIGNE	Setup Transaction Error Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
6	EOFERRE	EOF Error Detection Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
7	—	The read values are undefined. The write value should be 0.	R/W
8	LPMENDE	LPM Transaction End Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
9	L1RSMENDE	L1 Resume End Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
10	—	The read values are undefined. The write value should be 0.	R/W
11	ATTCHE	Connection Detection Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
12	DTCHE	Disconnection Detection Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
13	—	The read values are undefined. The write value should be 0.	R/W
14	BCHGE	USB Bus Change Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
15	OVRCRE	OVRCRE Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W

Note: S-TYPE-3, P-TYPE-3

When a status flag in the INTSTS1 register sets to 1 and the associated interrupt request enable bit setting in the INTENB1 register is 1, the USBHS issues a USBHS interrupt request.

Regardless of the INTENB1 register setting, the status flag in the INTSTS1 register sets to 1 in response to a state change that satisfies the associated condition.

When an interrupt request enable bit in the INTENB1 register is switched from 0 to 1 while the associated status flag in the INTSTS1 register is set to 1, a USBHS interrupt is requested.

30.2.13 BRDYENB : BRDY Interrupt Enable Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x036

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPEBRDYE[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPEBRDYE[9:0]	BRDY Interrupt Request Enable for Pipes [9:0]*1 0: Disable interrupt request 1: Enable interrupt request	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Each bit number corresponds to the same pipe number.

The BRDYENB register enables or disables the INTSTS0.BRDY bit to be set to 1 when a BRDY interrupt is detected for each pipe.

When a status flag in the BRDYSTS register sets to 1 and the associated PIPEBRDYEn (n = 9 to 0) bit setting in the BRDYENB register is 1, the INTSTS0.BRDY flag sets to 1. In this case, if the BRDYE bit in INTENB0 is 1, the USBHS generates a BRDY interrupt request. While at least one PIPEBRDYEn flag indicates 1, the INTSTS0.BRDY flag sets to 1 when the associated interrupt request enable bit in the BRDYENB register is changed from 0 to 1 by software.

30.2.14 NRDYENB : NRDY Interrupt Enable Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x038

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPENRDYE[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPENRDYE[9:0]	NRDY Interrupt Enable for Pipes [9:0]*1 0: Disable interrupt request 1: Enable interrupt request	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Each bit number corresponds to the same pipe number.

The NRDYENB register enables or disables the INTSTS0.NRDY bit to be set to 1 when a NRDY interrupt is detected for each pipe.

When a status flag in the NRDYSTS register sets to 1 and the associated PIPENRDYEn (n = 0 to 9) bit setting in the NRDYENB register is 1, the INTSTS0.NRDY flag sets to 1. In this case, if the NRDYE bit in INTENB0 is 1, the USBHS generates a NRDY interrupt request. While at least one PIPEBRDYEn flag indicates 1, the INTSTS0.NRDY flag sets to 1 when the associated interrupt request enable bit in the NRDYENB register is changed from 0 to 1 by software.

30.2.15 BEMPENB : BEMP Interrupt Enable Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x03A

Bit position:	15	14	13	12	11	10	9									0
Bit field:	—	—	—	—	—	—	PIPEBEMPE[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPEBEMPE[9:0]	BEMP Interrupt Enable for Pipes [9:0]*1 0: Disable interrupt request 1: Enable interrupt request	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Each bit number corresponds to the same pipe number.

The BEMPENB register enables or disables the INTSTS0.BEMP bit to be set to 1 when a BEMP interrupt is detected for each pipe.

When a status flag in the BEMPSTS register sets to 1 and the associated PIPEBEMPEn (n = 0 to 9) bit setting in the BEMPENB register is 1, the INTSTS0.BEMP flag sets to 1. In this case, if the BEMPE bit in INTENB0 is 1, the USBHS generates a BEMP interrupt request. While at least one PIPEBEMPEn flag indicates 1, the INTSTS0.BEMP flag sets to 1 when the associated interrupt request enable bit in the BEMPENB register is changed from 0 to 1 by software.

30.2.16 SOFCFG : SOF Output Configuration Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x03C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TRNENSEL	—	BRDYM	INTL	EDGESTS	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	0	x	0	0	0	x	x	x	x

Bit	Symbol	Function	R/W
3:0	—	The read values are undefined. The write value should be 0.	R/W
4	EDGESTS	Interrupt Edge Processing Status Flag*1 Indicates 1 during the edge processing of an edge interrupt output signal.	R
5	INTL	Interrupt Output Sense Select*2 0: Edge detection 1: Level detection	R/W
6	BRDYM	PIPEBRDY Interrupt Status Clear Timing*3 0: Clear BRDY flag through software 1: Clear BRDY flag by the USBHS through a data read from the FIFO buffer or data write to the FIFO buffer	R/W
7	—	The read values are undefined. The write value should be 0.	R/W
8	TRNENSEL	Transaction-Enabled Time Select*4 0: Not low-speed communication 1: Low-speed communication	R/W
15:9	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Confirm that the EDGESTS flag is 0 before stopping the clock supply to the USBHS.

Note 2. When the INTL bit is set to 0, to stop the PHY clock (LPSTS.SUSPENDM = 0) after clearing the interrupt status, write 0 to the LPSTS.SUSPENDM bit after confirming that the EDGESTS flag is cleared to 0.
Note 3. When setting the BRDYM bit to 1, set the INTL bit to 1.
Note 4. The setting in the TRNENSEL bit is only valid in host controller mode. Even in host controller mode, the setting of this bit has no effect on the transaction-enabled time during high-speed communication.

EDGESTS flag (Interrupt Edge Processing Status Flag)

The EDGESTS flag indicates 1 during the edge processing of an edge interrupt output signal. Confirm that this flag is 0 before stopping the PHY clock.

BRDYM bit (PIPEBRDY Interrupt Status Clear Timing)

The BRDYM bit specifies how the BRDY interrupt status flags for the pipes are cleared.

TRNENSEL bit (Transaction-Enabled Time Select)

When the USB port is in use for full-speed or low-speed communications, the TRNENSEL bit specifies the timing with which the USBHS issues tokens in a frame (transaction-enabled time).

Set this bit to 1 when a low-speed device is connected through a hub. The bit is only valid in host controller mode. Set this bit to 0 when the interface is in use as a device controller.

30.2.17 PHYSET : PHY Setting Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x03E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	HSEB	—	—	—	REPSTART	—	REPSEL[1:0]		—	—	CLKSEL[1:0]		CDPEN	—	PLLRESET	DIRPD
Value after reset:	x	x	x	x	0	x	0	0	x	x	1	1	0	x	1	1

Bit	Symbol	Function	R/W
0	DIRPD	Power-Down Control 0: Do not enter low power mode 1: Enter low power mode	R/W
1	PLLRESET	PLL Reset Control*1 0: Disable PLL reset control for UTMI_PHY 1: Enable PLL reset control for UTMI_PHY	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CDPEN	Charging Downstream Port Enable 0: Disable downstream port charging 1: Enable downstream port charging	R/W
5:4	CLKSEL[1:0]	Input Clock Frequency Select 0 0: 12 MHz 0 1: 48 MHz 1 0: 20 MHz 1 1: 24 MHz	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
9:8	REPSEL[1:0]	Terminating Resistance Adjustment Cycle 0 0: No cycle is set 0 1: Adjust terminating resistance at 16-second intervals 1 0: Adjust terminating resistance at 64-second intervals 1 1: Adjust terminating resistance at 128-second intervals	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	REPSTART	Forcibly Start Terminating Resistance Adjustment 0: Force terminating resistance adjustment to start 1: Do not force terminating resistance adjustment to start	R/W
14:12	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15	HSEB	CL-only mode 0: Disable CL-only mode 1: Enable CL-only mode	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Because the value of the PLLRESET bit is 1 after a reset, changing the setting after release from the reset state is not required. Do not set the PLLRESET bit to 1 after setting the PLLRESET bit to 0. Operation is not guaranteed.

CLKSEL[1:0] bits (Input Clock Frequency Select)

The CLKSEL[1:0] bits select the transfer clock source for the USBHS.

For the transfer clock generated in the USB-PHY internal PLL, these bits set the input clock frequency. To input the clock source from the EXTAL pin, the USB 2.0 clock specification must be strictly followed.

Writing to the CKSEL[1:0] bits is invalid in CL-only mode because the internal PLL is stopped (see the description for HSEB bit (CL-only mode)). For the clock settings, see section 30.3.3. Supplying the Clock.

HSEB bit (CL-only mode)

The HSEB bit selects whether the USBHS operates in CL-only mode. High-speed transfer by the USBHS requires the use of internal high-speed analog circuits including the PLL, clock, and data recovery (CDR) circuit in the USB-PHY block.

CL-only mode limits the transfer to the USB 1.1 specification (full-speed and low-speed transfer only). Power consumption can be reduced by stopping the internal PLL of the PHY module and other high-speed analog circuits.

In CL-only mode, the USBHS requires supply clocks of 48 MHz and 60 MHz, generated in the Clock Generation Circuit. For the clock supply method, see section 8, Clock Generation Circuit.

30.2.18 INTSTS0 : Interrupt Status Register 0

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x040

Bit position:	15	14	13	12	11	10	9	8	7	6			3	2		0
Bit field:	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CTSQ[2:0]	Control Transfer Stage Flag*1 0 0 0: Idle or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error	R
3	VALID	USB Request Reception Flag*1 0: Setup packet not received 1: Setup packet received	R/W*3
6:4	DVSQ[2:0]	Device State*1 Indicates the device state. 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state Others: Suspend state	R
7	VBSTS	VBUS Input Status Flag 0: USBHS_VBUS pin is low 1: USBHS_VBUS pin is high	R

Bit	Symbol	Function	R/W
8	BRDY	BRDY Interrupt Status Flag 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R
9	NRDY	NRDY Interrupt Status Flag 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R
10	BEMP	BEMP Interrupt Status Flag 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R
11	CTRT	Control Transfer Stage Transition Interrupt Status Flag*2 0: No control transfer stage transition interrupt occurred 1: Control transfer stage transition interrupt occurred	R/W*3
12	DVST	Device State Transition Interrupt Status Flag*2 0: No device state transition interrupt occurred 1: Device state transition interrupt occurred	R/W*3
13	SOFR	Frame Number Refresh Interrupt Status Flag 0: No SOF interrupt occurred 1: SOF interrupt occurred	R/W*3
14	RESM	Resume Interrupt Status Flag*2 *4 0: No resume interrupt occurred 1: Resume interrupt occurred	R/W*3
15	VBINT	VBUS Interrupt Status Flag*4 0: No VBUS interrupt occurred on detecting a change in the USBHS_VBUS pin 1: VBUS interrupt occurred on detecting a change in the USBHS_VBUS pin	R/W*3

Note: S-TYPE-3, P-TYPE-3
Note 1. The CTSQ[2:0], VALID, and DVSQ[2:0] flags are only valid in device controller mode.
Note 2. The status of the CTRT, DVST, and RESM flags are changed only in device controller mode. Set the associated interrupt enable bits to 0 (disabled) in host controller mode.
Note 3. To clear the CTRT, DVST, SOFR, RESM, or VBINT flags, write 0 only to the flags to be cleared. Write 1 to the other flags. Do not write 0 to the status flags indicating 0.
Note 4. The USBHS detects a change in the status in the RESM or VBINT flag even while the clock supply is stopped (LPSTS.SUSPENDM = 0), and it requests the interrupt when the associated interrupt request bit is 1. Enable the clock supply before clearing the status by software.

BRDY flag (BRDY Interrupt Status Flag)

The BRDY flag indicates the BRDY interrupt state. For the conditions that cause the flag to be set, see section 30.2.13. BRDYENB : BRDY Interrupt Enable Register.

The USBHS clears the BRDY flag to 0 when 0 is written to the BRDYSTS.PIPEBRDYn (n = 0 to 9) flags for all pipes for which the BRDY interrupt is enabled (BRDYENB.PIPEBRDYEn bits). Writing 0 to the BRDY flag in the software does not clear the flag.

NRDY flag (NRDY Interrupt Status Flag)

The NRDY flag indicates the NRDY interrupt state. For the conditions that cause the flag to be set, see section 30.2.14. NRDYENB : NRDY Interrupt Enable Register.

The USBHS clears the NRDY flag to 0 when 0 is written to the NRDYSTS.PIPENRDYn (n = 0 to 9) flags for all pipes for which the NRDY interrupt is enabled (NRDYENB.PIPENRDYEn bits). Writing 0 to the NRDY flag in the software does not clear the flag.

BEMP flag (BEMP Interrupt Status Flag)

The BEMP indicates the BEMP interrupt state. For the conditions that cause the flag to be set, see section 30.2.15. BEMPENB : BEMP Interrupt Enable Register.

The USBHS clears the BEMP flag to 0 when 0 is written to the BEMPSTS.PIPEBEMPn (n = 0 to 9) flags for all pipes for which the BEMP interrupt is enabled (BEMPENB.PIPEBEMPEn bits). Writing 0 to the BEMP flag in the software does not clear the flag.

CTRT flag (Control Transfer Stage Transition Interrupt Status Flag)

In device controller mode, the USBHS updates the value of the CTSQ[2:0] bits and sets the CTRT flag to 1 on detecting a transition in the control transfer stage. When a control transfer stage transition interrupt occurs, clear the CTRT flag before the USBHS detects the next control transfer stage transition.

Values read from the CTRT flag in host controller mode are invalid.

DVST flag (Device State Transition Interrupt Status Flag)

In device controller mode, the USBHS updates the value of the PL1CTRL1.DVSQ[3:0] bits and sets the DVST flag to 1 on detecting a change in the device state. When a device state transition interrupt occurs, clear the DVST flag before the USBHS detects the next device state transition.

Values read from the DVST flag in host controller mode are invalid.

SOFR flag (Frame Number Refresh Interrupt Status Flag)

In host controller mode, the USBHS sets the SOFR flag to 1 on updating the frame number when the DVSTCTR0.UACT bit is set to 1 by software. An SOFR interrupt is detected every 1 ms.

In device controller mode, the USBHS sets the SOFR flag to 1 on updating the frame number. An SOFR interrupt is detected every 1 ms. The USBHS can detect an SOFR interrupt through the SOF complementation function even when a corrupted SOF packet is received from the USB host. See section 30.3.13. SOF Complementation Function.

RESM flag (Resume Interrupt Status Flag)

In device controller mode, the USBHS sets the RESM flag to 1 on detecting the falling edge of the signal on the USBHS_DP pin in the Suspend state (PL1CTRL1.DVSQ[3:0] = 01xxb).

Values read from the RESM flag in host controller mode are invalid.

VBINT flag (VBUS Interrupt Status Flag)

The USBHS sets the VBINT flag to 1 on detecting a level change (high to low or low to high) in the USBHS_VBUS pin input value. The USBHS sets the VBSTS flag to indicate the USBHS_VBUS pin input value. When a VBINT interrupt occurs, eliminate transient elements by reading the VBSTS flag at least three times through software processing and check that the values read are the same.

30.2.19 INTSTS1 : Interrupt Status Register 1

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x042

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVRCR	BCHG	—	DTCH	ATTCH	—	L1RSMEND	LPMEND	—	EOFERR	SIGN	SACK	—	—	—	PDDETINT
Value after reset:	0	0	x	0	0	x	0	0	x	0	0	0	x	x	x	0

Bit	Symbol	Function	R/W
0	PDDETINT	PDDET Detection Interrupt Status Flag*1 0: No PDDET interrupt occurred 1: PDDET interrupt occurred	R/W*2
3:1	—	The read values are undefined. The write value should be 0.	R/W
4	SACK	Setup Transaction Normal Response Interrupt Status Flag 0: No SACK interrupt occurred 1: SACK interrupt occurred	R/W*2
5	SIGN	Setup Transaction Error Interrupt Status Flag 0: No SIGN interrupt occurred 1: SIGN interrupt occurred	R/W*2
6	EOFERR	EOF Error Detection Interrupt Status Flag 0: No EOFERR interrupt occurred 1: EOFERR interrupt occurred	R/W*2

Bit	Symbol	Function	R/W
7	—	The read value is undefined. The write value should be 0.	R/W
8	LPMEND	LPM Transaction End Interrupt Status Flag 0: No LPMEND interrupt occurred 1: LPMEND interrupt occurred	R/W*2
9	L1RSMEND	L1 Resume End Interrupt Status Flag 0: No L1RSMEND interrupt occurred 1: L1RSMEND interrupt occurred	R/W*2
10	—	The read value is undefined. The write value should be 0.	R/W
11	ATTCH	USB Connection Detection Interrupt Status Flag 0: No ATTCH interrupt occurred 1: ATTCH interrupt occurred	R/W*2
12	DTCH	USB Disconnection Detection Interrupt Status Flag 0: No DTCH interrupt occurred 1: DTCH interrupt occurred	R/W*2
13	—	The read value is undefined. The write value should be 0.	R/W
14	BCHG	USB Bus Change Interrupt Status Flag*1 0: No BCHG interrupt occurred 1: BCHG interrupt occurred	R/W*2
15	OVRCR	OVRCR Interrupt Status Flag*1 0: No OVRCR interrupt occurred 1: OVRCR interrupt occurred	R/W*2

Note: S-TYPE-3, P-TYPE-3
Note: Only enable the status change interrupts indicated in the flags in INTSTS1 in host controller mode, except for the PDDET detection interrupt.
Note 1. The USBHS detects a change in the status in the PDDETINT, BCHG, or OVRCR flag even while the clock supply is stopped (LPSTS.SUSPENDM = 0), and it requests the interrupt when the associated interrupt request bit is 1. Enable the clock supply before clearing the status by software. No other interrupts can be detected while the clock supply is stopped (LPSTS.SUSPENDM = 0).
Note 2. To clear the flags in INTSTS1, write 0 only to the flags to be cleared. Write 1 to the other bits.

PDDETINT flag (PDDET Detection Interrupt Status Flag)

The USBHS sets the PDDETINT flag to 1 on detecting a level change (high to low or low to high) in the PDDET pin input value. When the PDDETINT interrupt is generated, perform debouncing by reading the PDDETSTS flag at least three times through software processing and checking that the values read are the same.

SACK flag (Setup Transaction Normal Response Interrupt Status Flag)

The SACK flag indicates the status of the setup transaction normal response interrupt in host controller mode.

The USBHS detects the SACK interrupt and sets this bit to 1 when an ACK response is returned from a peripheral device during the setup transactions issued by the USBHS. If the associated interrupt enable bit is set to 1 by software, the USBHS generates the interrupt.

Values read from the SACK flag in device controller mode are invalid.

SIGN flag (Setup Transaction Error Interrupt Status Flag)

The SIGN flag indicates the status of setup transaction error interrupts in host controller mode.

The USBHS detects the SIGN interrupt and sets this bit to 1 when an ACK response is not returned from a peripheral device three consecutive times during the setup transactions issued by the USBHS. If the associated interrupt enable bit is set to 1 by software, the USBHS generates the interrupt.

The USBHS detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions:

- Timeout is detected by the USBHS when the peripheral device has returned no response
- A corrupted ACK packet is received
- A handshake other than ACK (NAK, NYET, or STALL) is received

Values read from the SIGN flag in device controller mode are invalid.

EOFERR flag (EOF Error Detection Interrupt Status Flag)

The EOFERR flag indicates the status of EOF error detection interrupts in host controller mode.

The USBHS detects the EOFERR interrupt and sets this bit to 1 on detecting that communication did not complete at the EOF2 timing defined in the USB 2.0 specification. If the associated interrupt enable bit is set to 1 by software, the USBHS generates the interrupt.

After detecting the EOFERR interrupt, the USBHS controls the hardware as follows, regardless of the associated interrupt enable bit setting:

- Sets the DVSTCTR0.UACT bit for the port in which the EOFERR interrupt was detected to 0
- Puts the port in which the EOFERR interrupt occurred into the idle state

The software must terminate all pipes in which communications are being carried out and re-enumerate the USB port.

Values read from the EOFERR flag in device controller mode are invalid.

LPMEND flag (LPM Transaction End Interrupt Status Flag)

The PLPMEND flag indicates the status of LPM transaction end interrupts in host controller mode.

When the HL1CTRL1.L1REQ bit sets to 1, the USBHS sends an LPM token. When the LPM transaction is ended because a response from the function device or a timeout is detected, the USBHS sets this flag to 1.

Values read from the LPMEND flag in device controller mode are invalid.

L1RSMEND flag (L1 Resume End Interrupt Status Flag)

The L1RSMEND flag indicates the status of L1 resume end interrupts in host controller mode.

When performing resume processing after transitioning to the L1 state because an ACK was received in response to an LPM token, the USBHS sets this flag to 1.

Values read from the L1RSMEND flag in device controller mode are invalid.

ATTCH flag (USB Connection Detection Interrupt Status Flag)

The ATTCH flag indicates the status of USB attach detection interrupts in host controller mode.

The USBHS detects the ATTCH interrupt and sets this bit to 1 on detecting a J- or K-state on the full-speed or low-speed signal level for 2.5 µs. If the associated interrupt enable bit is set to 1 by software, the USBHS generates the interrupt.

The USBHS detects the ATTCH interrupt on any of the following conditions:

- K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5 µs
- J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5 µs

Values read from the ATTCH flag in device controller mode are invalid.

DTCH flag (USB Disconnection Detection Interrupt Status Flag)

The DTCH flag indicates the status of USB detach detection interrupts in host controller mode.

The USBHS detects the DTCH interrupt and sets this bit to 1 on detecting a USB bus detach event. If the associated interrupt enable bit is set to 1 by software, the USBHS generates the interrupt.

The USBHS detects bus detach events based on the USB 2.0 specification.

After detecting the DTCH interrupt, the USBHS controls hardware as follows, regardless of the associated interrupt enable bit setting:

- Sets the DVSTCTR0.UACT bit for the port in which the DTCH interrupt was detected to 0
- Puts the port in which the DTCH interrupt occurred into the idle state

The software must terminate all pipes in which communications are being carried out and invoke the wait state for attaching to the USB port (waiting for ATTCH interrupt generation).

Values read from the DTCH flag in device controller mode are invalid.

BCHG flag (USB Bus Change Interrupt Status Flag)

The BCHG flag indicates the status of USB bus change interrupts in host controller mode.

The USBHS detects the BCHG interrupt and sets this bit to 1 when a change in the full-speed signal level occurs on the USB port. This includes any change from J-state, K-state, or SE0 to J-state, K-state, or SE0. If the associated interrupt enable bit is set to 1 by software, the USBHS generates the interrupt.

The USBHS sets the SYSSTS0.LNST[1:0] flags to indicate the input state of the USB port. When a BCHG interrupt occurs, eliminate transient elements by repeat reading the LNST[1:0] bits by software until the same value is read at least three times.

Changes in the USB bus state can be detected while the PHY clock is stopped.

Values read from the BCHG flag in device controller mode are invalid.

OVRCR flag (OVRCR Interrupt Status Flag)

The OVRCR flag indicates the input status on the USBHS_OVCUR0A pin or changes on the USBHS_OVCUR0B pin. If the INTENB1.OVRCRE bit sets to 1, the USBHS requests the interrupt.

The USBHS sets the SYSSTS0.OVCMON[1:0] flags to indicate the input state of the USBHS_OVCUR0A and USBHS_OVCUR0B pins.

These pins allow overcurrent detection by software in host controller mode. To implement this function, connect the overcurrent signal from the external power supply IC that supplies VBUS to connected USB devices to the OVCUR0A or OVCUR0B pin. On detection of an OVRCR interrupt, eliminate transients by repeatedly reading the OVCMON[1:0] flags through the software until the same value is read at least three times.

30.2.20 BRDYSTS : BRDY Interrupt Status Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x046

Bit position:	15	14	13	12	11	10	9									0
Bit field:	—	—	—	—	—	—	PIPEBRDY[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPEBRDY[9:0]	BRDY Interrupt Status Flag for Pipe[9:0]*1 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*2
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Each bit number corresponds to the same pipe number.
Note 2. When the SOFCFG.BRDYM bit is set to 0, to clear the status indicated in the PIPEBRDY[9:0] flags, write 0 only to the bits to be cleared. Write 1 to the other bits.
When the SOFCFG.BRDYM bit is set to 0, clear BRDY interrupts before accessing the FIFO.

PIPEBRDY[9:0] flags (BRDY Interrupt Status Flag for Pipe[9:0])

When the BRDY interrupt is detected, the USBHS sets the associated bit in the PIPEBRDY[9:0] flags to 1. For details on BRDY interrupts, see section 30.3.6.1. BRDY interrupt.

30.2.21 NRDYSTS : NRDY Interrupt Status Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x048

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPENRDY[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPENRDY[9:0]	NRDY Interrupt Status Flag for Pipe[9:0]*1 0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R/W*2
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Each bit number corresponds to the same pipe number.
Note 2. To clear the status indicated in the PIPENRDY[9:0] flags, write 0 only to the bits to be cleared. Write 1 to the other bits.

PIPENRDY[9:0] flags (NRDY Interrupt Status Flag for Pipe[9:0])

If an internal NRDY interrupt is detected while the PID[1:0] bits in a pipe control register are 01b (BUF response), the USBHS sets the associated bit in the PIPENRDY[9:0] flags to 1. For details on NRDY interrupts, see section 30.3.6.2. NRDY interrupt.

30.2.22 BEMPSTS : BEMP Interrupt Status Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x04A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPEBEMP[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPEBEMP[9:0]	BEMP Interrupt Status Flag for Pipe[9:0]*1 0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R/W*2
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Each bit number corresponds to the same pipe number.
Note 2. To clear the status indicated in the PIPEBEMP[9:0] flags, write 0 only to the bits to be cleared. Write 1 to the other bits.

PIPEBEMP[9:0] flags (BEMP Interrupt Status Flag for Pipe[9:0])

If an BEMP interrupt is detected while the PID[1:0] bits in a pipe control register are 01b (BUF response), the USBHS sets the associated bit in the PIPEBEMP[9:0] flags to 1. For details on BEMP interrupts, see section 30.3.6.3. BEMP interrupt.

30.2.23 FRMNUM : Frame Number Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x04C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVRN	CRCE	—	—	—	FRNM[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	FRNM[10:0]*1	Frame Number Flag	R
13:11	—	These bits are read as 0. The write value should be 0.	R/W
14	CRCE	CRC Error Detection Status Flag 0: No error occurred 1: Error occurred	R/W
15	OVRN	Overrun/Underrun Detection Status Flag 0: No error occurred 1: Error occurred.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The OVRN flag is for debugging. Design the timing so that no overrun or underrun occurs in the system.

FRNM[10:0] flags (Frame Number Flag)

The USBHS sets the FRNM[10:0] flags to indicate the latest frame number, which is updated every 1 ms, when an SOF packet is issued or received.

CRCE flag (CRC Error Detection Status Flag)

The CRCE flag sets to 1 when a CRC error or bit stuffing error occurs during isochronous transfer. On detecting a CRC error, the USBHS generates an internal NRDY interrupt.

To clear the CRCE flag, write 0 to it while writing 1 to the other bits in the FRMNUM register.

OVRN flag (Overrun/Underrun Detection Status Flag)

The OVRN flag sets to 1 when an overrun or underrun error occurs during isochronous transfer. To clear the flag, write 0 to it while writing 1 to the other bits in the FRMNUM register.

In host controller mode, the OVRN flag sets to 1 on any of the following conditions:

- For a transmitting isochronous pipe, the time to issue an OUT token comes before all of the transmit data is written to the FIFO buffer
- For a receiving isochronous pipe, the time to issue an IN token comes when no FIFO buffer planes are empty

In device controller mode, the OVRN flag sets to 1 on any of the following conditions:

- For a transmitting isochronous pipe, the IN token is received before all of the transmit data is written to the FIFO buffer
- For a receiving isochronous pipe, the OUT token is received when no FIFO buffer planes are empty

30.2.24 UFRMNUM : µFrame Number Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x04E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DVCHG	—	—	—	—	—	—	—	—	—	—	—	—	UFRNM[2:0]		
Value after reset:	0	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	UFRNM[2:0]	Microframe number	R
14:3	—	These bits are read as 0. The write value should be 0.	R/W
15	DVCHG	Device State Change 0: Disable writes to the USBADDR.STSRECOV0[2:0] and USBADDR.USBADDR [6:0] bits 1: Enable writes to the USBADDR.STSRECOV0[2:0] and USBADDR.USBADDR[6:0] bits	R/W

Note: S-TYPE-3, P-TYPE-3

UFRNM[2:0] flags (Microframe number)

The USBHS sets the UFRNM[2:0] flags to indicate the microframe number during high-speed operation. When not in high-speed operation, the USBHS sets these bits to 00b.

Read these bits repeatedly until the same value is read twice.

30.2.25 USBADDR : USB Address Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x050

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	STSRECOV0[2:0]			—	USBADDR[6:0]						
Value after reset:	x	x	x	x	x	0	0	0	x	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	USBADDR[6:0]	USB Address Flag In device controller mode, these flags indicate the USB address assigned by the host when the USBHS processed the SET_ADDRESS request successfully.	R
7	—	The read value is undefined. The write value should be 0.	R/W
10:8	STSRECOV0[2:0]	Status Recovery [D]: In deice controller mode [H]: In host controller mode (settings other than 010b, 100b, or 110b are prohibited) 0 0 0: Reserved 0 0 1: [D] Return to the full-speed connection and Default state 0 1 0: [D] Return to the full-speed connection and Address state [H] Return to the low-speed state (bits DVSTCTR0.RHST[2:0] = 001b) 0 1 1: [D] Return to the full-speed connection and Configured state 1 0 0: [D] Return to the suspend connection and Suspend state [H] Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b) 1 0 1: [D] Return to the high-speed connection and Default state 1 1 0: [D] Return to the high-speed connection and Address state [H] Return to the high-speed state (bits DVSTCTR0.RHST[2:0] = 011b) 1 1 1: [D] Return to the high-speed connection and Configured state	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

USBADDR[6:0] flags (USB Address Flag)

In device controller mode, the USBADDR[6:0] flags indicate the USB address received when the USBHS processed a SetAddress request successfully. The USBHS sets the USBADDR[6:0] bits to 00b on detecting a USB bus reset.

In host controller mode, the USBADDR[6:0] bits are invalid.

STSRECOV0[2:0] bits (Status Recovery)

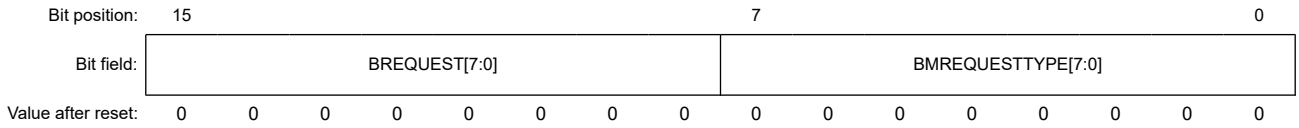
Use the STSRECOV[3:0] bits to resume the state of the internal sequencer on recovering from USB power shut-off. For details, see section 30.3.17. Deep Software Standby Mode 1 Because of USB Suspend/Resume Interrupts.

Writing to these bits is enabled while the UFRMNUM.DVCHG bit is set to 1.

30.2.26 USBREQ : USB Request Type Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x054



Bit	Symbol	Function	R/W
7:0	BMREQUESTTYPE [7:0]	USB request bmRequestType value	R/W*1
15:8	BREQUEST[7:0]	USB request bRequest value	R/W*1

Note: S-TYPE-3, P-TYPE-3
Note 1. In device controller mode, these bits can be read, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

BMREQUESTTYPE[7:0] bits (USB request bmRequestType value)

The BMREQUESTTYPE[7:0] bits hold the bmRequestType value of USB requests.

- In host controller mode:
  Set these bits to the value of the USB request data in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the value of the USB request data in reception setup transactions. Writing to the bits has no effect.

BREQUEST[7:0] bits (USB request bRequest value)

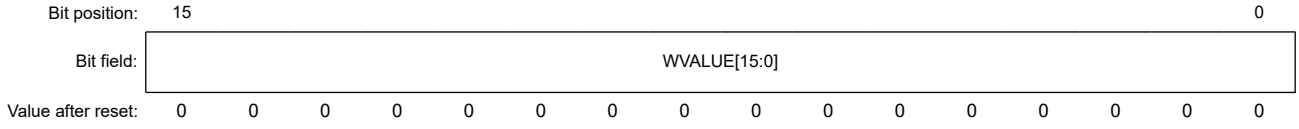
The BREQUEST[7:0] bits hold the bRequest value of USB requests.

- In host controller mode:
  Set these bits to the value of the USB request data in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the value of the USB request data in reception setup transactions. Writing to the bits has no effect.

30.2.27 USBVAL : USB Request Value Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x056



Bit	Symbol	Function	R/W
15:0	WVALUE[15:0]	USB request wValue value	R/W*1

Note: S-TYPE-3, P-TYPE-3
Note 1. In device controller mode, these bits are readable, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

WVALUE[15:0] bits (USB request wValue value)

The WVALUE[15:0] bits hold the wValue value of USB requests.

- In host controller mode:

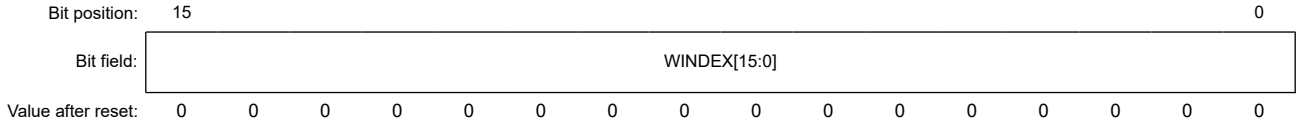
Set these bits to the wValue value for USB requests in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.

- In device controller mode:
  These bits indicate the wValue value of USB requests in reception setup transactions. Writing to the bits has no effect.

30.2.28 USBINDX : USB Request Index Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x058



Bit	Symbol	Function	R/W
15:0	WINDEX[15:0]	USB request wIndex value	R/W*1

Note: S-TYPE-3, P-TYPE-3
Note 1. In device controller mode, these bits are readable, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

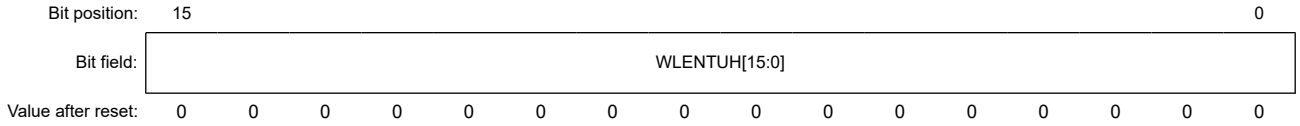
WINDEX[15:0] bits (USB request wIndex value)

- In host controller mode:
  Set these bits to the wIndex value of USB requests in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the wIndex value of USB requests received in reception setup transactions. Writing to the bits has no effect.

30.2.29 USBLENG : USB Request Length Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x05A



Bit	Symbol	Function	R/W
15:0	WLENTUH[15:0]	USB request wLength value	R/W*1

Note: S-TYPE-3, P-TYPE-3
Note 1. In device controller mode, these bits are readable, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

WLENTUH[15:0] bits (USB request wLength value)

The WLENTUH[15:0] bits hold the wLength value of USB requests.

- In host controller mode:
  Set the wLength value of USB requests in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the wLength value of USB requests in reception setup transactions. Writing to the bits has no effect.

30.2.30 DCPCFG : DCP Configuration Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x05C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CNTMD	SHTNAK	—	—	DIR	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	The read values are undefined. The write value should be 0.	R/W
4	DIR	Transfer Direction 0: Data receiving direction 1: Data transmitting direction	R/W
6:5	—	The read values are undefined. The write value should be 0.	R/W
7	SHTNAK	Pipe Blocking on End of Transfer 0: Keep pipe open after transfer ends 1: Disable pipe after transfer ends	R/W
8	CNTMD	Continuous Transfer Mode 0: Non-continuous transfer mode 1: Continuous transfer mode	R/W
15:9	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Only set the bits in the DCPCFG register while the PID is NAK. Before setting the bits, check that the DCPCTR.PBUSY bit is 0, and then change the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBHS, checking the PBUSY bit through software is not necessary.

DIR bit (Transfer Direction)

In host controller mode, the DIR bit sets the transfer direction of the data stage and status stage for control transfers. In device controller mode, set the DIR bit to 0.

SHTNAK bit (Pipe Blocking on End of Transfer)

The SHTNAK bit specifies whether to change PID to NAK on transfer end when the selected pipe is receiving. It is only valid when the selected pipe is receiving.

When the SHTNAK bit is 1, the USBHS changes the DCPCTR.PID[1:0] bits for the DCP to NAK on determining that a transfer has ended. The USBHS determines transfer end on the following condition:

- A short packet, including a zero-length packet, is successfully received

CNTMD bit (Continuous Transfer Mode)

The CNTMD bit indicates whether transfer through the default control pipe is in continuous transfer mode.

30.2.31 DCPMAXP : DCP Maximum Packet Size Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x05E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DEVSEL[3:0]				—	—	—	—	—	MXPS[6:0]						
Value after reset:	0	0	0	0	x	x	x	x	x	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	MXPS[6:0]	Maximum Packet Size*1 Maximum data payload specification (maximum packet size) for the DCP	R/W
11:7	—	The read values are undefined. The write value should be 0.	R/W
15:12	DEVSEL[3:0]	Device Select*2 0x0: Address 0x0 0x1: Address 0x1 0x2: Address 0x2 0x3: Address 0x3 0x4: Address 0x4 0x5: Address 0x5	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Only set the MXPS[6:0] bits while PID is NAK. Before setting this bit, check that the CSSTS and PBUSY bits are 0, and then change the DCPCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK), and the CFIFOSEL.CURPIPE[3:0] bits to 0000b. If the DCPCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the CSSTS and PBUSY bits through software is not necessary. After the MXPS[6:0] bits are set and the DCP is set to the CURPIPE[3:0] bits in a port select register, clear the buffer by setting the BCLR bit the port control register to 1.

Note 2. Only set the DEVSEL[3:0] bits while PID is NAK and the DCPCTR.SUREQ bits are 0. Before setting these bits, check that the CSSTS and PBUSY flags are 0, and then change the DCPCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK), and the DCPCTR.SUREQ bits to 0. If the DCPCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the CSSTS and PBUSY bits through software is not necessary.

MXPS[6:0] bits (Maximum Packet Size)

The MXPS[6:0] bits specify the maximum data payload (maximum packet size) for the DCP. The initial value is 0x40 (64 bytes). Set the bits to a USB 2.0-compliant value. Do not write to the FIFO buffer or set PID = BUF while MXPS[6:0] is set to 0.

DEVSEL[3:0] bits (Device Select)

In host controller mode, the DEVSEL[3:0] bits specify the address of the target peripheral device for a control transfer. Set up the associated DEVADDn (n = 0 to A) register first, and then set these bits to the corresponding value. To set the DEVSEL[3:0] bits to 0010b, for example, first set the DEVADD2 register. In device controller mode, set these bits to 0000b.

30.2.32 DCPCTR : DCP Control Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x060

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSTS	SUREQ	CSCLR	CSSTS	SUREQCLR	—	—	SQCLR	SQSET	SQMON	PBUSY	PINGE	—	CCPL	PID[1:0]	
Value after reset:	0	0	0	0	x	x	x	0	0	1	0	0	x	0	0	0

Bit	Symbol	Function	R/W
1:0	PID[1:0]	Response PID 0 0: NAK response 0 1: BUF response (depends on buffer state) 1 0: STALL response 1 0: STALL response	R/W
2	CCPL	Control Transfer End Enable 0: Disable control transfer completion 1: Enable control transfer completion	R/W
3	—	The read value is undefined. The write value should be 0.	R/W
4	PINGE	PING Token Issue Enable*1 0: Disable PING token 1: Enable normal PING operation	R/W

Bit	Symbol	Function	R/W
5	PBUSY	Pipe Busy Flag 0: DCP not used for the USB bus 1: DCP in use for the USB bus	R
6	SQMON	Sequence Toggle Bit Monitor Flag 0: DATA0 1: DATA1	R
7	SQSET	Sequence Toggle Bit Set*1 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA1	W
8	SQCLR	Sequence Toggle Bit Clear*1 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0	W
10:9	—	The read values are undefined. The write value should be 0.	R/W
11	SUREQCLR	SUREQ Bit Clear 0: Invalid (writing 0 has no effect) 1: Clear SUREQ to 0	W
12	CSSTS	CSSTS Status Flag 0: Start-split (SSPLIT) transaction, or processing for devices that are not using split transactions, in progress 1: Complete-split (CSPLIT) transaction in progress	R
13	CSCLR	CSSTS Status Flag Clear 0: (writing 0 has no effect) 1: Clear CSSTS to 0	W
14	SUREQ	SETUP Token Transmission 0: Invalid (writing 0 has no effect) 1: Transmit setup packet	R/W
15	BSTS	Buffer Status Flag 0: Buffer access disabled 1: Buffer access enabled	R

Note: S-TYPE-3, P-TYPE-3
Note 1. Only set the SQSET, SQCLR, and PINGE bits while PID is NAK. Before setting these bits, check that the CSSTS and PBUSY bits are 0, and then change the DCPCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the DCPCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the CSSTS and PBUSY bits through the software is not necessary.

PID[1:0] bits (Response PID)

The PID[1:0] bits control the USB response type during control transfers.

In host controller mode, to change the PID[1:0] setting from NAK to BUF:

- When the transmitting direction is set:
  a. Write all of the transmit data to the FIFO buffer while the DVSTCTR0.UACT bit is 1 and PID is NAK.
  b. Set PID[1:0] bits to 01b (BUF).
  The USBHS then executes the OUT transaction (or PING transaction)
- When the receiving direction is set:
  a. Check that the FIFO buffer is empty (or empty the buffer) while the DVSTCTR0.UACT bit is 1 and PID is NAK.
  b. Set PID[1:0] bits to 01b (BUF).
  The USBHS then executes the IN transaction.

The USBHS changes the PID[1:0] setting as follows:

- When the PID[1:0] bits are set to BUF (01b) by software and the USBHS has received data exceeding MaxPacketSize, the USBHS sets PID[1:0] to STALL (11b)
- When a reception error, such as a CRC error, is detected three times consecutively, the USBHS sets PID[1:0] to NAK (00b)
- On receiving the STALL handshake, the USBHS sets PID[1:0] to STALL (11b)

In device controller mode, the USBHS changes the PID[1:0] setting as follows:

- On receiving a setup packet, the USBHS sets PID[1:0] to NAK (00b). The USBHS then sets the INTSTS0.VALID flag to 1, and the PID[1:0] setting cannot be changed until the software clears the VALID flag to 0.
- When the PID[1:0] bits are set to BUF (01b) by software and the USBHS has received data exceeding MaxPacketSize, the USBHS sets PID[1:0] to STALL (11b)
- On detecting a control transfer sequence error, the USBHS sets PID[1:0] to STALL (1xb)
- On detecting a USB bus reset, the USBHS sets PID[1:0] to NAK

The USBHS does not check the PID[1:0] setting while processing a SET_ADDRESS request.

CCPL bit (Control Transfer End Enable)

In device controller mode, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed. When the bit is set to 1 by software while the associated PID[1:0] bits are set to BUF, the USBHS completes the control transfer status stage.

During control read transfers, the USBHS transmits the ACK handshake in response to the OUT transaction from the USB host. During control write or no-data control transfers, it transmits the zero-length packet in response to the IN transaction from the USB host. On detecting a SET_ADDRESS request, the USBHS operates in auto response mode from the setup stage up to status stage completion regardless of the CCPL bit setting.

The USBHS changes the CCPL bit from 1 to 0 on receiving a new setup packet. The software cannot write 1 to the bit while the INTSTS0.VALID bit is 1. The bit is initialized by a USB bus reset.

In host controller mode, always write 0 to the CCPL bit.

PINGE bit (PING Token Issue Enable)

In host controller mode, when the software sets the PINGE bit to 1, the USBHS issues a PING token for transfer in the transmitting direction, which triggers the transfer to start. If an ACK handshake is detected in the PING transaction, the OUT transaction is executed in the next transaction. If a NAK or NYET handshake is detected in the OUT transaction, the PING transaction is executed in the next transaction.

If the software sets this bit to 0, the USBHS issues no PING token for transfer in the transmitting direction. All transfers in the transmitting direction are executed in the OUT transaction.

PBUSY flag (Pipe Busy Flag)

The PBUSY bit indicates whether DCP is used for the transaction when USBHS changes the PID[1:0] bits from BUF to NAK. The USBHS changes the PBUSY flag from 0 to 1 on start of a USB transaction for the selected pipe. It changes the PBUSY flag from 1 to 0 on completion of one transaction.

After PID is set to NAK by software, the value in the PBUSY flag indicates whether changes to pipe settings can proceed.

For details, see section 30.3.7.1. Pipe control register switching procedures.

SQMON flag (Sequence Toggle Bit Monitor Flag)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction during a DCP transfer.

The USBHS toggles the bit on normal completion of the transaction. It does not toggle the bit, however, when a DATAPID mismatch occurs during a transfer in the receiving direction.

In device controller mode, the USBHS sets the SQMON bit to 1 (specifies DATA1 as the expected value) on successful reception of the setup packet.

In device controller mode, the USBHS does not reference this bit during IN or OUT transactions at the status stage, and it does not toggle the bit on normal completion.

SQSET bit (Sequence Toggle Bit Set)

The SQSET bit specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during a DCP transfer.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

SQCLR bit (Sequence Toggle Bit Clear)

The SQCLR bit specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during a DCP transfer. It is read as 0.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

SUREQCLR bit (SUREQ Bit Clear)

In host controller mode, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The bit is read as 0.

If transfer stops while the SUREQ bit is set to 1 in a setup transaction, set the SUREQCLR bit to 1 through software. This is not necessary at the end of a normal setup transaction, because the USBHS automatically clears the SUREQ bit to 0.

Only control the SUREQ bit through the SUREQCLR bit while the DVSTCTR0.UACT bit is 0. When UACT is 0, communication is halted or no transfer is occurring because a bus disconnection was detected.

In device controller mode, always write 0 to the SUREQCLR bit.

CSSTS flag (CSSTS Status Flag)

In host controller mode, the CSSTS flag indicates the complete-split state in split transactions for pipes that are not isochronous. The USBHS sets the CSSTS flag to 1 at the beginning of a complete-split transaction and sets the flag back to 0 when it detects transaction completion.

Values read from the CSSTS flag in device controller mode are invalid.

CSCLR bit (CSSTS Status Flag Clear)

In host controller mode, setting the CSCLR bit to 1 clears the CSSTS bit to 0.

Set this bit to 1 through software when forcing the next transfer to restart from start-split in transfers using split transactions. This is not necessary at the end of a successful complete-split transaction in a normal split transaction, because the USBHS automatically clears the CSSTS flag to 0.

Only control the CSSTS flag through the CSCLR bit while the DVSTCTR0.UACT bit is 0. When UACT is 0, communication is halted or no transfer is occurring because a port disconnection was detected. Writing 1 to this bit while the CSSTS flag is 0 has no effect; the flag remains 0.

In device controller mode, always write 0 to this bit.

SUREQ bit (SETUP Token Transmission)

In host controller mode, setting the SUREQ bit to 1 triggers the USBHS to transmit the setup packet. After completing the setup transaction process, the USBHS generates either the SACK or SIGN interrupt and clears the SUREQ bit to 0. The USBHS also clears the SUREQ bit to 0 when the software sets the SUREQCLR bit to 1.

Before setting the SUREQ bit to 1, set the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, and USBLENG appropriately to transmit the wanted USB request in the setup transaction. Also check that the PID[1:0] bits for the DCP are set to NAK. After setting the SUREQ bit to 1, do not change the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, or USBLENG until the setup transaction is complete (SUREQ bit = 1). Write 1 to the SUREQ bit only when transmitting the setup token. Otherwise, write 0.

In device controller mode, always write 0 to this bit.

BSTS flag (Buffer Status Flag)

The BSTS flag indicates the status of access to the DCP FIFO buffer. The meaning of this flag varies as follows depending on the CFIFOSEL.ISEL setting:

- When ISEL = 0, the bit indicates whether receive data can be read from the buffer
- When ISEL = 1, the bit indicates whether transmit data can be written to the buffer

30.2.33 PIPESEL : Pipe Window Select Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x064

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PIPESEL[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	PIPESEL[3:0]	Pipe Window Select 0x0: No pipe selected 0x1: Pipe 1 0x2: Pipe 2 0x3: Pipe 3 0x4: Pipe 4 0x5: Pipe 5 0x6 Pipe 6 0x7: Pipe 7 0x8: Pipe 8 0x9: Pipe 9 Others: Setting prohibited	R/W
15:4	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Set pipes and functions using the PIPESEL, PIPECFG, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEmTRE, and PIPEmTRN registers (n = 1 to 9, m = 1 to 5).

The pipe is selected by PIPESEL.

After selecting the pipe in the PIPESEL register, pipe functions must be set in the associated PIPECFG, PIPEMAXP, and PIPEPERI registers. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set independently of the pipe selection in this register.

PIPESEL[3:0] bits (Pipe Window Select)

The PIPESEL[3:0] bits select the pipe number associated with the PIPECFG, PIPEMAXP, and PIPEPERI registers used for data writing and reading. Selecting a pipe number in the PIPESEL[3:0] bits allows writing to and reading from PIPECFG, PIPEMAXP, and PIPEPERI associated with the selected pipe number.

When PIPESEL[3:0] = 0x0, 0 is read from all of the bits in PIPECFG, PIPEMAXP, and PIPEPERI. Writing to these bits has no effect.

30.2.34 PIPECFG : Pipe Configuration Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x068

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3			0
Bit field:	TYPE[1:0]		—	—	—	BFRE	DBLB	CNTMD	SHTNAK	—	—	DIR	EPNUM[3:0]			
Value after reset:	0	0	x	x	x	0	0	0	0	x	x	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	EPNUM[3:0]	Endpoint Number*1 Specifies the endpoint number for the selected pipe. Setting 0x0 indicates the pipe is not used.	R/W

Bit	Symbol	Function	R/W
4	DIR	Transfer Direction*2 *3 0: Receiving direction 1: Transmitting direction	R/W
6:5	—	The read values are undefined. The write value should be 0.	R/W
7	SHTNAK	Pipe Disabled at End of Transfer*1 0: Continue pipe operation after transfer ends 1: Disable pipe after transfer ends	R/W
8	CNTMD	Continuous Transfer Mode*2 *3 0: Discontinuous transfer mode 1: Continuous transfer mode	R/W
9	DBLB	Double Buffer Mode*2 *3 0: Single buffer 1: Double buffer	R/W
10	BFRE	BRDY Interrupt Operation Specification*2 *3 0: Generate BRDY interrupt on transmitting or receiving data 1: Generate BRDY interrupt on completion of reading data	R/W
13:11	—	The read values are undefined. The write value should be 0.	R/W
15:14	TYPE[1:0]	Transfer Type*1 0 0: Pipe not used 0 1: (Pipe 1 to 5) Bulk transfer (Pipe 6 to 9) Setting prohibited 1 0: (Pipe 1 to 5) Setting prohibited (Pipe 6 to 9) Interrupt transfer 1 1: (Pipe 1 to 2) Isochronous transfer (Pipe 3 to 9) Setting prohibited	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Only set the TYPE[1:0], SHTNAK, and EPNUM[3:0] bits while PID is NAK. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the CSSTS and PBUSY flags through the software is not necessary.

Note 2. Only set the BFRE, DBLB, CNTMD, and DIR bits while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the PBUSY flag through the software is not necessary.

Note 3. To change the BFRE, DBLB, CNTMD, or DIR bit after completing USB communication on the selected pipe, in addition to the constraints described in note 2, write 1 and 0 to the PIPEnCTR.ACLRM bit continuously through software and clear the FIFO buffer assigned to the pipe.

EPNUM[3:0] bits (Endpoint Number)

The EPNUM[3:0] bits specify the endpoint number for the selected pipe. Setting 0000b indicates the pipe not used.

Set these bits so that the combination of the DIR and EPNUM[3:0] settings is different from those for other pipes. (The EPNUM[3:0] bits can be set to 0000b for all pipes.)

DIR bit (Transfer Direction)

The DIR bit specifies the transfer direction for the selected pipe.

When the software sets this bit to 0, the USBHS uses the selected pipe for receiving. When the software sets this bit to 1, the USBHS uses the selected pipe for transmitting.

SHTNAK bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to change the PIPEnCTR.PID[1:0] bits to 00b (NAK) at the end of transfer when the selected pipe is set in the receiving direction. The bit is valid for pipes 1 to 5 in the receiving direction.

When the software sets this bit to 1 for a receiving pipe, the USBHS changes the associated PIPEnCTR.PID[1:0] bits to 00b (NAK) on determining the transfer end. The USBHS determines that the transfer has ended on the following conditions:

- Short packet data (including a zero-length packet) was successfully received
- The transaction counter is used and the number of packets specified for the transaction counter were successfully received

CNTMD bit (Continuous Transfer Mode)

The CNTMD bit specifies whether to operate the selected pipe in continuous transfer mode. The bit is valid for pipes 1 to 5 of the bulk transfer type.

Based on this bit setting, the USBHS determines the completion of transmission or reception for the FIFO buffer allocated to the selected pipe as shown in Table 30.9.

Table 30.9 Relationship between the CNTMD setting and methods for determining completion of FIFO buffer transmission or reception

CNTMD bit setting	Methods for determining readable state and transmittable state
0	Condition for FIFO buffer readable state in receiving direction (DIR = 0): ● The USBHS received one packet
	Conditions for FIFO buffer transmittable state in transmitting direction (DIR = 1): When one of the following is satisfied: 1. CPU or DMAC/DTC wrote data of the maximum packet size to the FIFO buffer 2. CPU or DMAC/DTC wrote data of the short packet size (including 0 bytes) to the FIFO buffer and set the BVAL flag in the port control register to 1
1	Condition for FIFO buffer readable state in receiving direction (DIR = 0): 1. The byte count of data received in the FIFO buffer allocated to the selected pipe is equal to the allocated byte count ((BUFSIZE + 1) × 64). 2. The USBHS received a short packet, other than a zero-length packet. 3. The USBHS received a zero-length packet when data was already contained in the FIFO buffer allocated to the selected pipe. 4. Software received the number of packets specified for the transaction counter set for the selected pipe.
	Conditions for FIFO buffer transmittable state in transmitting direction (DIR = 1): When one of the following is satisfied. 1. The amount of data written by CPU or DMAC/DTC is equal to the size of the FIFO buffer allocated to the selected pipe. 2. CPU or DMAC/DTC wrote data of smaller size than that of the FIFO buffer allocated to the selected pipe (including 0 bytes) and set the BVAL flag in the port control register to 1. 3. CPU or DMAC/DTC wrote data of smaller size than that of one FIFO buffer allocated to the selected pipe (including 0 bytes) and asserted the DENDx_N signal on the last write.

DBLB bit (Double Buffer Mode)

The DBLB bit selects either single or double buffer mode for the FIFO buffer used by the selected pipe. The bit is valid for pipes 1 to 5.

When the software sets this bit to 1, the USBHS allocates twice the FIFO buffer size specified in the PIPEBUF.BUFSIZE[5:0] bits for the selected pipe. The FIFO buffer size that the USBHS allocates to the selected pipe is as follows:

(BUFSIZE + 1) × 64 × (DBLB + 1) [bytes]

BFRE bit (BRDY Interrupt Operation Specification)

The BFRE bit specifies the BRDY interrupt generation timing from the USBHS to the CPU for the selected pipe.

When the software sets the BFRE bit to 1 and the selected pipe is in the receiving direction, the USBHS detects the transfer completion and generates the BRDY interrupt on reading the packet.

When a BRDY interrupt is generated with this setting, the software must write 1 to the BCLR bit in the port control register. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit.

When the BFRE bit is set to 1 by software and the selected pipe is in the transmitting direction, the USBHS does not generate the BRDY interrupt. For details, see section 30.3.6.1. BRDY interrupt.

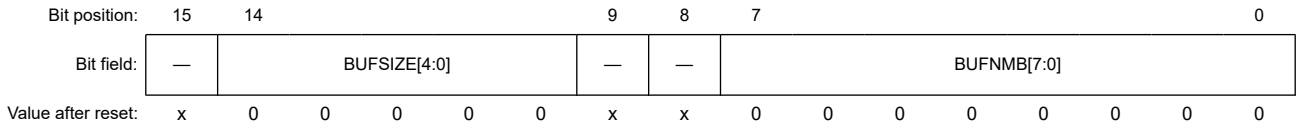
TYPE[1:0] bits (Transfer Type)

The TYPE[1:0] bits specify the transfer type for the pipe selected in the PIPESEL.PIPESEL[3:0] bits. Before setting PID to BUF and starting USB communication on the selected pipe, set the TYPE[1:0] bits to a value other than 00b.

30.2.35 PIPEBUF : Pipe Buffer Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x06A



Bit	Symbol	Function	R/W
7:0	BUFNMB[7:0]	Buffer Number Specifies the FIFO buffer number of the selected pipe (0x04 to 0x87).	R/W
9:8	—	The read values are undefined. The write value should be 0.	R/W
14:10	BUFSIZE[4:0]	Buffer Size 0x00: 64 bytes 0x01: 128 bytes ⋮ 0x1F: 2 KB	R/W
15	—	The read value is undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Only set the bits in the PIPEBUF register while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the CSSTS and PBUSY flags through the software is not necessary.

BUFNMB[7:0] bits (Buffer Number)

The BUFNMB[7:0] bits specify the first block number of the FIFO buffer to be allocated to the selected pipe.

The USBHS allocates the FIFO buffer blocks to the selected pipe as follows:

Block number: BUFNMB to block number: BUFNMB + (BUFSIZE + 1) × (DBLB + 1) - 1

Set a value within the memory size range for these bits (0 [0x00] to 8640 [0x87] for 8.5 KB), while observing the following conditions:

- 0x00 is for DCP only
- 0x04 is for pipe 6 only, but is available for other pipes when pipe 6 is not used. When pipe 6 is selected, writes to these bits are disabled. The USBHS automatically allocates 0x04 to the BUFNMB bits for pipe 6.
- 0x05 is for pipe 7 only, but is available for other pipes when pipe 7 is not used. When pipe 7 is selected, writes to these bits are disabled. The USBHS automatically allocates 0x05 to the BUFNMB bits for pipe 7.
- 0x06 is for pipe 8 only, but is available for other pipes when pipe 8 is not used. When pipe 8 is selected, writes to these bits are disabled. The USBHS automatically allocates 0x06 to the BUFNMB bits for pipe 8.
- 0x07 is for pipe 9 only, but is available for other pipes when pipe 9 is not used. When pipe 9 is selected, writes to these bits are disabled. The USBHS automatically allocates 0x07 to the BUFNMB bits for pipe 9.

BUFSIZE[4:0] bits (Buffer Size)

The BUFSIZE[4:0] bits specify the FIFO buffer size (number of blocks) to be allocated to the selected pipe. One block is 64 bytes.

When the software sets the DBLB bit to 1, the USBHS allocates twice the FIFO buffer size specified in these bits to the selected pipe. The DBLB = 1 setting is valid for pipes 1 to 5.

The USBHS allocates the FIFO buffer blocks to the selected pipe as follows:

(BUFSIZE + 1) × 64 × (DBLB + 1) [bytes]

Set the value within the following range:

- For pipes 1 to 5, set a value from 0x00 to 0x1F (up to 2 KB)

- For pipes 6 to 9, only set a value of 0x00 (64 bytes)

30.2.36 PIPEMAXP : Pipe Maximum Packet Size Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x06C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DEVSEL[3:0]				—	MXPS[10:0]										
Value after reset:	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	MXPS[10:0]*1 *2	Maximum Packet Size ● Pipes 1 and 2 1 byte (0x001) to 1024 bytes (0x400) ● Pipes 3 to 5 8 bytes (0x008), 16 bytes (0x010), 32 bytes (0x020), 64 bytes (0x040), 512 bytes (0x200) (Bits 2 to 0 not supported.) ● Pipes 6 to 9 1 byte (0x001) to 64 bytes (0x040) (Bits 10 to 7 not supported.)	R/W
11	—	The read value is undefined. The write value should be 0.	R/W
15:12	DEVSEL[3:0]*3	Device Select 0x0: Address 0x0 0x1: Address 0x1 ⋮ 0x9: Address 0x9 0xA: Address 0xA Others: Reserved	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. The initial value of the MXPS[10:0] bits is 0x00 when no pipe is selected in the PIPESEL.PIPESEL[3:0] bits and 0x40 when a pipe is selected.

Note 2. Only set the MXPS[10:0] bits while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the CSSTS and PBUSY flags through the software is not necessary.

Note 3. Only set the DEVSEL[3:0] bits while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the PBUSY flag through the software is not necessary.

MXPS[10:0] bits (Maximum Packet Size)

The MXPS[10:0] bits specify the maximum data payload (maximum packet size) for the selected pipe.

Set these bits to the appropriate value for each transfer type based on the USB 2.0 specification. When MXPS[10:0] = 0, do not write to the FIFO buffer or set PID to BUF. These writes have no effect.

To communicate on an isochronous pipe using a split transaction, set the value in the MXPS[10:0] bits to 188 bytes or less.

DEVSEL[3:0] bits (Device Select)

In host controller mode, the DEVSEL[3:0] bits specify the address of the target device for USB communication. Set up the associated DEVADDn (n = 0 to A) register first, and then set these bits to the corresponding value. To set the DEVSEL[3:0] bits to 0x2, for example, first set the DEVADD2 register.

In device controller mode, set these bits to 0x0.

30.2.37 PIPEPERI : Pipe Cycle Control Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x06E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2		0
Bit field:	—	—	—	IFIS	—	—	—	—	—	—	—	—	—	IITV[2:0]		
Value after reset:	x	x	x	0	x	x	x	x	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	IITV[2:0]*1	Interval Error Detection Interval Specifies the interval error detection timing for the selected pipe as the n-th power of 2 of the frame timing.	R/W
11:3	—	The read values are undefined. The write value should be 0.	R/W
12	IFIS	Isochronous IN Buffer Flush 0: Do not flush buffer 1: Flush buffer	R/W
15:13	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Only set the IITV[2:0] bits while PID is NAK. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the PBUSY flag through the software is not necessary.

PIPEPERI selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfers, and sets the interval error detection interval for pipes 1 to 9.

IITV[2:0] bits (Interval Error Detection Interval)

To change the IITV[2:0] bits to another value after they are set and USB communication is performed, set the PIPEnCTR.PID[1:0] bits to 00b (NAK) and then set the PIPEnCTR.ACLRM bit to 1 to initialize the interval timer.

The IITV[2:0] bits are not provided for pipes 3 to 5. Write 000b to bit positions of the IITV[2:0] bits associated with pipes 3 to 5.

IFIS bit (Isochronous IN Buffer Flush)

The IFIS bit specifies whether to flush the buffer when the pipe specified in the PIPESEL.PIPESEL[3:0] bits is used for isochronous IN transfers.

In device controller mode when the selected pipe is for isochronous IN transfers, the USBHS automatically clears the FIFO buffer if the USBHS fails to receive the IN token from the USB host within the interval set in the IITV[2:0] bits in terms of frames.

When double buffering is specified (PIPECFG.DBLB = 1), the USBHS only clears the data in the previously used plane.

The USBHS clears the FIFO buffer on receiving the SOF packet immediately after the frame in which the USBHS expected to receive the IN token. Even if the SOF packet is corrupted, the FIFO buffer is cleared at the time the SOF packet is expected to be received by using the internal complementation function.

In host controller mode, set the IITV[2:0] bits to 000b.

Set the IITV[2:0] bits to 000b when the selected pipe is not used for isochronous transfers.

30.2.38 PIPEnCTR : Pipe n Control Register (n = 1 to 9)

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x070 + 0x2 × (n - 1)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSTS	INBUFM	CSCLR	CSSTS	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	x	0	0	0	0	0	0	x	x	x	0	0

Bit	Symbol	Function	R/W
1:0	PID[1:0]	Response PID 0 0: NAK response 0 1: BUF response (depends on buffer state) 1 0: STALL response 1 1: STALL response	R/W
4:2	—	The read values are undefined. The write value should be 0.	R/W
5	PBUSY	Pipe Busy Flag 0: Pipe n not in use for the transaction 1: Pipe n in use for the transaction	R
6	SQMON	Sequence Toggle Bit Monitor Flag 0: DATA0 1: DATA1	R
7	SQSET	Sequence Toggle Bit Set*1 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA1. This bit is read as 0.	R/W
8	SQCLR	Sequence Toggle Bit Clear*1 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0	W
9	ACLRM	Auto Buffer Clear Mode*2 0: Disable 1: Enable (initialize all buffers)	R/W
10	ATREPM	Auto Response Mode*1 *3 0: Disable auto response mode 1: Enable auto response mode	R/W
11	—	The read value is undefined. The write value should be 0.	R/W
12	CSSTS	CSSTS Status Flag 0: Start-split (SSPLIT) transaction, or processing for devices that are not using split transactions, in progress. 1: Complete-split (CSPLIT) transaction in progress.	R
13	CSCLR	CSPLIT Status Clear 0: Invalid (writing 0 has no effect) 1: Clear CSSTS to 0	W
14	INBUFM	Transmit Buffer Monitor Flag*3 0: No data to be transmitted is in the FIFO buffer 1: Data to be transmitted is in the FIFO buffer	R
15	BSTS	Buffer Status Flag 0: Buffer access disabled 1: Buffer access enabled	R

Note: S-TYPE-3, P-TYPE-3

Note 1. Only set the ATREPM, SQCLR, and SQSET bits while PID is NAK. Before setting this bit, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the PBUSY flag through the software is not necessary.

Note 2. Only set the ACLRM bit while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting this bit, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits

from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the PBUSY flag through the software is not necessary.

Note 3. The ATREPM bit and the INBUFM flag in the PIPE6CTR to PIPE9CTR registers are reserved. The read value is undefined. The write value must be 0.

PID[1:0] bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction on the selected pipe.

The default PID[1:0] setting is NAK. Change the PID[1:0] setting to BUF to use the associated pipe for USB transfer. Table 30.10 and Table 30.11 show the basic operations of the USBHS (when there are no errors in the communication packets) based on the PID[1:0] bit setting.

After changing the PID[1:0] setting from BUF to NAK through the software during USB communication on the selected pipe, check that the PBUSY bit is 1 to see if USB transfer on the selected pipe has actually entered the NAK state. If the USBHS changes the PID[1:0] bits to NAK, checking the PBUSY bit through the software is not necessary.

The USBHS changes the PIPEnCTR.PID[1:0] setting in the following cases:

- The USBHS sets PID to NAK on recognizing completion of the transfer when the selected pipe is in the receiving direction and the PIPECFG.SHTNAK bit for the selected pipe is set to 1 by software
- The USBHS sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the selected pipe
- The USBHS sets PID to NAK on detecting a USB bus reset in device controller mode
- The USBHS sets PID to NAK on detecting a reception error, such as a CRC error, three consecutive times in host controller mode
- The USBHS sets PID to STALL (11b) on receiving the STALL handshake in host controller mode

To specify the response type, set the PID[1:0] bits as follows:

- To transition from NAK (00b) to STALL, set 10b
- To transition from BUF (01b) to STALL, set 11b
- To transition from STALL (11b) to NAK, set 10b and then 00b
- To transition from STALL to BUF, set 00b (NAK) and then 01b (BUF)

Table 30.10 Operation of the USBHS based on the PIPEnCTR.PID[1:0] setting in host controller mode

PID[1:0] value	Transfer type (TYPE[1:0] value)	Transfer direction (DIR value)	USBHS operation
00b (NAK)	Does not depend on the setting	Does not depend on the setting	Does not issue tokens
01b (BUF)	Bulk or Interrupt	Does not depend on the setting	Issues tokens when the DVSTCTR0.UACT bit is 1 and the FIFO buffer associated with the selected pipe is ready for transmission and reception. Does not issue tokens when the DVSTCTR0.UACT bit is 0 or the FIFO buffer associated with the selected pipe is not ready for transmission or reception.
	Isochronous	Does not depend on the setting	Issues tokens when the DVSTCTR0.UACT bit is 1, regardless of the state of the FIFO buffer associated with the selected pipe. Does not issue tokens when UACT = 0.
10b (STALL) or 11b (STALL)	Does not depend on the setting	Does not depend on the setting	Does not issue tokens.

Table 30.11 Operation of the USBHS based on the PIPEnCTR.PID[1:0] setting in device controller mode

PID[1:0] value	Transfer type (TYPE[1:0] value)	Transfer direction (DIR value)	USBHS operation
00b (NAK)	Bulk or Interrupt	Does not depend on the setting	Returns NAK in response to the token from the USB host
	Isochronous	Receiving direction (DIR = 0)	Returns nothing in response to the token from the USB host
		Transmitting direction (DIR = 1)	Transmits a zero-length packet in response to the token from the USB host
01b (BUF)	Bulk	Receiving direction (DIR = 0)	Receives data and returns ACK or NYET in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception. Otherwise, returns NAK. Returns ACK in response to the PING token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception. Otherwise, returns NAK.
	Interrupt	Receiving direction (DIR = 0)	Receives data and returns ACK response in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception. Otherwise, returns NAK.
	Bulk or Interrupt	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the FIFO buffer associated with the selected pipe is ready for transmission. Otherwise, returns NAK.
	Isochronous	Receiving direction (DIR = 0)	Receives data in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception. Otherwise, discards the data.
		Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the associated FIFO buffer is ready for transmission. Otherwise, transmits a zero-length packet.
10b (STALL) or 11b (STALL)	Bulk or Interrupt	Does not depend on the setting.	Returns STALL in response to the token from the USB host
	Isochronous	Does not depend on the setting.	Returns nothing in response to the token from the USB host

PBUSY flag (Pipe Busy Flag)

The PBUSY flag indicates whether the selected pipe is being used for the current transaction.

The USBHS changes the PBUSY bit from 0 to 1 on start of the USB transaction for the selected pipe, and changes the PBUSY bit from 1 to 0 on completion of one transaction.

Reading the PBUSY bit by software after PID is set to NAK allows you to check whether changing the pipe setting is possible. For details, see section 30.3.7.1. Pipe control register switching procedures.

SQMON flag (Sequence Toggle Bit Monitor Flag)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction of the selected pipe.

When the selected pipe is not the isochronous transfer type, the USBHS toggles the SQMON flag on successful completion of the transaction. However, the USBHS does not toggle the SQMON flag when a DATA-PID mismatch occurs during transfer in the receiving direction.

SQSET bit (Sequence Toggle Bit Set)

Setting the SQSET bit to 1 through the software causes the USBHS to set DATA1 as the expected value of the sequence toggle bit for the next transaction on the selected pipe.

SQCLR bit (Sequence Toggle Bit Clear)

Setting the SQCLR bit to 1 through the software causes the USBHS to clear the expected value of the sequence toggle bit for the next transaction on the selected pipe to DATA0.

In host controller mode, when this bit is set to 1 for a bulk OUT transfer pipe, the USBHS starts the next transfer for the selected pipe from a PING token.

ACLRM bit (Auto Buffer Clear Mode)

The ACLRM bit enables or disables auto buffer clear mode for the selected pipe. To completely clear the data in the FIFO buffer allocated to the selected pipe, write 1 and then 0 to the ACLRM bit continuously.

Table 30.12 shows the data cleared by writing 1 and 0 continuously to the ACLRM bit and the cases in which this processing is required.

Table 30.12 Data cleared by the USBHS when ACLRM = 1

Number	Data cleared by setting the ACLRM bit	Situations requiring data clear
1	All data in the FIFO buffer allocated to the selected pipe (two FIFO buffers in double buffer mode)	When clearing all data in the FIFO buffer allocated to the selected pipe
2	Interval count value when the selected pipe is the isochronous transfer type	When resetting the interval count value

ATREPM bit (Auto Response Mode)

The ATREPM bit enables or disables auto response mode for the selected pipe.

This bit can be set to 1 in device controller mode when the selected pipe is the bulk transfer type. When the bit is set to 1, the USBHS responds to the token from the USB host as follows:

- When the selected pipe is set for bulk IN transfers (PIPECFG.TYPE[1:0] = 01b and PIPECFG.DIR = 1):
  - When the ATREPM bit = 1 and PID = BUF, the USBHS transmits a zero-length packet in response to the IN token.
  - The USBHS updates the sequence toggle bit (DATA-PID) each time the USBHS receives ACK from the USB host. In a single transaction, the IN token is received, a zero-length packet is transmitted, and then ACK is received. The USBHS does not generate the BRDY or BEMP interrupt.
- When the selected pipe is set for bulk OUT transfers (PIPECFG.TYPE[1:0] = 01b and PIPECFG.DIR = 0):
  When the ATREPM bit = 1 and PID = BUF, the USBHS returns NAK in response to the OUT token or PING token and generates an NRDY interrupt.

For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode. When the selected pipe uses isochronous transfer, always set this bit to 0.

In host controller mode, always set the ATREPM bit to 0.

CSSTS flag (CSSTS Status Flag)

In host controller mode, the CSSTS flag indicates the complete-split status of a split transaction. It is valid for pipes that are not the isochronous transfer type.

The USBHS sets the CSSTS flag to 1 at the beginning of the complete-split transaction, and sets the CSSTS flag to 0 on detecting completion of the complete-split transaction. If a detach event is detected during the transaction, the CSSTS flag might stay set to 1. In this case, clear the CSSTS flag by setting the CSCLR bit to 1.

Values read from the CSSTS flag in device controller mode are invalid.

CSCLR bit (CSPLIT Status Clear)

In host controller mode, if the software sets the CSCLR bit to 1, the USBHS clears the CSSTS flag to 0. In split transactions, set the CSCLR bit to 1 by software to force the next transfer to restart from start-split. Because the USBHS automatically clears the CSSTS flag to 0 at the end of a successful complete-split transaction in a normal split transaction, clearing the flag through software is not required. Only clear the CSSTS flag using the CSCLR bit when the DVSTCTR0.UACT bit is set to 0 or when no transfer was made after a detach detect. If the CSCLR bit is set to 1 while the CSSTS flag is 0, the CSSTS flag remains 0.

In device controller mode, always write 0 to the CSCLR bit.

INBUFM flag (Transmit Buffer Monitor Flag)

The INBUMFM flag indicates the FIFO buffer status for the selected pipe in the transmitting direction.

When the selected pipe is set in the transmitting direction (PIPECFG.DIR = 1), the USBHS sets this bit to 1 when the CPU or DMAC/DTC completes writing data to at least one FIFO buffer plane.

The USBHS sets this bit to 0 when the USBHS completes transmission of data from the FIFO buffer plane to which all the data is written. In double buffer mode (PIPECFG.DBLB = 1), the USBHS sets the INBUFM flag to 0 when the USBHS completes transmission of data from the two FIFO buffer planes before the CPU or DMAC/DTC completes writing data to one FIFO buffer plane.

The INBUFM flag indicates the same value as the BSTS flag when the selected pipe is in the receiving direction (PIPECFG.DIR = 0).

BSTS flag (Buffer Status Flag)

The BSTS flag indicates the FIFO buffer status for the selected pipe. The meaning of the BSTS flag depends on the PIPECFG.DIR, PIPECFG.BFRE, and DnFIFOSEL.DCLRM settings, as shown in Table 30.13.

Table 30.13 BSTS flag operation

DIR value	BFRE value	DCLRM value	Meaning of BSTS flag
0	0	0	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 on completion of data read
		1	Setting prohibited
	1	0	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 when the software sets the BCLR bit in the port control register to 1 after the data read is complete
		1	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 on completion of data read
1	0	0	Sets to 1 when transmit data can be written to the FIFO buffer, and clears to 0 on completion of data write
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

30.2.39 PIPEnTRE : Pipe n Transaction Counter Enable Register (n = 1 to 5)

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x090 + 0x4 × (n - 1)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TRENB	TRCLR	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	0	0	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	—	The read values are undefined. The write value should be 0.	R/W
8	TRCLR	Transaction Counter Clear 0: Invalid (writing 0 has no effect) 1: Clear current counter value	R/W
9	TRENB	Transaction Counter Enable 0: Disable transaction counter 1: Enable transaction counter	R/W
15:10	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Only change the PIPEnTRE register settings while the PIPEnCTR.CSSTS flag is 0 and the PIPEnCTR.PID[1:0] bits are 00b (NAK response). Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY and PIPEnCTR.CSSTS flags is 0. However, software processing to check the PIPEnCTR.PBUSY flag is not required if the USBHS has changed the PID[1:0] bits to 00b (NAK response).

TRCLR bit (Transaction Counter Clear)

When the TRCLR bit sets to 1, the USBHS clears the count value of the transaction counter associated with the selected pipe and then clears the TRCLR bit to 0.

TRENB bit (Transaction Counter Enable)

The TRENB bit enables or disables the transaction counter.

For receiving pipes, setting the TRENB bit to 1 after setting the total number of the packets to be received in the PIPEnTRN.TRNCNT[15:0] bits through the software allows the USBHS to control hardware on having received the number of packets equal to the TRNCNT[15:0] setting as follows:

- When the PIPECFG.SHTNAK bit is 1, the USBHS changes the PID bits to NAK for the associated pipe on having received the number of packets equal to the TRNCNT[15:0] setting
- When the PIPECFG.BFRE bit is 1, the USBHS asserts the BRDY interrupt on having received the number of packets equal to the TRNCNT[15:0] setting and then reading the last received data.

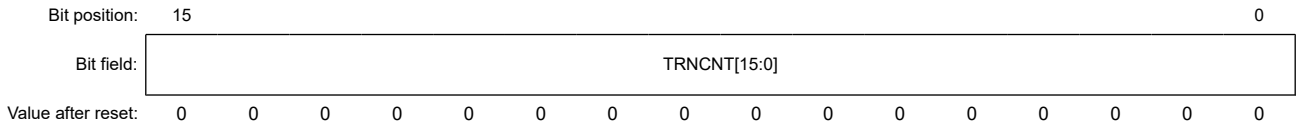
For transmitting pipes, set the TRENB bit to 0.

When the transaction counter is not used, set this bit to 0. When the transaction counter is used, set the TRNCNT[15:0] bits before setting this bit to 1. Set this bit to 1 before receiving the first packet to be counted by the transaction counter.

30.2.40 PIPEnTRN : Pipe n Transaction Counter Register (n = 1 to 5)

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x092 + 0x4 × (n - 1)



Bit	Symbol	Function	R/W
15:0	TRNCNT[15:0]	Transaction Counter*1 ● When written to: Specifies the total packets (number of transactions) to be received by pipe n. ● When read from: When PIPEnTRE.TRENB is 0, indicates the specified number of transactions. When PIPEnTRE.TRENB is 1, indicates the current transaction count.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Only set the TRNCNT[15:0] bits while PID is NAK and PIPEnTRE.TRENB is 0. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the PBUSY flag through the software is not necessary.

The PIPEnTRN registers retain their settings during a USB bus reset.

TRNCNT[15:0] bits (Transaction Counter)

The USBHS increments the value of the TRNCNT[15:0] bits by one when all of the following conditions are satisfied on receiving the packet:

- The PIPEnTRE.TRENB bit is 1
- (TRNCNT[15:0] set value ≠ current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the PIPEMAXP.MXPS[8:0] setting

The USBHS clears the value of the TRNCNT[15:0] bits to 0 when any of the following conditions is satisfied.

All of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- (TRNCNT[15:0] set value = current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the PIPEMAXP.MXPS[8:0] setting

Both the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- The USBHS received a short packet

Both the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- The PIPEnTRE.TRCLR bit was set to 1 by software

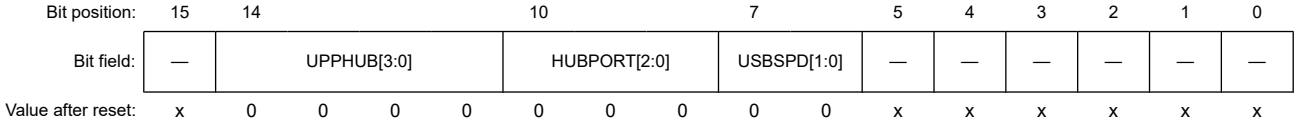
For transmitting pipes, set the TRNCNT[15:0] bits to 0. When the transaction counter is not used, set the TRNCNT[15:0] bits to 0.

Setting the number of transactions to be transferred to the TRNCNT[15:0] bits is enabled only when the PIPEnTRE.TRENB bit is 0. To set the number of transactions to be transferred, set the TRCLR bit to 1 to clear the current counter value before setting the PIPEnTRE.TRENB bit to 1.

30.2.41 DEVADDn : Device Address n Configuration Register (n = 0 to 9, A)

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x0D0 + 0x2 × n (n = 0 to 9)
0x0E4 (n = A)



Bit	Symbol	Function	R/W
5:0	—	The read values are undefined. The write value should be 0.	R/W
7:6	USBSPD[1:0]	Transfer Speed of Communication Target Device 0 0: Do not use DEVADDm 0 1: Low speed 1 0: Full speed 1 1: High speed	R/W
10:8	HUBPORT[2:0]	Communication Target Connecting Hub Port 0 0 0: Connect directly to the USBHS port Others: Port number of the hub	R/W
14:11	UPPHUB[3:0]	Communication Target Connecting Hub Register 0x0: Connect directly to the USBHS port Others: USB address of the hub. The value as 0xB or more is reserved.	R/W
15	—	The read value is undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The DEVADDn register specifies the transfer speed of the peripheral device that is the communication target for pipes 0 to 9.

In host controller mode, set all DEVADDn bits before starting communication to any pipes. Only change the bits in DEVADDn when no valid pipes are using the bit settings. A valid pipe is defined as one that satisfies both of the following conditions:

- DEVADDm is selected in the DEVSEL[3:0] bits
- The PID[1:0] bits are set to BUF for the selected pipe, or the selected pipe is the DCP with the DCPCTR.SUREQ bit set to 1

In device controller mode, set all bits in this register to 0.

USBSPD[1:0] bits (Transfer Speed of Communication Target Device)

The USBSPD[1:0] bits specify the USB transfer speed of the target peripheral device. In host controller mode, the USBHS generates packets based on the USBSPD[1:0] setting. In device controller mode, set these bits to 00b.

HUBPORT[2:0] bits (Communication Target Connecting Hub Port)

In host controller mode, the USBHS generates packets based on the HUBPORT[2:0] setting when performing a split transaction.

UPPHUB[3:0] bits (Communication Target Connecting Hub Register)

In host controller mode, the USBHS generates packets based on the UPPHUB[3:0] setting when performing a split transaction.

30.2.42 LPCTRL : Low Power Control Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x100

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	HWUPM	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	0	0	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
6:1	—	The read values are undefined. The write value should be 0.	R/W
7	HWUPM	Resume Return Mode Setting 0: Hardware does not recover while CPU clock inactive 1: Hardware recovers while CPU clock inactive	R/W
8	—	This bit is read as 0. The write value should be 0.	R/W
15:9	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

HWUPM bit (Resume Return Mode Setting)

The HWUPM bit specifies whether to enable hardware processing for return from low power mode even while the CPU clock is inactive.

In device controller mode, processing for return from low power mode on detecting Resume is enabled even while the CPU clock is inactive.

This bit specifies whether to detect Resume while the CPU clock is inactive. The PL1CTRL1.L1EXTMD bit controls whether to make a hardware return. To make a hardware return from the LPM L1 low power state while the CPU clock is inactive, set this bit and the PL1CTRL1.L1EXTMD bit to 1.

30.2.43 LPSTS : Low Power Status Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x102

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	SUSPENDM	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	0	x	0	x	x	x	0	x	x	x	x	0	x	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	—	The read value is undefined. The write value should be 0.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
7:4	—	The read values are undefined. The write value should be 0.	R/W
8	—	This bit is read as 0. The write value should be 0.	R/W
11:9	—	The read values are undefined. The write value should be 0.	R/W
12	—	This bit is read as 0. The write value should be 0.	R/W
13	—	The read value is undefined. The write value should be 0.	R/W
14	SUSPENDM	UTMI SuspendM Control 0: UTMI suspension mode 1: UTMI normal mode	R/W
15	—	The read value is undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

SUSPENDM bit (UTMI SuspendM Control)

The SUSPENDM bit controls the SuspendM signal to be sent to the PHY designed under the UTMI specification. The initial value is 0 with the UTMI is in suspension mode.

Set this bit to 1 to supply the PHY clock to operate the USB2.0 host or device controller.

In compliance with the UTMI specification, clock output is normally controlled by the SuspendM signal. When the SUSPENDM bit is 0, the clock to LINK is stopped. Because the PHY in this MCU follows the UTMI specification, setting the SUSPENDM bit to 1 is required to supply the PHY clock. For the clock settings, see section 30.3.3. Supplying the Clock.

When the SUSPENDM bit is 0, the USBHS cannot be written to but can be read from. The registers listed in Table 30.14 are writable even when the SUSPENDM bit is 0.

Table 30.14 Registers that can be written to by software when SUSPENDM = 0

Address	Register or bit name
0x4006_0000	SYSCFG register
0x4006_0002	BUSWAIT register
0x4006_0032	INTENB1.PDDETINTE bit
0x4006_0100	LPCTRL register
0x4006_0102	LPSTS register
0x4006_0140	BCCTRL register

The value written to the SYSCFG register while the PHY clock is inactive is updated only after the PHY clock begins oscillating. The PHY clock oscillates in the following cases described in this section.

When SUSPENDM bit is set to 1, the PLLSTA.PLLLOCK flag is set to 1 after the predetermined time has passed. The USB-PHY internal PLL is stopped when the SUSPENDM bit is set to 0.

For details on CL-only mode, see section 30.2.17. PHYSET : PHY Setting Register.

If the PL1CTRL1.L1EXTMD bit is 0, setting or clearing of this bit is controlled by software. If the PL1CTRL1.L1EXTMD bit is 1, transitions to the L1 or L2 state of this bit are controlled by software and recovery from the L1 or L2 state is controlled by hardware.

30.2.44 BCCTRL : Battery Charging Control Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x140

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PDDETSTS	CHGDETSTS	—	—	DCPMODE	VDMSRCE	IDPSINKE	VDPSRCE	IDMSINKE	IDPSRCE
Value after reset:	x	x	x	x	x	x	0	0	x	x	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IDPSRCE	IDPSRC Control*2 0: Disable IDP_SRC circuit 1: Enable IDP_SRC circuit	R/W
1	IDMSINKE	IDMSINK Control*2 0: Disable IDM_SINK circuit 1: Enable IDM_SINK circuit	R/W*1
2	VDPSRCE	VDPSRC Control*2 0: Disable VDP_SRC circuit 1: Enable VDP_SRC circuit	R/W
3	IDPSINKE	IDPSINK Control*2 0: Disable IDP_SINK circuit 1: Enable IDP_SINK circuit	R/W
4	VDMSRCE	VDMSRC Control*2 0: Disable VDM_SRC circuit 1: Enable VDM_SRC circuit	R/W
5	DCPMODE	DCP Mode Control 0: Disable RDCP_DAT resistor 1: Enable RDCP_DAT resistor	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	CHGDETSTS	CHGDET Status Flag 0: The CHGDET pin is at low level 1: The CHGDET pin is at high level	R
9	PDDETSTS	PDDET Status Flag 0: The PDDET pin is at low level 1: The PDDET pin is at high level	R
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. All bits in the BCCTRL register can be changed while the UTMI clock is inactive.
Note 2. In device controller mode, set the IDPSRCE, IDMSINKE, VDPSRCE, IDPSINKE, and VDMSRCE bits to 1 after setting the SYSCFG.DRPD bit to 0.

IDPSRCE bit (IDPSRC Control)

In device controller mode, set the IDPSRCE bit to 1 to perform data contact detection.

The Battery Charging Standard provides two ways to handle data contact detection, one through the software and one using hardware to contact the data line. The IDPSRCE bit uses the hardware method.

When the IDPSRCE bit is set to 1, the USBHS enables the IDP_SRC circuit and, at the same time, controls D- pull-down.

(D- pull-down is controlled with the VUH_DMPULLDOWN signal.)

IDMSINKE bit (IDMSINK Control)

In device controller mode, set the IDMSINKE bit to 1 to perform primary detection.

VDPSRCE bit (VDPSRC Control)

In device controller mode, set the VDPSRCE bit to 1 to perform primary detection.

IDPSINKE bit (IDPSINK Control)

In device controller mode, set the IDPSINKE bit to 1 to perform secondary detection. In host controller mode, set this bit to 1 to enable the portable device detection circuit.

VDMSRCE bit (VDMSRC Control)

In device controller mode, set the VDMSRCE bit to 1 to perform secondary detection. Setting this bit to 1 enables the DCP detection circuit. In host controller mode, set this bit to 1 when a portable device is detected. Setting this bit to 1 allows the device that is performing primary detection to determine the charger detection method.

DCPMODE bit (DCP Mode Control)

Set the DCPMODE bit to 1 to operate as a dedicated charging port (DCP). Setting this bit to 1 disables USB communication.

CHGDETSTS flag (CHGDET Status Flag)

The CHGDETSTS flag indicates the charger port detection state.

PDDETSTS flag (PDDET Status Flag)

The PDDETSTS flag indicates the following states based on the controller mode:

- In host controller mode: PD detection state
- In device controller mode: DCP detection state

30.2.45 PL1CTRL1 : Function L1 Control Register 1

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x144

Bit position:	15	14	13	11	7	3	2	0
Bit field:	—	L1EXTMD	—	HIRDTHR[3:0]	DVSQ[3:0]	L1NEGOMD	L1RESPMD[1:0]	L1RESPEN
Value after reset:	x	0	x	x	0 0 0 0	0 0 0 0	0	0 0	0

Bit	Symbol	Function	R/W
0	L1RESPEN	L1 Response Enable 0: Do not support LPM 1: Support LPM	R/W
2:1	L1RESPMD[1:0]	L1 Response Mode 0 0: NYET response 0 1: ACK response 1 0: STALL response 1 1: Response based on L1NEGOMD setting	R/W
3	L1NEGOMD	L1 Response Negotiation Control This bit is only valid when the L1RESPMD[1:0] value is 11b. 0: Return ACK when received HIRD is larger than HIRDTHR[3:0]. Otherwise (including when HIRD = HIRDTHR[3:0]), return NYET 1: Return ACK when received HIRD is smaller than HIRDTHR[3:0]. Otherwise (including when HIRD = HIRDTHR[3:0]), return NYET	R/W

Bit	Symbol	Function	R/W
7:4	DVSQ[3:0]	DVSQ Extension Flag 0 0 0 0: Powered state 0 0 0 1: Default state 0 0 1 0: Address state 0 0 1 1: Configured state 0 1 x x: Suspend state 1 0 x x: L1 state	R
11:8	HIRDTHR[3:0]	L1 Response Negotiation Threshold Value HIRD threshold value used when the L1RESPMD[1:0] bits are 11b. The format is the same as the HIRD field in HL1CTRL.	R/W
13:12	—	The read values are undefined. The write value should be 0.	R/W
14	L1EXTMD	PHY Control Mode at L1 Return 0: Do not set LPSTS.SUSPENDM bit through hardware when Host K is received 1: Set LPSTS.SUSPENDM bit through hardware when Host K is received	R/W
15	—	The read value is undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

L1RESPEN bit (L1 Response Enable)

If the USBHS receives an LPM token while the L1RESPEN bit is 0, it returns no response. If the USBHS receives an LPM token while this bit is 1, it returns a response based on the L1RESPMD[1:0] setting.

L1RESPMD[1:0] bits (L1 Response Mode)

When the L1RESPEN bit is set to 1, the USBHS returns a response to the LPM token based on the setting in the L1RESPMD[1:0] bits.

L1NEGOMD bit (L1 Response Negotiation Control)

The L1NEGOMD bit specifies the negotiation function for the HIRD value.

HIRDTHR[3:0] bits (L1 Response Negotiation Threshold Value)

The HIRDTHR[3:0] bits specify the HIRD threshold value used for L1NEGOMD. The format of the set value is the same as the HIRD field in HL1CTRL.

L1EXTMD bit (PHY Control Mode at L1 Return)

The L1EXTMD bit specifies the LPSTS.SUSPENDM bit control method when a host K signal is received in the L1 state while the LPSTS.SUSPENDM bit is 0 and the PHY is inactive.

Similar to the Suspend constraints, because the minimum host K period is 50 µs, the PHY might not recover within the host K period specified for software settings on return. The initial value is within software control, so set this bit to 1 during the initialization process when the L1 state is supported.

The LPSTS.SUSPENDM bit is controlled by software on transition to the L1 state regardless of the setting in this bit. It is not cleared by hardware.

When this bit is set to 1, the LPSTS.SUSPENDM bit is also set to 1 on return from L2.

30.2.46 PL1CTRL2 : Function L1 Control Register 2

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x146

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	RWEMON	HIRDMON[3:0]				—	—	—	—	—	—	—	—
Value after reset:	x	x	x	0	0	0	0	0	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	—	The read values are undefined. The write value should be 0.	R/W
11:8	HIRDMON[3:0]	HIRD Value Monitor When set, indicates that the HIRD field value reflects the lastreceived LPM token.	R
12	RWEMON	RWE Value Monitor When set, indicates that the RWE bit value reflects the lastreceived LPM token.	R
15:13	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

HIRDMON[3:0] bits (HIRD Value Monitor)

Access the HIRDMON[3:0] bits when monitoring the HIRD field value of the received LPM token. The bits reflect the HIRD field value of the last received LPM token.

RWEMON bit (RWE Value Monitor)

Access the RWEMON bit when monitoring the RWE field value of the received LPM token. The bits reflect the RWE field value of the last received LPM token.

30.2.47 HL1CTRL1 : Host L1 Control Register 1

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x148

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	L1STATUS[1:0]		L1REQ
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
0	L1REQ	L1 Transition Request Set this bit to 1 when requesting a transition to the L1 state. This bit is cleared to 0 by the hardware when the LPM transaction is complete.	R/W
2:1	L1STATUS[1:0]	L1 Request Completion Status Indicates the result of the LPM transaction made by the L1REQ bit: 0 0: ACK received 0 1: NYET received 1 0: STALL received 1 1: Transaction error	R
15:3	—	The read value is undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

L1REQ bit (L1 Transition Request)

Set the L1REQ bit to 1 to transition to the L1 state. When the USBHS detects that this bit is 1, it starts the LPM transaction. The USBHS clears this bit to 0 through hardware on completion of the transaction.

L1STATUS[1:0] bits (L1 Request Completion Status)

The L1STATUS[1:0] bits indicate the result of the LPM transaction initiated by the L1REQ bit.

30.2.48 HL1CTRL2 : Host L1 Control Register 2

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x14A

Bit position:	15	14	13	12	11				7	6	5	4	3			0
Bit field:	BESL	—	—	L1RWE	HIRD[3:0]				—	—	—	—	L1ADDR[3:0]			
Value after reset:	0	x	x	0	0	0	0	0	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	L1ADDR[3:0]	LPM Token DeviceAddress Specify the value to be set in the ADDR field of the LPM token	R/W
7:4	—	The read values are undefined. The write value should be 0.	R/W
11:8	HIRD[3:0]	LPM Token HIRD Specify the value to be set in the HIRD field of the LPM token	R/W
12	L1RWE	LPM Token L1 RemoteWake Enable Specify the value to be set in the RWE field of the LPM token	R/W
14:13	—	The read values are undefined. The write value should be 0.	R/W
15	BESL	BESL & Alternate HIRD Selects the K-State drive period on L1 Resume	R/W

Note: S-TYPE-3, P-TYPE-3

L1ADDR[3:0] bits (LPM Token DeviceAddress)

The L1ADDR[3:0] bits specify the value to be set in the ADDR field of the LPM token that the USBHS transmits when the HL1CTRL1.L1REQ bit is set to 1.

HIRD[3:0] bits (LPM Token HIRD)

The HIRD[3:0] bits specify the value to be set in the HIRD field of the LPM token that the USBHS transmits when the HL1CTRL1.L1REQ bit is set to 1. Table 30.15 shows the relationship between the HIRD settings and the HIRD field values.

Table 30.15 Relationship between the HIRD bit settings and the HIRD field values (1 of 2)

HIRD[3:0] setting	When BESL = 0	When BESL = 1
0x0	50 µs (setting prohibited)	75 µs
0x1	125 µs	100 µs
0x2	200 µs	150 µs
0x3	275 µs	250 µs
0x4	350 µs	350 µs
0x5	425 µs	450 µs
0x6	500 µs	950 µs
0x7	575 µs	1950 µs
0x8	650 µs	2950 µs
0x9	725 µs	3950 µs
0xA	800 µs	4950 µs
0xB	875 µs	5950 µs
0xC	950 µs	6950 µs
0xD	1025 µs (setting prohibited)	7950 µs
0xE	1100 µs (setting prohibited)	8950 µs

Table 30.15 Relationship between the HIRD bit settings and the HIRD field values (2 of 2)

HIRD[3:0] setting	When BESL = 0	When BESL = 1
0xF	1175 µs (setting prohibited)	9950 µs

Note: The set value of the HIRD bit is used for the host K drive period on host resume and for the host K period on remote wakeup.

L1RWE bit (LPM Token L1 RemoteWake Enable)

The L1RWE bit specifies the value to be set in the RWE field of the LPM token that the USBHS transmits when the HL1CTRL1.L1REQ bit is set to 1.

The USBHS does not control detection of the remote wakeup signal in the L1 state with this bit. The remote wakeup signal is controlled by the DVSTCTR0.RWUPE bit, as with Suspend.

BESL bit (BESL & Alternate HIRD)

The BESL bit selects the K-state drive period on L1 Resume. For details, see the description of the HIRD bits.

30.2.49 DPUSR0R : Deep Software Standby USB Transceiver Control/Pin Monitor Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x160

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DVBSTSHM	—	DOVCBHM	DOVCAHM	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	x	0	x	x	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
19:0	—	These bits are read as 0. The write value should be 0.	R
20	DOVCAHM	OVRCURA Input Flag Indicates OVRCURA input signal on the USBHS side	R
21	DOVCBHM	OVRCURB Input Flag Indicates OVRCURB input signal on the USBHS side	R
22	—	This bit is read as 0. The write value should be 0.	R
23	DVBSTSHM	VBUS Input Flag Indicates VBUS input signal on the USBHS side	R
31:24	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE-3, P-TYPE-3
Note: This register is accessed by the clock of PCLKA/64 to reduce current consumption in Deep Software Standby mode 1.

30.2.50 DPUSR1R : Deep Software Standby USB Suspend/Resume Interrupt Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x164

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DVBSTSH	—	DOVCBH	DOVCAH	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	DVBSTSHE	—	DOVCBHE	DOVCAHE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	DOVCAHE	OVRCURA Interrupt Enable Clear 0: Disable recovery from Deep Software Standby mode 1 1: Enable recovery from Deep Software Standby mode 1	R/W
5	DOVCBHE	OVRCURB Interrupt Enable Clear 0: Disable recovery from Deep Software Standby mode 1 1: Enable recovery from Deep Software Standby mode 1	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	DVBSTSHE	VBUS Interrupt Enable/Clear 0: Disable recovery from Deep Software Standby mode 1 1: Enable recovery from Deep Software Standby mode 1	R/W
19:8	—	These bits are read as 0. The write value should be 0.	R/W
20	DOVCAH	OVRCURA Interrupt Source Return Status Flag 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1	R
21	DOVCBH	OVRCURB Interrupt Source Return Status Flag 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1	R
22	—	This bit is read as 0. The write value should be 0.	R/W
23	DVBSTSH	VBUS Interrupt Source Return Status Flag 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1	R
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This register is accessed by the clock of PCLKA/64 to reduce current consumption in Deep Software Standby mode 1.

30.2.51 DPUSR2R : Deep Software Standby USB Suspend/Resume Interrupt Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x168

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	DMINTE	DPINTE	—	—	DMVAL	DPVAL	—	—	DMINT	DPINT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DPINT	Indication of Return from DP Interrupt Source 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1	R
1	DMINT	Indication of Return from DM Interrupt Source 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1	R
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	DPVAL	DP Input Indicates DP input signal on the USBHS side	R
5	DMVAL	DM Input Indicates DM input signal on the USBHS side	R
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	DPINTE	DP Interrupt Enable Clear 0: Disable recovery from Deep Software Standby mode 1 1: Enable recovery from Deep Software Standby mode 1	R/W
9	DMINTE	DM Interrupt Enable Clear 0: Disable recovery from Deep Software Standby mode 1 1: Enable recovery from Deep Software Standby mode 1	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This register is accessed by the clock of PCLKA/64 to reduce current consumption in Deep Software Standby mode 1.

30.2.52 DPUSRCR : Deep Software Standby USB Suspend/Resume Command Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x16A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FIXPHYPD	FIXPHY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FIXPHY	USB Transceiver Control Fix 0: Normal mode 1: Invoke/recover from Deep Software Standby mode 1	R/W
1	FIXPHYPD	USB Transceiver Control Fix for PLL 0: Normal mode 1: Invoke/recover from Deep Software Standby mode 1	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This register is accessed by the clock of PCLKA/64 to reduce current consumption in Deep Software Standby mode 1.

30.3 Operation

30.3.1 System Control

This section describes register settings required for initializing the USBHS and controlling power consumption.

30.3.1.1 Setting data to the USBHS registers

Setting the SYSCFG.USBE bit to 1 after starting the PHY clock supply enables and starts USBHS operation. For information on how to supply the PHY clock, see section 30.3.3. Supplying the Clock.

30.3.1.2 Selecting the controller function

The USBHS can operate as a host or device controller.

Use the SYSCFG.DCFM bit to select one of these USBHS functions. The DCFM bit must be changed in the initial settings immediately after a reset or in the D+ pull-up-disabled state (SYSCFG.DPRPU bit = 0) and D+ and D- pull-down- disabled state (SYSCFG.DRPD bit = 0).

30.3.2 Controlling the USB data bus using resistors

The USBHS provides pull-up and pull-down resistors for the D+ and D- lines. Pull these lines up or down by setting the SYSCFG.DPRPU and DRPD bits.

In device controller mode, confirm that connection to the USB host is made, and then set the SYSCFG.DPRPU bit to 1 and pull up the D+ line (in full-speed communication).

When the SYSCFG.DPRPU bit is set to 0 during communication with a PC, the USBHS disables the pull-up resistor for the USB data line, thereby notifying the USB host of disconnection.

Table 30.16 shows the settings for controlling the resistors for the USB data bus. Control the USB data bus appropriately for your system using the DRPD and DPRPU bit settings.

Table 30.16 Control settings for the USB data bus resistors (excluding OTG operation)

SYSCFG register settings		USB data bus control		
DRPD bit	**DPRPU bit**	**D-Line**	**D+Line**	**Function**
0	0	Open	Open	When resistors not used
0	1	Open	Pull-Up	When operating as a device controller at full-speed
1	0	Pull-Down	Pull-Down	When operating as a host controller
1	1	—	—	Setting prohibited except during OTG operation

30.3.3 Supplying the Clock

Table 30.17 shows the two input clocks required for the USBHS.

Table 30.17 Input clocks

Input clock name	Description
PCLKA	Peripheral module clock A input. There is no constraint on the frequency of the PCLKA input.
PHY clock	PHY clock generated from external input or internal supply ● External input: The clock is generated by the USB-PHY internal PLL based on a 12-MHz, 20-MHz, 24-MHz, or 48-MHz clock supplied to the EXTAL pin from outside the MCU. For the external clock specifications, especially the jitter characteristics, strictly follow the specifications of ±50 ppm. ● Internal supply: The clock is generated by supplying 48 MHz and 60 MHz to the USB-PHY module and selecting CL-only mode (PHYSET.HSEB). High-speed operation is not supported with this mode.

Figure 30.2 illustrates the PHY clock settings.

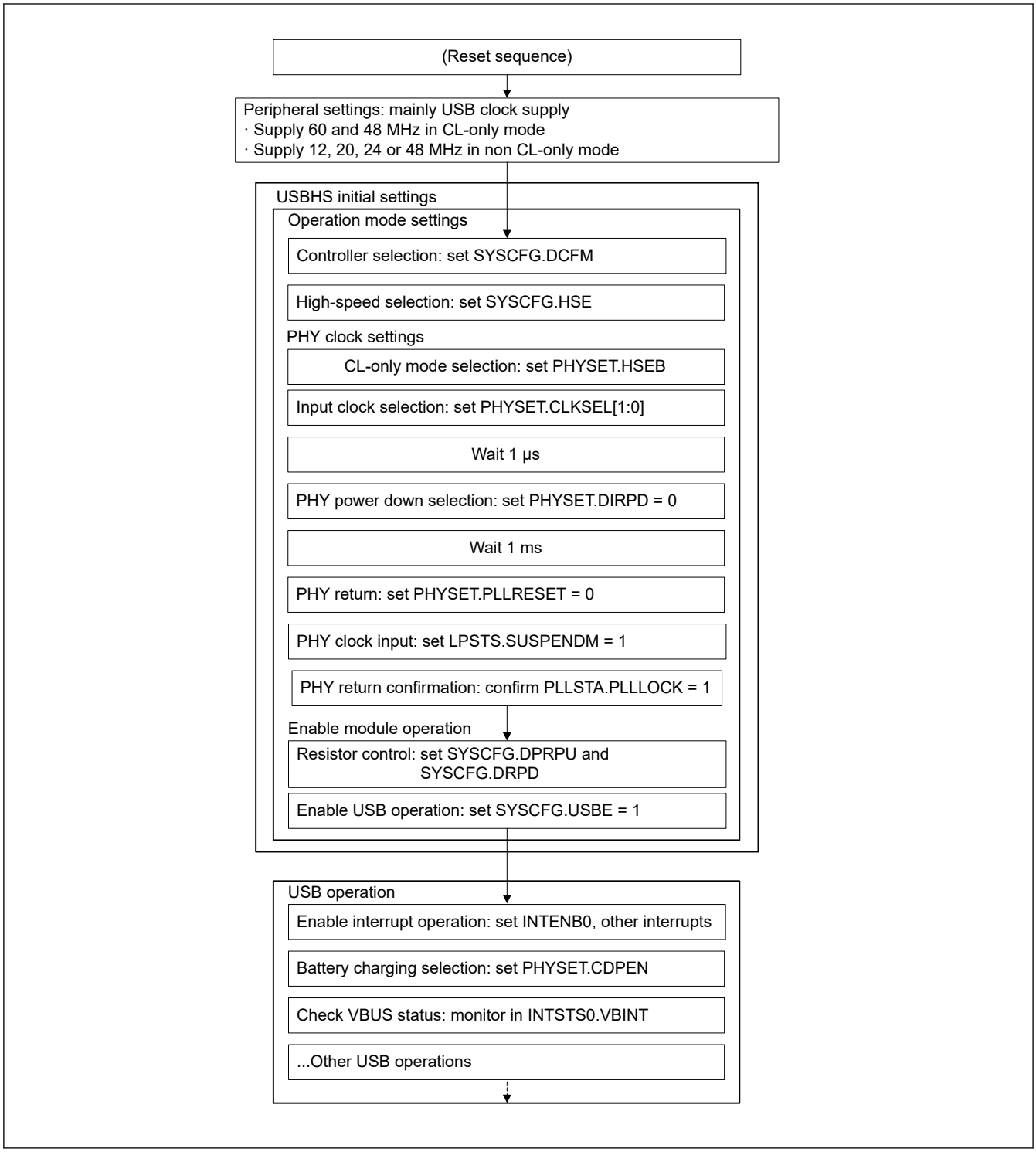


Figure 30.2 PHY clock settings

30.3.4 Constraints on Stopping the Clock

PCLKA and PHY clock can be stopped during disconnection or suspension. However, to stop any of these clocks while the USB is suspended in device controller mode, the stopped clock must be resupplied using the resume interrupt. The PHY clock must be resupplied within 5.5 ms after the resume interrupt is generated.

30.3.5 Interrupts

Table 30.18 lists the interrupt sources in the USBHS. When an interrupt generation condition is satisfied and the interrupt output is enabled using the associated interrupt enable register, the USBHS issues a USBHS interrupt request to the Interrupt Controller Unit (ICU) and a USBHS interrupt is generated.

Table 30.18 Interrupt sources (1 of 2)

Flag to be set to 1	Interrupt name	Interrupt source	Applicable controller function	Status flag
VBINT	VBUS interrupt	• A change in the state of the USB_VBUS input pin is detected (low to high or high to low)	Host or function*1	INTSTS0.VBSTS
RESM	Resume interrupt	• A change in the state of the USB bus is detected in the Suspend state (J-state to K-state or J-state to SE0)	Function	—
SOFR	Frame number update interrupt	In host controller mode: • An SOF packet with a different frame number is transmitted In device controller mode: • When SOFRM is 0: An SOF packet with a different frame number is received • When SOFRM is 1: Failed to receive an SOF packet with the µ frame number 0 because the packet is corrupted.	Host or function	—
DVST	Device state transition interrupt	• A device state transition is detected because of one of the following: – USB bus reset is detected – Suspend state is detected – SET_ADDRESS request is received – SET_CONFIGURATION request is received	Function	PL1CTRL.DVSQ[3:0]
CTRT	Control transfer stage transition interrupt	• A control transfer stage transition is detected because of one of the following: – Setup stage completed – Control write transfer status stage transition occurred – Control read transfer status stage transition occurred – Control transfer completed – Control transfer sequence error occurred	Function	INTSTS0.CTSQ[2:0]
BEMP	Buffer empty interrupt	• The buffer is empty after all FIFO buffer data is transmitted • A packet larger than the maximum packet size is received	Host or function	BEMPSTS.PIPEBEMP
NRDY	Buffer not ready interrupt	In host controller mode: • A STALL response is received from the peripheral device in response to the issued token • The response from the peripheral device in response to the issued token is not received successfully (no response three times consecutively or packet reception error three times consecutively) • An overrun or underrun error occurred during isochronous transfer In device controller mode: • NAK is returned for an IN or OUT token while the PID [1:0] bits were set to 01b (BUF) • A CRC error or bit stuffing error occurred during data reception in isochronous transfer • An interval error occurred during data reception in isochronous transfer	Host or function	NRDYSTS.PIPENRDY
BRDY	Buffer ready interrupt	• The buffer is ready (read or write state)	Host or function	BRDYSTS.PIPEBRDY
OVRCR	Overcurrent input change interrupt	• USBHS_OVRCURA, USBHS_OVRCURA-DS, USBHS_OVRCURB or USBHS_OVRCURB-DS pin state change is detected (low to high or high to low)	Host or function	SYSSTS0.OVCMON [1:0]

Table 30.18 Interrupt sources (2 of 2)

Flag to be set to 1	Interrupt name	Interrupt source	Applicable controller function	Status flag
BCHG	Bus change interrupt	• USB bus state change is detected	Host	—
DTCH	Device disconnect detection interrupt	• Peripheral device disconnect is detected	Host	—
ATTCH	Device connect detection interrupt	• J-state or K-state is detected on the USB bus for 2.5 µs continuously This interrupt can be used to check whether peripheral devices are connected.	Host	—
EOFERR	EOF error detection interrupt	• An EOF error is detected for a peripheral device	Host	—
SACK	Setup normal interrupt	• A setup transaction normal response (ACK) is received	Host	—
SIGN	Setup error interrupt	• A setup transaction error (no response or ACK packet corruption) is detected three consecutive times	Host	—
PDDETINT	PDDETSTS change detect interrupt	• PDDET pin change is detected	Host or function	BCCTRL.PDDETSTS
LPMEND	LPM transaction end interrupt	• LPM transaction is complete	Host	PL1CTRL.DVSQ[3:0]
L1RSMEND	L1 resume end interrupt	• Resume (from L1 state) processing is complete	Host	PL1CTRL.DVSQ[3:0]

Note 1. Although this interrupt can be generated in host controller mode, it is not usually used in this mode.

30.3.5.1 Selecting the USBHS interrupt detection method

Table 30.19 shows operations for an USBHS interrupt output from the USBHS. In case two or more interrupt sources are generated, the USBHS interrupt output method can be set in the SOFCFG.INTL bit. Set the USBHS interrupt output operation based on your system.

Table 30.19 USBHS interrupt operation

USBHS interrupt output (INTL setting)	When one interrupt source is generated	When two or more interrupt sources are generated
Edge detection (SOFCFG.INTL bit = 0)	Low level output until the source is cleared	When one source is cleared, the USBHS interrupt is negated for 32 clocks at 48 MHz (high pulse output)
Level detection (SOFCFG.INTL bit = 1)	Low level output until the source is cleared	Low level output until all sources are cleared

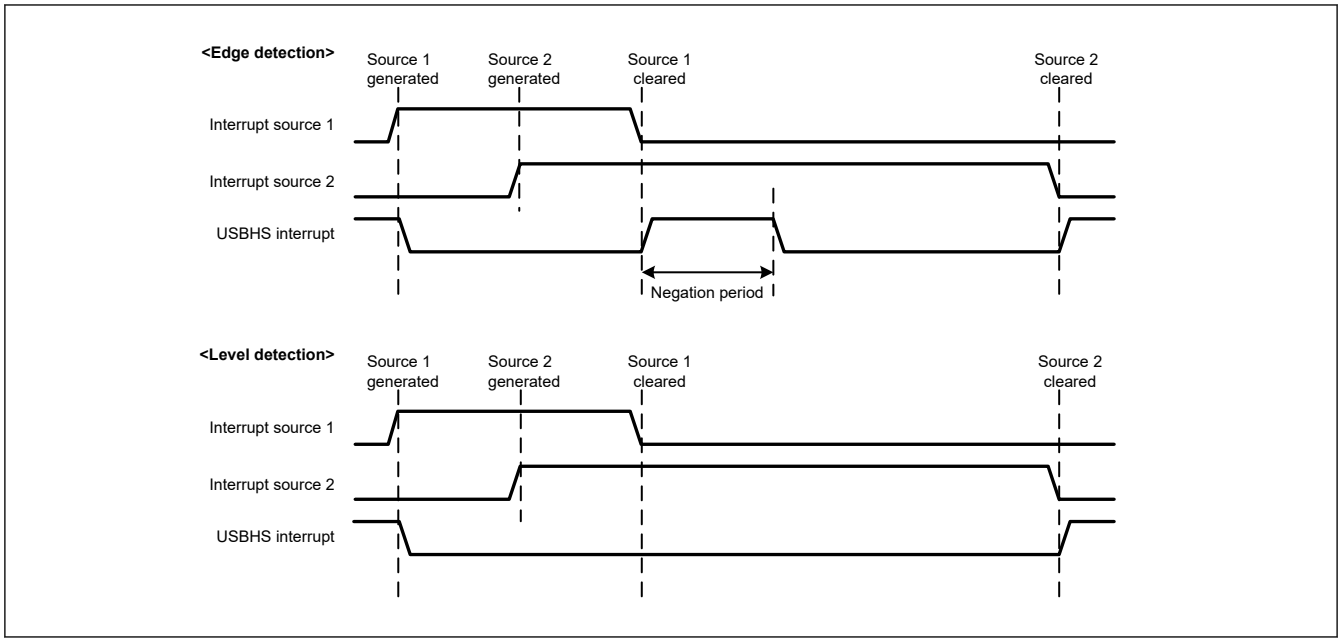


Figure 30.3 USBHS interrupt operation

Figure 30.4 shows an interrupt association chart of the USBHS.

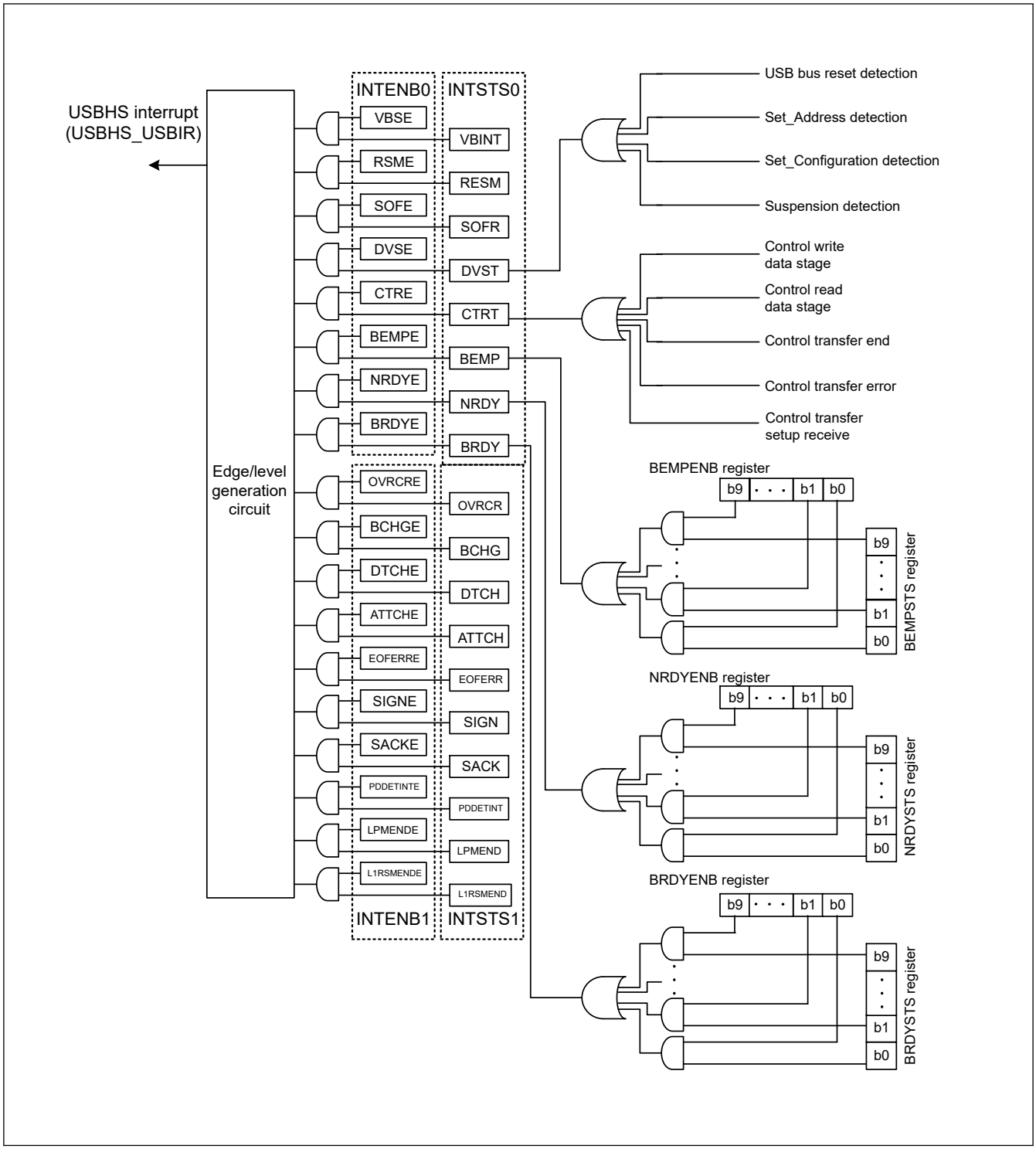


Figure 30.4 USBHS interrupt-related circuits

Table 30.20 shows the interrupts generated by the USBHS.

Table 30.20 USBHS interrupts (1 of 2)

Interrupt name	Interrupt status flag	DTC activation	DMAC activation
USBHS_D0FIFO	DMA transfer request 0	Possible	Possible
USBHS_D1FIFO	DMA transfer request 1	Possible	Possible

Table 30.20 USBHS interrupts (2 of 2)

Interrupt name	Interrupt status flag	DTC activation	DMAC activation
USBHS_USBIR	VBUS interrupt, resume interrupt, frame number update interrupt, device state transition interrupt, control transfer stage transition interrupt, buffer empty interrupt, buffer not ready interrupt, buffer ready interrupt, overcurrent interrupt, bus change interrupt, device disconnect detection interrupt, device connect detection interrupt, EOF error detection interrupt, normal setup operation interrupt, setup error interrupt, PDDETSTS change detection interrupt, LPM transaction end interrupt, and L1 resume end interrupt	Not possible	Not possible

30.3.6 Interrupt Descriptions

30.3.6.1 BRDY interrupt

The BRDY interrupt is generated in both host and device controller modes. This section describes the conditions in which the USBHS sets the associated bit in BRDYSTS to 1. Under these conditions, the USBHS generates a BRDY interrupt if the software sets the bit in BRDYENB associated with the given pipe to 1 and INTENB0.BRDYE bit to 1.

The conditions for generating and clearing the BRDY interrupt depend on the SOFCFG.BRDYM and PIPECFG.BFRE settings for each pipe as follows:

(1) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.

On any of the following conditions, the USBHS generates an internal BRDY interrupt request trigger and sets the BRDYSTS.PIPEBRDY flag associated with the pipe to 1.

For transmitting pipes

- When the DIR bit is changed from 0 to 1 by software
- When writing by the CPU to the FIFO buffer is disabled for a pipe (when the BSTS flag is read as 0) and the USBHS has completed packet transmission. In continuous transfer, a BRDY interrupt is generated on completion of the transmission of data from one FIFO buffer.
- When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode
- No request trigger is generated until completion of writing data to the currently-written FIFO buffer even if transmission to the other FIFO buffer is complete
- When the hardware flushes the buffer of the pipe for isochronous transfers
- When 1 is written to the PIPEnCTR.ACLRM bit, which causes the FIFO buffer to transition from the write-disabled to write-enabled state.

No request trigger is generated for the DCP, that is, during data transmission for control transfers.

For receiving pipes

- When packet reception is successfully complete, enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the given pipe is disabled (when the BSTS flag is read as 0). No request trigger is generated for transactions in which DATA-PID mismatch has occurred. In continuous transmission or reception mode, the request trigger is not generated when the data is of the specified maximum packet size and the buffer has available space. When a short packet is received, the request trigger is generated even if the FIFO buffer has available space. When the transaction counter is used, the request trigger is generated on receiving the specified number of packets. In this case, the request trigger is generated even if the FIFO buffer has available space.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode. No request trigger is generated until completion of reading data from the currently-read FIFO buffer, even if reception by the other FIFO buffer is complete.
  In device controller mode, the BRDY interrupt is not generated in the status stage of control transfers. The PIPEBRDY interrupt status of the selected pipe can be set to 0 by writing 0 to the associated PIPEBRDY flag through the software. In this case, 1s must be written to the associated bits for the other pipes. Clear the BRDY status before accessing the FIFO buffer.

(2) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 1

With these settings, the USBHS generates a BRDY interrupt on completion of reading all data for a single transfer using the receiving pipe, and sets the bit in BRDYSTS associated with the pipe to 1.

On any of the following conditions, the USBHS determines that the last data for a single transfer was received:

- When a short packet including a zero-length packet is received
- When the PIPEnTRN register is used and the number of packets specified in the PIPEnTRN.TRNCNT[15:0] bits are completely received

When the data is completely read after any of these conditions is satisfied, the USBHS determines that all data for a single transfer is completely read.

When a zero-length packet is received while the FIFO buffer is empty, the USBHS determines that all data for a single transfer is completely read when the FRDY flag in the FIFO port control register is 1 and the DTLN[11:0] flags are 0. In this case, to start the next transfer, write 1 to the BCLR bit in the associated port control register through software. With these settings, the USBHS does not detect a BRDY interrupt for the transmitting pipe.

The PIPEBRDY interrupt status of a pipe can be set to 0 by writing 0 to the associated BRDYSTS.PIPEBRDY flag through the software. In this case, 1s must be written to the PIPEBRDY bits for the other pipes.

In this mode, do not change the PIPECFG.BFRE bit setting until all data for a single transfer is processed. When it is necessary to change the PIPECFG.BFRE bit before completion of processing, all FIFO buffers for the pipe must be cleared using the PIPEnCTR.ACLRM bit.

(3) When SOFCFG.BRDYM = 1 and PIPECFG.BFRE = 0

With these settings, the BRDYSTS.PIPEBRDY flag values are linked to the BSTS flag setting for each pipe. In other words, the BRDY interrupt status bits are set to 1 or 0 by the USBHS depending on the FIFO buffer status.

For transmitting pipes

The BRDY interrupt status bits are set to 1 when the FIFO port is ready for write access, and are set to 0 when it is not ready. The BRDY interrupt is not generated for the DCP in the transmitting direction even when it is ready for write access.

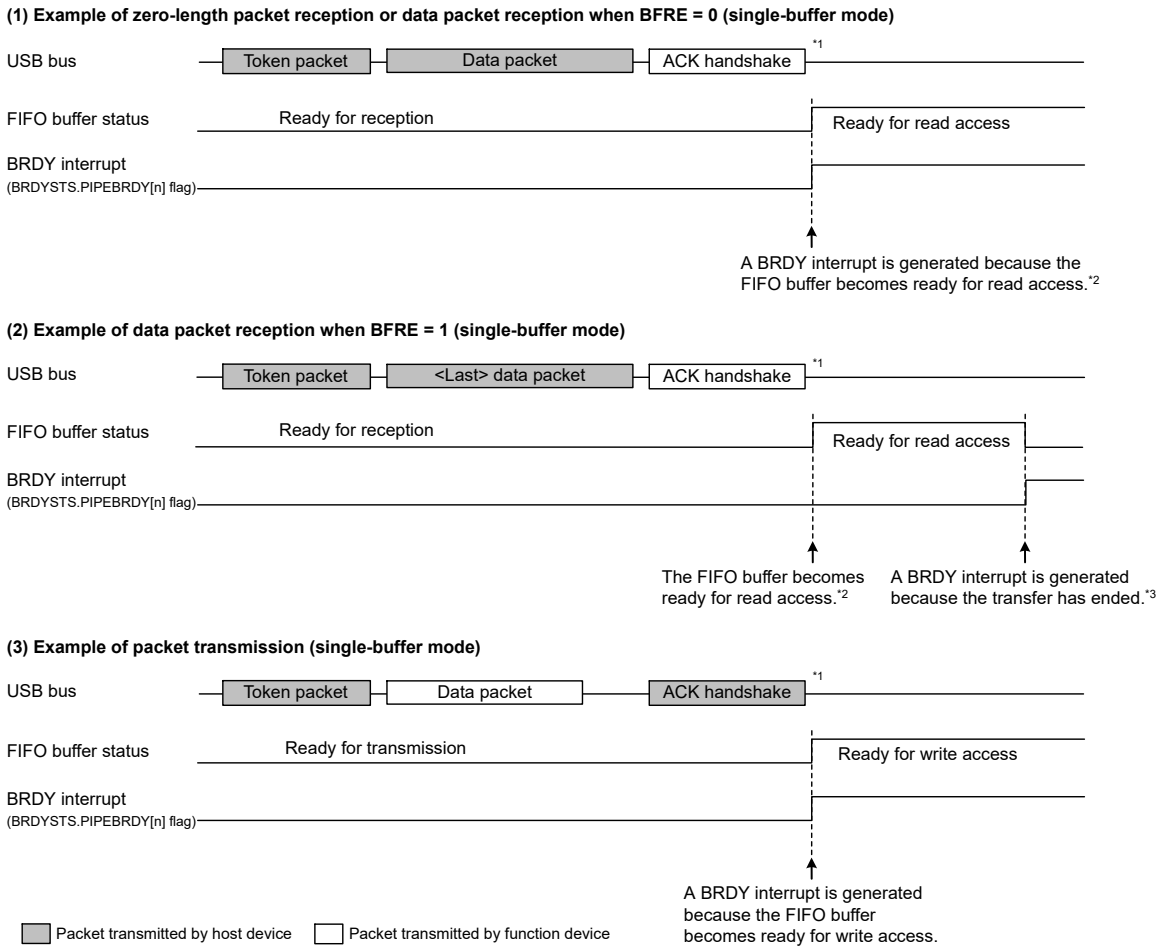
For receiving pipes

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for read access, and are set to 0 when all data is read (not ready for read access).

When a zero-length packet is received while the FIFO buffer is empty, the associated bit is set to 1 and the BRDY interrupt is continuously generated until the software writes 1 to BCLR. With this setting, the PIPEBRDY flag cannot be set to 0 by software.

When the SOFCFG.BRDYM bit is set to 1, set the PIPECFG.BFRE bit for all pipes to 0, and the SOFCFG.INTL bit to 1 for level detection.

Figure 30.5 shows the timing of BRDY interrupt generation.



Note 1. The ACK handshake is not used in isochronous transfers.
Note 2. The FIFO buffer becomes ready for read access under the following condition:
When a packet is received while no unread data remains in the FIFO buffer in the CPU.
Note 3. A transfer ends under either of the following conditions:
1. When a short packet including a zero-length packet is received
2. When the number of packets specified in the transaction counter are received

Figure 30.5 Timing of BRDY interrupt generation

The condition for clearing the INTSTS0.BRDY flag depends on the SOFCFG.BRDYM bit setting value, as shown in Table 30.21

Table 30.21 Conditions for clearing the BRDY flag

BRDYM bit	Condition for clearing BRDY flag
0	The USBHS clears the BRDY flag to 0 when all bits in BRDYSTS are set to 0 by software
1	The USBHS clears the BRDY flag to 0 when the BSTS flags for all pipes have cleared to 0

30.3.6.2 NRDY interrupt

On generating an internal NRDY interrupt request for the pipe whose PID[1:0] bits are set to 01b (BUF response) by software, the USBHS sets the associated NRDYSTS.PIPENRDY flag to 1. If the associated bit in NRDYENB is set to 1 by software, the USBHS sets the INTSTS0.NRDY flag to 1 and generates a USBHS interrupt.

This section describes the conditions in which the USBHS generates the internal NRDY interrupt request for a given pipe.

The internal NRDY interrupt request is not generated during setup transaction execution in host controller mode. During setup transactions in host controller mode, the SACK or SIGN interrupt is detected.

The internal NRDY interrupt request is not generated during status stage execution of the control transfer in device controller mode

(1) In host controller mode when no split transactions occur in the connection

For transmitting pipes

On any of the following conditions, the USBHS detects an NRDY interrupt:

- For isochronous transfer pipes, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer. In this case, the USBHS transmits a zero-length packet following the OUT token and sets the associated NRDYSTS.PIPENRDY flag and the FRMNUM.OVRN flag to 1.
- During communications other than setup transactions on pipes not used for isochronous transfers, when any combination of the following two conditions occurs three consecutive times:
  - No response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device
  - An error is detected in the packet from the peripheral device. In this case, the USBHS sets the associated PIPENRDY flag to 1 and changes the PID[1:0] setting for the associated pipe to 00b (NAK response)
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device (includes STALL for both OUT and PING). In this case, the USBHS sets the associated PIPENRDY flag to 1 and changes the PID[1:0] setting for the associated pipe to 11b (STALL response).

For receiving pipes

- For isochronous transfer pipes, when the time to issue an IN token comes but there is no space available in the FIFO buffer. In this case, the USBHS discards the received data for the IN token and sets the associated PIPENRDY flag and the OVRN flag to 1. When a packet error is detected in the received data for the IN token, the USBHS also sets the FRMNUM.CRCE flag to 1.
- For non-isochronous transfer pipes, when any combination of the following two cases occur three consecutive times:
  - No response is returned from the peripheral device for the IN token issued by the USBHS (when timeout is detected before detection of the DATA packet from the peripheral device)
  - An error is detected in the packet from the peripheral device. In this case, the USBHS sets the associated PIPENRDY flag to 1 and changes the associated PID[1:0] setting for the pipe to 00b (NAK response).
- For isochronous transfer pipes, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the DATA packet from the peripheral device) or an error is detected in the packet from the peripheral device. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for each pipe to 1. The PID[1:0] setting for the pipe is not changed.
- For isochronous transfer pipes, when a CRC error or a bit stuffing error is detected in the received data packet. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for each pipe and the CRCE flag to 1.
- When the STALL handshake is received. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for each pipe to 1 and changes the PID[1:0] setting for the associated pipe to STALL.

(2) In host controller mode when split transactions occur in the connection

For transmitting pipes

On any of the following conditions, the USBHS detects an NRDY interrupt:

- For isochronous transfer pipes, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer. In this case, the USBHS sets the associated RDYSTS.PIPENRDY flag for the given pipes to 1 on issuing a start-split transaction and sets the FRMNUM.OVRN flag to 1. The USBHS also transmits a zero-length packet following the OUT token.
- For non-isochronous transfer pipes, when any combination of the following two cases occurs three consecutive times:
  - No response is returned from the hub for start-split and complete-split transactions (when timeout is detected before detection of the handshake packet from the hub)

  - An error is detected in the packet from the hub. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for the pipe to 1 and changes the associated PID[1:0] setting for the pipe to 00b (NAK response).When an NRDY interrupt is detected on complete-split issuance, the USBHS clears the CSSTS flag to 0.

- When a STALL handshake is received for the complete-split transaction. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for the pipe to 1, changes the associated PID[1:0] setting for the pipe to 11b (STALL response), and clears the CSSTS flag to 0. An interrupt is not detected during setup transaction.

For receiving pipes

- For isochronous transfer pipes, when the time to issue an IN token comes but there is no space available in the FIFO buffer. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for the given pipe and the FRMNUM.OVRN flag to 1 on start-split issuance. The USBHS discards the received data for the IN token.
- During bulk-pipe transfers or transfers other than setup transactions with the DCP, when any combination of the following two cases occurs three consecutive times:
  - No response is returned from the hub for the IN token the USBHS issued on issuance of the start-split or complete-split transactions (when timeout is detected before detection of the data packet from the hub)
  - An error is detected in the packet from the hub. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for the pipe to 1 and changes the associated PID[1:0] setting for the pipe to 00b (NAK response). When this condition occurs during complete-split, the USBHS clears the CSSTS flag to 0.
- During a complete-split transaction for isochronous transfer or interrupt transfer pipes, when any combination of the following two cases occurs three consecutive times:
  - No response is returned from the hub for the IN token issued by the USBHS (when a timeout is detected before detection of the DATA packet from the hub)
  - An error is detected in the packet from the hub. On generating this condition for an interrupt transfer pipe, the USBHS sets the associated NRDYSTS.PIPENRDY flag to 1, changes the associated PID[1:0] setting for the pipe to 00b (NAK response), and clears the CSSTS flag to 0. On generating this condition for the pipe for isochronous transfers, the USBHS sets the associated NRDYSTS.PIPENRDY flag for the pipe to 1, CRCE flag to 1, and clears the CSSTS bit to 0. It does not change the PID[1:0] setting.
- During a complete-split transaction, when the STALL handshake is received for a non-isochronous transfer pipe. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for the pipe to 1, changes the associated PID[1:0] setting for the pipe to 11b (STALL response), and clears the CSSTS flag to 0.
- During a complete-split transaction, when the NYET handshake is received for an isochronous transfer or interrupt transfer pipe for the microframe number = 4. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for each pipe to 1 and the CRCE flag to 1, and clears the CSSTS flag to 0. It does not change the PID[1:0] setting.

(3) In device controller mode

For transmitting pipes

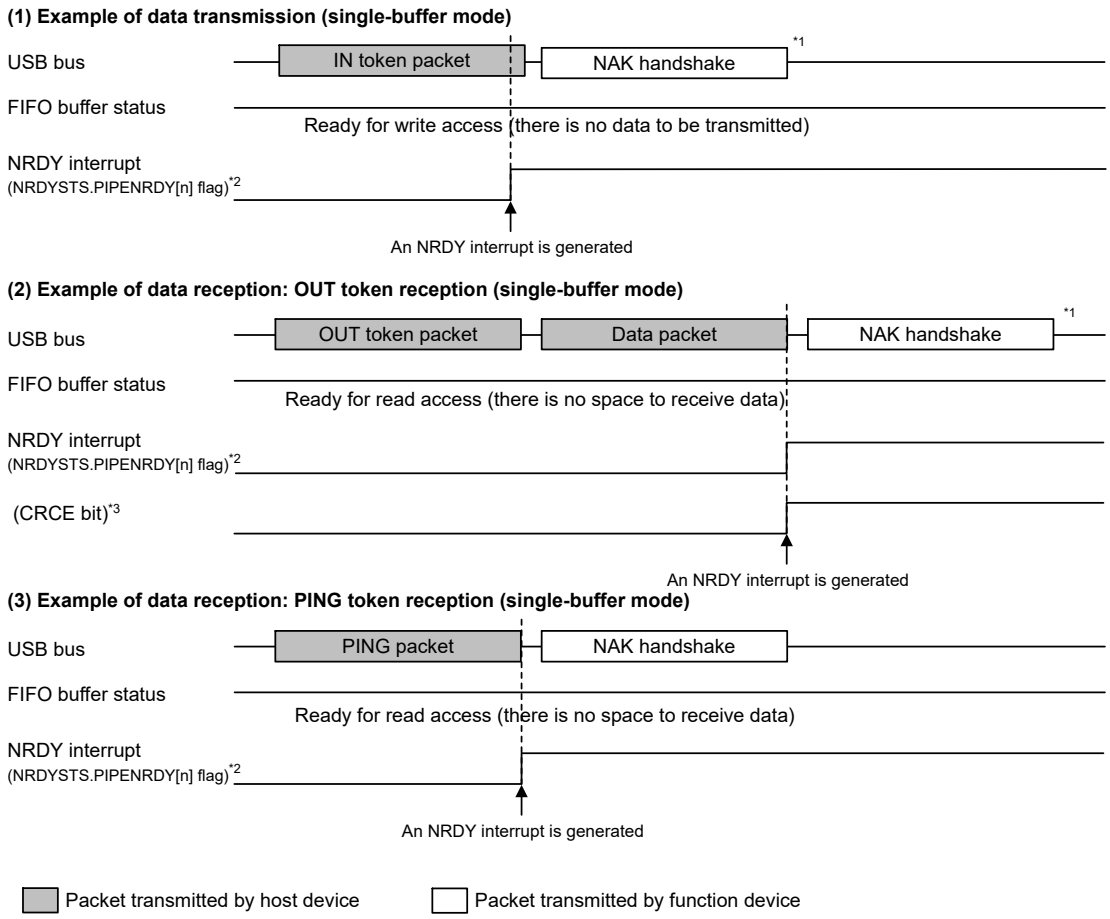
- When an IN token is received while there is no data to be transmitted in the FIFO buffer. In this case, the USBHS generates an NRDY interrupt request on reception of the IN token and sets the NRDYSTS.PIPENRDY flag to 1. For an isochronous transfer pipe in which an interrupt is generated, the USBHS transmits a zero-length packet and sets the FRMNUM.OVRN flag to 1.

For receiving pipes

- When an OUT token is received but there is no space available in the FIFO buffer. For an isochronous transfer pipe in which an interrupt is generated, the USBHS generates an NRDY interrupt request on reception of the OUT token and sets the NRDYSTS.PIPENRDY flag to 1 and the FRMNUM.OVRN flag to 1. For a non-isochronous transfer pipe in which an interrupt is generated, the USBHS generates an NRDY interrupt request when an NAK handshake is transferred after the data following the OUT token is received, and sets the NRDYSTS.PIPENRDY flag to 1. The NRDY interrupt request is not generated during retransmission because of a DATA-PID mismatch. In addition, the NRDY interrupt request is not generated if an error occurs in the DATA packet.
- On receiving a PING token when there is no space available in the FIFO buffer. The USBHS generates an NRDY interrupt request on reception of the PING token, setting the NRDYSTS.PIPENRDY flag to 1.

- For isochronous transfer pipes, when a token is not received successfully within an interval frame. In this case, the USBHS generates an NRDY interrupt request when the SOF is received, and sets the NRDYSTS.PIPENRDY flag to 1.

Figure 30.6 shows the timing of NRDY interrupt generation in device controller mode.



Note 1. The handshake is not used in isochronous transfers.
Note 2. The PIPENRDY[n] flag is changed to 1 only when PIPEnCTR.PID[1:0] bits are set to 01b (BUF response).
Note 3. The CRCE and OVRN flags change only while the target pipe is set to isochronous transfers.

Figure 30.6 Timing of NRDY interrupt generation in device controller mode

30.3.6.3 BEMP interrupt

On detecting a BEMP interrupt for the pipe whose PID[1:0] bits in the pipe control register are set to 01b (BUF response) by software, the USBHS sets the associated BEMPSTS.PIPEBEMP flag to 1. If the associated BEMPENB bit is set to 1 by software, the USBHS sets the INTSTS0.BEMP flag to 1 and generates a USB interrupt. This section describes the conditions in which the USBHS generates an internal BEMP interrupt request.

(1) For transmitting pipes

When the FIFO buffer of the associated pipe is empty on completion of transmission, including zero-length packet transmission, and in single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for a non-DCP pipe. The internal BEMP interrupt request is not generated in any of the following conditions:

- When the CPU or DMAC/DTC has already started writing data to the FIFO buffer of the CPU on completion of transmitting data from one FIFO buffer in double buffer mode

- When the buffer is cleared (emptied) by setting 1 to the PIPEnCTR.ACLRM or the BCLR bit in the port control register
- When an IN transfer (zero-length packet transmission) is performed during the control transfer status stage in device controller mode

(2) For receiving pipes

When a successfully-received data packet size exceeds the specified maximum packet size. In this case, the USBHS generates a BEMP interrupt request, sets the associated BEMPSTS.PIPEBEMP flag to 1, discards the received data, and changes the associated PID[1:0] setting for the pipe to STALL (11b). The USBHS returns no response in host controller mode, and returns STALL response in device controller mode.

The internal BEMP interrupt request is not generated in any of the following conditions:

- When a CRC error or a bit stuffing error is detected in the received data
- When a setup transaction is being performed:
  - Writing 0 to the BEMPSTS.PIPEBEMP flag clears the status
  - Writing 1 to the BEMPSTS.PIPEBEMP flag has no effect

Figure 30.7 shows the timing of BEMP interrupt generation in device controller mode.

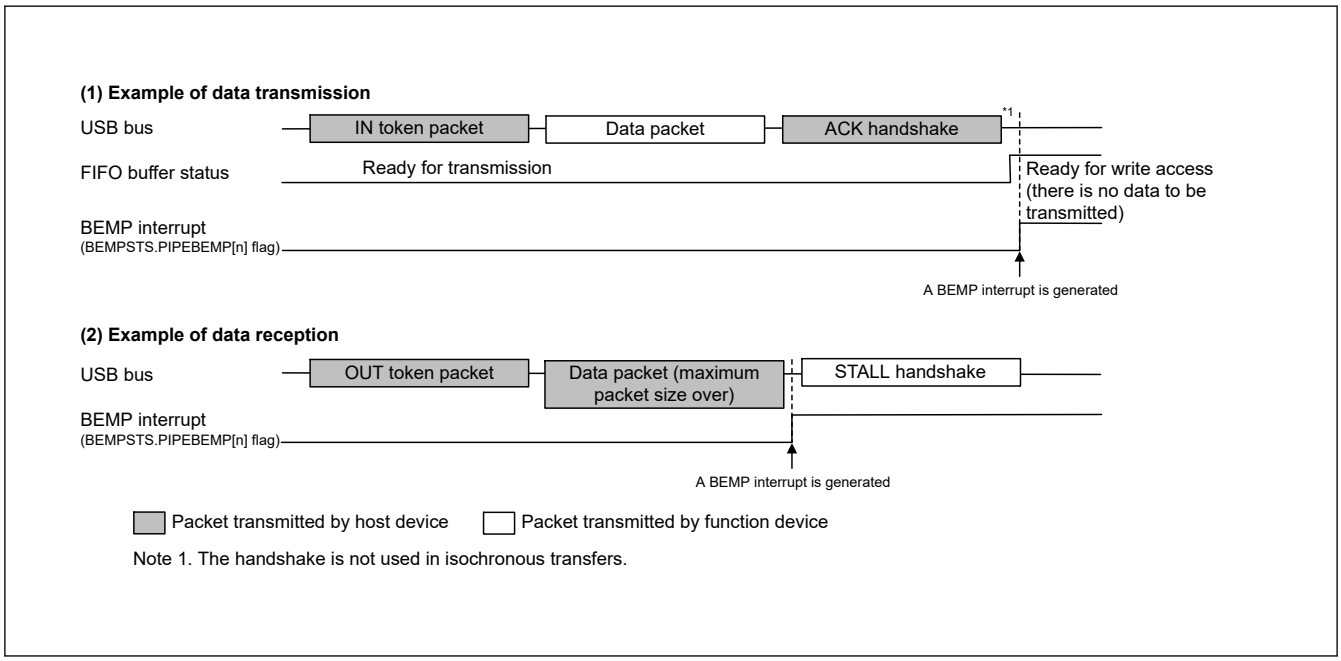


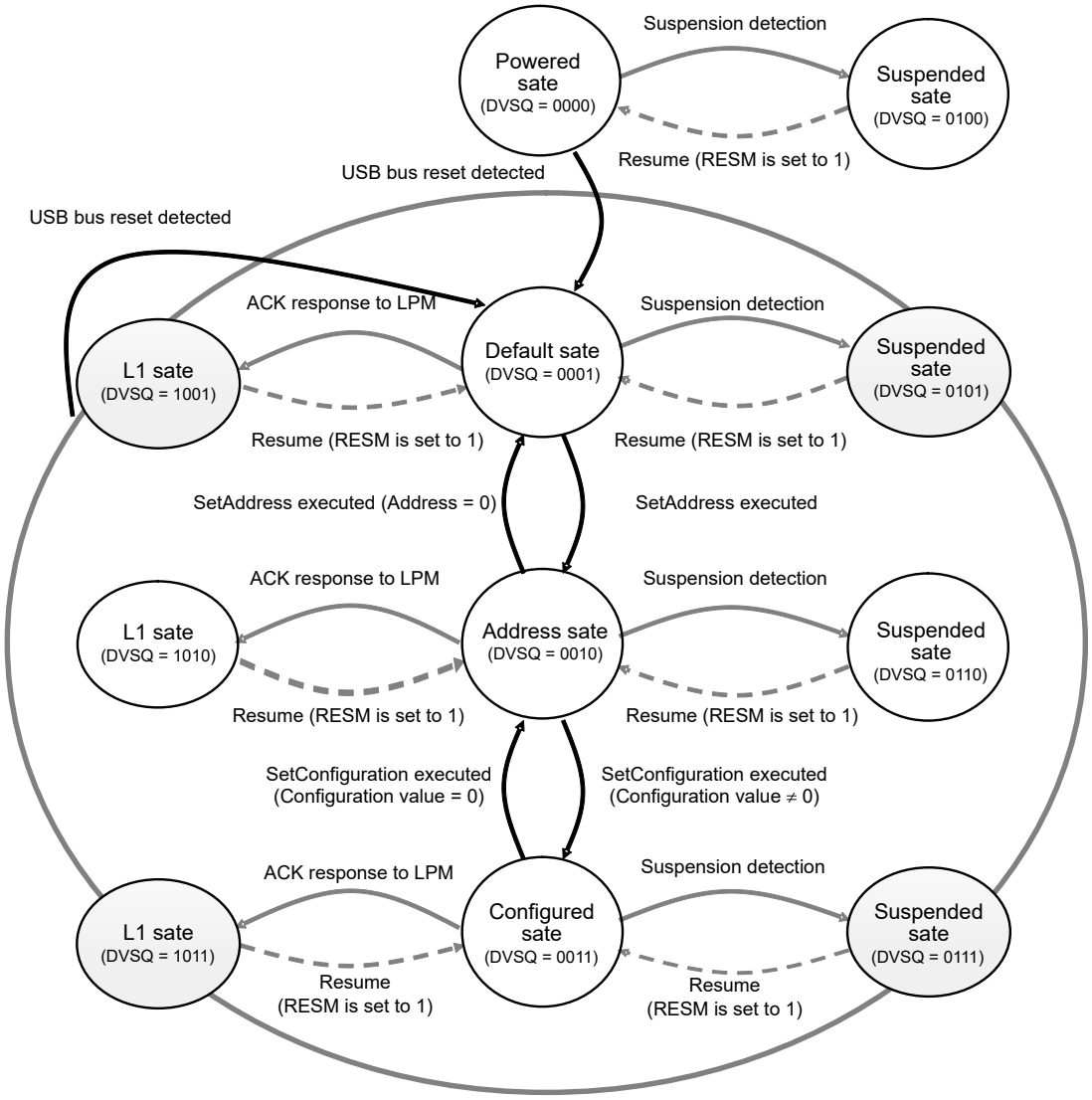
Figure 30.7 Timing of BEMP interrupt generation in device controller mode

30.3.6.4 Device state transition interrupt (device controller mode)

Figure 30.8 shows a diagram of the USBHS device state transitions. The USBHS controls device states and generates device state transition interrupts. However, recovery from the Suspend state (resume signal detection) is detected by means of the resume interrupt. Device state transition interrupts can be enabled or disabled independently in INTENB0. Devices whose states have changed can be checked in the PL1CTRL.DVSQ[3:0] flags.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

The USBHS controls device states, and device state transition interrupts can be generated, only in device controller mode.



Note 1. For a transition, indicated by a solid line, the DVST bit is set to 1. For a resume, indicated by a dashed line, the RESM bit is set to 1.

Figure 30.8 Device state transitions

30.3.6.5 Control transfer stage transition interrupt (device controller mode)

Figure 30.9 shows a diagram of the control transfer stage transitions of the USBHS. The USBHS controls the control transfer sequence and generates control transfer stage transition interrupts. Control transfer stage transition interrupts can be enabled or disabled independently in INTENB0. Transfer stages that have transitioned can be checked in the INTSTS0.CTSQ[2:0] bits.

Control transfer stage transition interrupts are generated only in device controller mode. This section describes control transfer sequence errors. When an error occurs, the DCPCTR.PID[1:0] bits are set to 1xb (STALL response).

- Control read transfer errors
  - An OUT token or PING token is received but no data is transferred in response to the IN token at the data stage
  - An IN token is received at the status stage

– A data packet with DATAPID = DATA0 is received at the status stage

- Control write transfer errors
  – An IN token is received but no ACK returned in response to the OUT token in the data stage
  – A data packet with DATAPID = DATA0 is received as the first data packet at the data stage
  – An OUT token or PING token is received in the status stage
- Control write no data transfer errors
  – An OUT token or PING token is received at the status stage

At the control write transfer data stage, if the receive data length exceeds the wLength value of the USB request, it is not recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (INTSTS0.CTRT flag = 1), the CTSQ[2:0] = 110b value is saved until CTRT flag clears to 0, clearing the interrupt status. While CTSQ[2:0] bits = 110b is being saved, no CTRT interrupt for ending the setup stage is generated, even if a new USB request is received. The USBHS saves the setup stage completion status, and it generates a CTRT interrupt after the interrupt status is cleared by software.

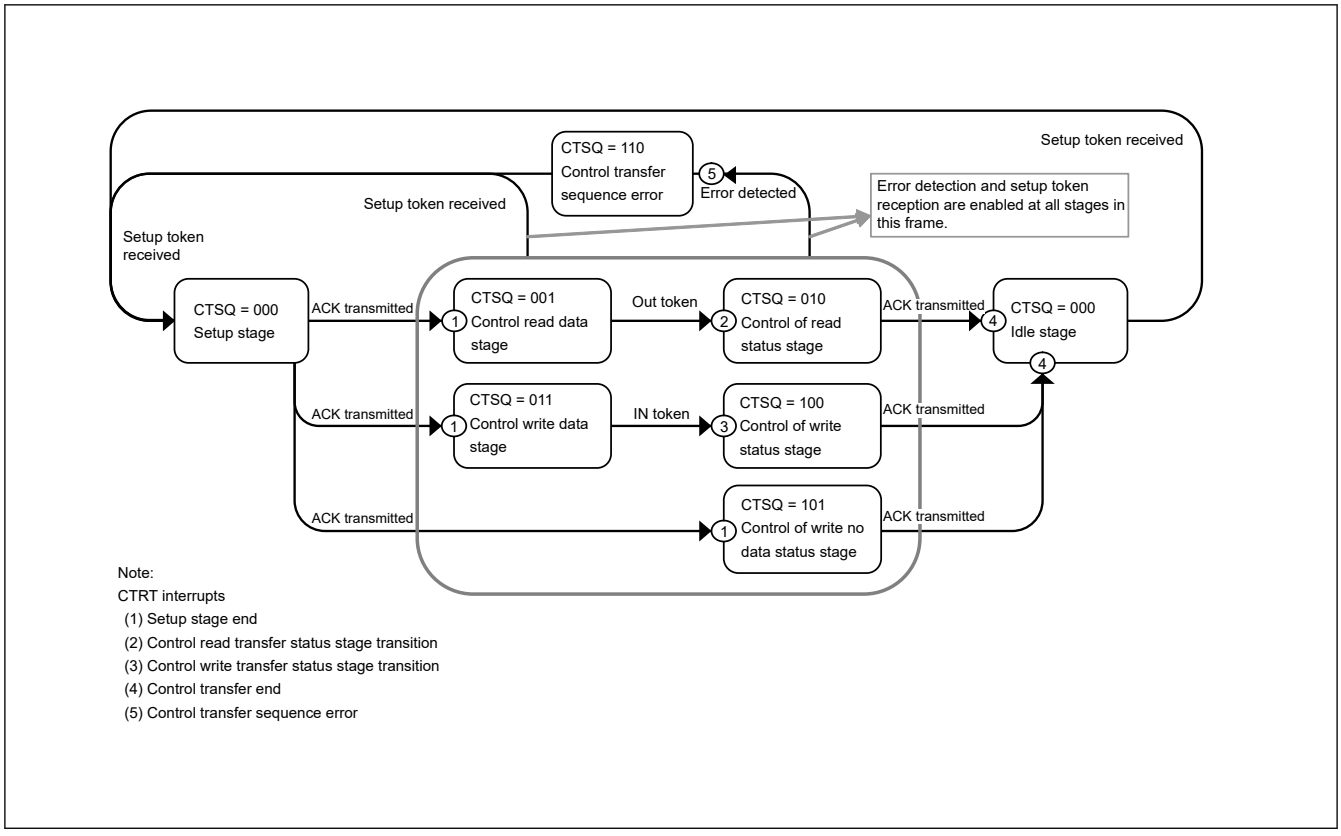


Figure 30.9 Control transfer stage transitions

30.3.6.6 Frame update interrupt

In host controller mode, an interrupt is generated when the frame number is updated.

In device controller mode, an SOFR interrupt is generated when the frame number is updated. The USBHS updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

30.3.6.7 VBUS interrupt

When the USBHS_VBUS pin level changes, a VBUS interrupt is generated. The level of the USBHS_VBUS pin can be checked in the INTSTS0.VBSTS flag. Whether the host controller is connected or disconnected can be confirmed using

the VBUS interrupt. If the system is activated with the host controller connected, the first VBUS interrupt is not generated, because there is no change in the USBHS_VBUS pin level.

30.3.6.8 Resume interrupt

In device controller mode, a resume interrupt is generated when the device is in the Suspend state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the Suspend state is detected by means of the resume interrupt.

In host controller mode, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

30.3.6.9 OVRCR interrupt

An OVRCR interrupt is generated when the USBHS_OVRCURA, USBHS_OVRCURA-DS, USBHS_OVRCURB or USBHS_OVRCURB-DS pin level has changed. The levels of the USBHS_OVRCURA, USBHS_OVRCURA-DS, USBHS_OVRCURB or USBHS_OVRCURB-DS pins can be checked in the SYSSTS0.OVCMON[1:0] flags. The external power supply IC can check whether overcurrent is detected using the OVRCR interrupt.

For OTG connections, the OVRCR interrupt allows you to check whether a change is detected in the VBUS comparator.

30.3.6.10 BCHG interrupt

A BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether a peripheral device is connected. It can also be used to detect a remote wakeup in host controller mode. The BCHG interrupt is generated in both host and device controller modes.

30.3.6.11 DTCH interrupt

A DTCH interrupt occurs when a USB bus disconnect is detected in host controller mode. The USBHS detects bus disconnects in compliance with the USB 2.0 specification.

On interrupt detection, all pipes in which communications are being carried out for the relevant port must be terminated by software. The pipes enter the wait state for a bus connection to the port, waiting for an ATTCH interrupt to occur. Regardless of the value set in the associated interrupt enable bit, the USBHS hardware:

- Sets the DVSTCTR0.UACT bit for the port in which the DTCH interrupt is detected to 0
- Puts the port in which the DTCH interrupt occurred into the idle state

30.3.6.12 SACK interrupt

A SACK interrupt is generated when an ACK response for the transmitted setup packet is received from the peripheral device in host controller mode. The SACK interrupt can be used to confirm that the setup transaction is successfully complete.

30.3.6.13 SIGN interrupt

A SIGN interrupt is generated when an ACK response for the transmitted setup packet is not correctly received from the peripheral device three consecutive times in host controller mode. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

30.3.6.14 ATTCH interrupt

An ATTCH interrupt is generated when J-state or K-state of the full-speed signal level is detected on the USB port for 2.5 µs in host controller mode. To be more specific, an ATTCH interrupt is detected in any of the following conditions:

- When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5 µs
- When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 µs

30.3.6.15 EOFERR interrupt

An EOFERR interrupt occurs when the USBHS detects that communication is not complete at the EOF2 timing defined in the USB 2.0 specification.

On interrupt detection, all pipes in which communications are being carried out for the relevant port must be terminated by software, and the port must be re-enumerated. Regardless of the value set in the associated interrupt enable bit, the USBHS hardware:

- Sets the DVSTCTR0.UACT bit for the port in which the EOFERR interrupt is detected to 0
- Puts the port in which the EOFERR interrupt is generated into the idle state

30.3.6.16 PDDETINT interrupt

The USBHS sets the INTSTS1.PDDETINT flag to 1 on detecting a level change (high to low or low to high) in the PDDET pin input value and generates the PDDETINT interrupt. When the PDDETINT interrupt is generated, use software to repeatedly read the BCCTRL.PDDETSTS flag until the same value is read three or more times, and perform debounce processing.

30.3.6.17 LPMEND interrupt

When the LPM transaction ends because a response from the peripheral device or a timeout is detected, the INTSTS1.LPMEND flag sets to 1 and the LPMEND interrupt is generated.

30.3.6.18 L1RSMEND interrupt

When performing resume processing when the USBHS has transitioned to the L1 state because an ACK is received in response to an LPM token, the USBHS sets the INTSTS1.L1RSMEND flag to 1 on completion of the resume processing.

30.3.7 Pipe Control

Table 30.22 lists the pipe settings for the USBHS. USB data transfer is performed through logical pipes that the software associates with endpoints. The USBHS provides 10 pipes for data transfer. Set up the pipes based on your system specifications.

Table 30.22 Pipe settings (1 of 2)

Register name	Bit name	Setting	Notes
DCPCFG PIPECFG	TYPE[1:0]	Transfer type	Pipes 1 to 9: Settable
	BFRE	BRDY interrupt mode	Pipes 1 to 5: Settable
	DBLB	Double buffer select	Pipes 1 to 5: Settable
	CNTMD	Selection of continuous transfer or discontinuous transfer	Pipes 1, 2: Settable only for bulk transfers Pipes 3 to 5: Settable
	DIR	Transfer direction select	IN or OUT settable
	EPNUM[3:0]	Endpoint number	Pipes 1 to 9: Settable Set this number to a value other than 0000 when one or more pipes are used.
	SHTNAK	Selects disabled state for pipe when transfer ends	Pipes 1, 2: Settable only for bulk transfers Pipes 3 to 5: Settable
PIPEBUF	BUFSIZE	FIFO buffer size	DCP: Setting disabled (fixed to 256 bytes) Pipes 1 to 5: Settable up to 2 KB Pipes 6 to 9: Setting disabled (fixed to 64 bytes)
	BUFNMB	FIFO buffer number	DCP: Setting disabled (fixed to 0x0-0x3 area) Pipes 1 to 5: Setting disabled (0x8-0x87 area specifiable) Pipes 6 to 9: Setting disabled (fixed to 0x4-0x7 area)
DCPMAXP PIPEMAXP	DEVSEL[3:0]	Device select	Viewable only in host controller mode
	MXPS	Maximum packet size	Setting compliant with USB specification

Table 30.22 Pipe settings (2 of 2)

Register name	Bit name	Setting	Notes
PIPEPERI	IFIS	Buffer flush	Pipes 1, 2: Settable only for isochronous transfers Pipes 3 to 5: Setting disabled Pipes 6 to 9: Setting disabled
	IITV[2:0]	Interval counter	Pipes 1, 2: Settable only for isochronous transfers Pipes 3 to 5: Setting disabled Pipes 6 to 9: Settable only in host controller mode
DCPCTR PIPEnCTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit
	INBUFM	IN buffer monitor	Available only for pipes 1 to 5
	SUREQ	Setup request	Settable only for the DCP and controlled in host controller mode
	SUREQCLR	SUREQ clear	Settable only for the DCP and controlled in host controller mode
	CSCLR	CSSTS clear	Controllable only in host controller mode
	CSSTS	Split status check	Viewable only in host controller mode
	ATREPM	Auto response mode	Pipes 1 to 5: Settable only in device controller mode
	ACLRM	Auto buffer clear	Pipes 1 to 9: Settable
	SQCLR	Sequence clear	Clears the data toggle bit
	SQSET	Sequence set	Sets the data toggle bit
	SQMON	Sequence check	Monitors the data toggle bit
	PBUSY	PIPE busy check	—
	PID[1:0]	Response PID	—
PIPEnTRE	TRENB	Transaction count enable	Pipes 1 to 5: Settable
	TRCLR	Current transaction counter clear	Pipes 1 to 5: Settable
PIPEnTRN	TRNCNT	Transaction counter	Pipes 1 to 5: Settable

30.3.7.1 Pipe control register switching procedures

The following bits in the pipe control registers can be changed only when USB communication is prohibited (PID[1:0] bits are 00b (NAK response)). Figure 30.10 shows pipe control register switching procedures when USB communication is enabled (PID[1:0] bits are 00b (BUF response)).

Do not change the following registers and bits when USB communication is enabled (PID[1:0] bits are 01b (BUFresponse)):

- Bits in DCPCFG and DCPMAXP
- SQCLR and SQSET bits in DCPCTR
- Bits in PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI
- ATREPM, ACLRM, SQCLR, and SQSET bits in PIPEnCTR
- Bits in PIPEnTRE and PIPEnTRN
- Bits in DEVADDn (n = 0 to A)

To set the CSCLR bits and bits in DEVADDn (n = 0 to A), follow the procedures described in section 30.2. Register Descriptions.

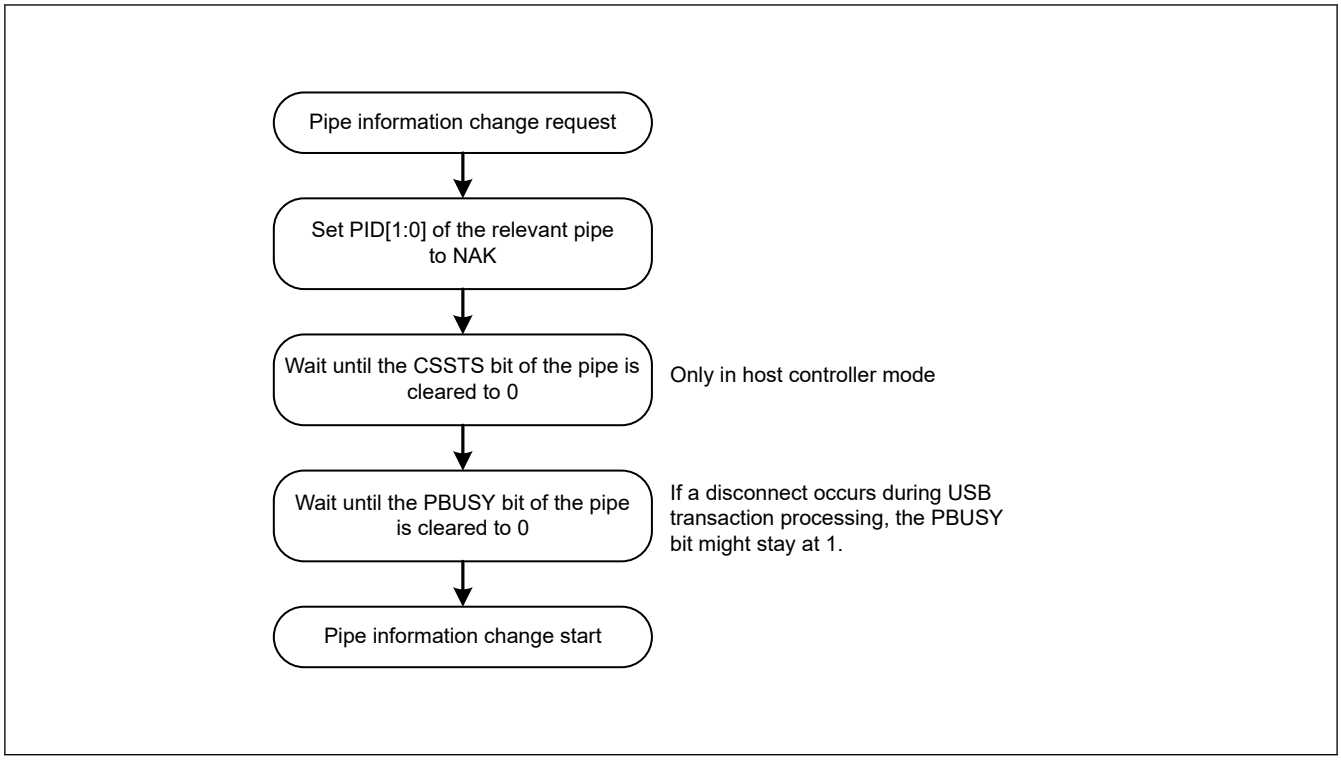


Figure 30.10 Procedure for changing pipe information when USB communication is enabled and PID[1:0] bits are 01b (BUF response)

The following bits in the pipe control registers can be changed only when the selected pipe information is not set in the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Do not set the following registers while the CURPIPE[3:0] bits are set:

- Bits in DCPCFG and DCPMAXP
- Bits in PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI
- PIPEnCTR and ACLRM bits

To change pipe information, you must set the CURPIPE[3:0] bits to a pipe other than the one to be changed. For the DCP, the buffer must be cleared using the BCLR bit after the pipe information is changed.

30.3.7.2 Transfer types

The PIPECFG.TYPE[1:0] bits specify the following transfer types for each pipe:

- DCP: No setting necessary (fixed at control transfer)
- Pipes 1 and 2: Set to bulk transfer or isochronous transfer
- Pipes 3 to 5: Set to bulk transfer
- Pipes 6 to 9: Set to interrupt transfer

30.3.7.3 Endpoint number

The PIPECFG.EPNUM[3:0] bits are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to 15.

- DCP: No setting is necessary (fixed at endpoint 0)
- Pipes 1 to 9: Select and set the endpoint numbers from 1 to 15 so that the combination of the PIPECFG.DIR and EPNUM[3:0] bits is unique

30.3.7.4 Maximum packet size setting

Specify the maximum packet size for each pipe in the MXPS bits in DCPMAXP and PIPEMAXP. The DCP and pipes 1 to 5 can be set to any of the maximum pipe sizes defined in the USB 2.0 specification. For pipes 6 to 9, the maximum packet size is 64 bytes. Set the maximum packet size as follows before starting a transfer (PID[1:0] bits are set to 01b (BUF response)):

- DCP: Set to 64 for high-speed operation
- DCP: Set to 8, 16, 32, or 64 for full-speed operation
- Pipes 1 to 5: Set to 512 for high-speed bulk transfers
- Pipes 1 to 5: Set to 8, 16, 32, or 64 for full-speed bulk transfers
- Pipes 1, 2: Set between 1 and 1024 for high-speed isochronous transfers
- Pipes 1, 2: Set between 1 and 1023 for full-speed isochronous transfers
- Pipes 6 to 9: Set between 1 and 64

High-bandwidth interrupt transfers and isochronous transfers are not supported.

30.3.7.5 Transaction counter for pipes 1 to 5 in the receiving direction

When the specified number of transactions is complete in the data packet receiving direction, the USBHS recognizes that the transfer has ended. Two transaction counters are provided. One is the PIPEnTRN register, which specifies the number of transactions to be executed, and the other is the current counter, which internally counts the number of executed transactions. If the PIPECFG.SHTNAK bit is set to 1, when the current counter value matches the specified number of transactions, the associated PIPEnCTR.PID[1:0] bits are set to 00b (NAK response) and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the PIPEnTRE.TRCLR bit. The data read from PIPEnTRN differs depending on the PIPEnTRE.TRENB setting as follows:

- The TRENB bit = 0: Specified transaction counter value can be read
- The TRENB bit = 1: Current counter value indicating the internally counted number of executed transactions can be read.

The following constraints apply when working with the TRCLR bit:

- If the transactions are being counted and the PIPEnCTR.PID[1:0] bits are set to 01b (BUF response), the current counter cannot be cleared
- If there is any data left in the buffer, the current counter cannot be cleared

30.3.7.6 Response PID

Specify the response PID for each pipe in the PID[1:0] bits in DCPCTR and PIPEnCTR. This section describes the USBHS operation with different response PID settings.

Software response PID settings in host controller mode

Select the response PID to specify the execution of transactions as follows:

- NAK setting: Using pipes is disabled and no transactions are executed
- BUF setting: Transactions are executed based on the FIFO buffer state:
  OUT direction: An OUT token is issued if the FIFO buffer contains transmit data.
  IN direction: An IN token is issued if the FIFO buffer is not full and can receive data.
- STALL setting: Using pipes is disabled and no transactions are executed

Note: Use the SUREQ bit to execute setup transactions for the DCP.

Software response PID settings in device controller mode

Select the response PID to respond as follows to transactions from the host:

- NAK setting: A NAK response is returned to all generated transactions

- BUF setting: A response is returned to transactions based on the FIFO buffer
- STALL setting: A STALL response is returned to all generated transactions

Note: For setup transactions, an ACK response is always returned, regardless of the PID[1:0] bits setting, and the USB request is stored in the register.

Hardware response PID settings in host controller mode and Hardware response PID settings in device controller mode describe situations in which the USBHS writes to the PID[1:0] bits because of specific transaction results.

Hardware response PID settings in host controller mode

- NAK setting: The PID[1:0] bits are set to 00b (NAK response) in the following cases, and issuing of tokens is automatically stopped:
  - When a non-isochronous transfer is performed and an NRDY interrupt is generated. For details, see section 30.3.6.2. NRDY interrupt
  - If a short packet is received when the PIPECFG.SHTNAK bit is set to 1 for bulk transfers
  - If transaction counting ends when the SHTNAK bit is set to 1 for bulk transfers
- BUF setting: The USBHS does not write this setting
- STALL setting: The PID[1:0] bits are set to STALL in the following cases, and issuing of tokens is automatically stopped:
  - When STALL is received in response to a transmitted token
  - When a received data packet exceeds the maximum packet size

Hardware response PID settings in device controller mode

- NAK setting: The PID[1:0] bits are set to 00b (NAK response) in the following cases, and a NAK response is returned to transactions:
  - When the setup token is received normally (DCP only)
  - If transaction counting ends or a short packet is received when the PIPECFG.SHTNAK bit is set to 1 for bulk transfers
- BUF setting: The USBHS does not write this setting
- STALL setting: The PID[1:0] bits are set to STALL in the following cases, and a STALL response is returned to transactions:
  - When a received data packet exceeds the maximum packet size
  - When a control transfer sequence error is detected

30.3.7.7 Data PID sequence bit

The USBHS automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON bit in DCPCTR and PIPEnCTR. When data is transmitted, the sequence bit toggles on ACK handshake reception. When data is received, the sequence bit toggles on ACK handshake transmission. The SQCLR bit in DCPCTR and the SQSET bit in PIPEnCTR can be used to change the data PID sequence bit.

In device controller mode when control transfers are used, the USBHS automatically sets the sequence bit for stage transitions. DATA1 is returned when the setup stage ends. The sequence bit is not referenced and PID = DATA1 is returned in the status stage. Therefore, no software settings are required. However, in host controller mode when control transfers are used, the sequence bit must be set by software for the stage transitions.

For ClearFeature requests for transmission or reception, the data PID sequence bit must be set by software in both host and device controller modes.

30.3.7.8 Response PID = NAK function

The USBHS provides a function for disabling pipe operation (PID[1:0] bits are set to 00b (NAK response)) when the final data packet of a transaction is received. The USBHS automatically distinguishes this based on reception of a short packet or the transaction counter. Enable this function by setting the PIPECFG.SHTNAK bit to 1.

When the double buffer mode is being used for the FIFO buffer, using this function enables reception of data packets in transfer units. If pipe operation is disabled, software must enable the pipe again (PID[1:0] bits are set to 01b (BUF response)).

The response PID = NAK function can be used only for bulk transfers.

30.3.7.9 Auto response mode

For bulk transfer pipes (1 to 5), when the PIPEnCTR.ATREPM bit is set to 1, a transition is made to auto response mode.

During an OUT transfer (PIPECFG.DIR = 0), OUT-NAK mode is invoked, and during an IN transfer (DIR = 1), null auto response mode is invoked.

30.3.7.10 OUT-NAK mode

For bulk OUT transfer pipes, NAK is returned in response to an OUT token, and an NRDY interrupt is output when the PIPEnCTR.ATREPM bit is set to 1. To transition from normal mode to OUT-NAK mode, specify OUT-NAK mode while pipe operation is disabled (PID[1:0] = 00b for NAK response). Next, enable pipe operation (PID[1:0] = 01b for BUF response), on which OUT-NAK mode becomes valid. If an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is returned to the host.

To transition from OUT-NAK mode to normal mode, cancel OUT-NAK mode while pipe operation is disabled (NAK).

Next enable pipe operation (BUF). In normal mode, reception of OUT data is enabled.

30.3.7.11 Null auto response mode

For bulk IN transfer pipes, zero-length packets are continuously transmitted when the PIPEnCTR.ATREPM bit is set to 1.

To transition from normal mode to null auto response mode, specify null auto response mode while pipe operation is disabled (PID[1:0] bits are set to 00b (NAK response)). Next, enable pipe operation (PID[1:0] bits are set to 01b (BUF response)), on which null auto response mode becomes valid. Before setting null auto response mode, check that PIPEnCTR.INBUFM = 0, because the mode can be set only when the buffer is empty. If the INBUFM bit is 1, empty the buffer using the PIPEnCTR.ACLRM bit. Do not write data from the FIFO port while a transition to null auto response mode is being made.

To transition from null auto response mode to normal mode, keep pipe operation disabled (PID[1:0] bits are set to 00b (NAK response)) for the period of the zero-length packet transmission (about 10 µs) before canceling the null auto response mode. In normal mode, data can be written from the FIFO port, so packet transmission to the host is enabled by enabling pipe operation (PID[1:0] bits are set to 01b (BUF response)).

30.3.8 FIFO Buffer

The USBHS provides a FIFO buffer for data transfers, and it manages the memory area used for each pipe. The FIFO buffer has two states depending on whether the access right is assigned to the system (CPU side) or the USBHS (SIE side).

30.3.8.1 Buffer status

Table 30.23 and Table 30.24 show the buffer status in the USBHS. The FIFO buffer status can be confirmed using the DCPCTR.BSTS and PIPEnCTR.INBUFM bits. The transfer direction for the FIFO buffer can be specified in the PIPECFG.DIR or CFIFOSEL.ISEL bit (when DCP is selected).

The INBUFM bit is valid for pipes 1 to 5 in the transmitting direction.

When a transmitting pipe uses double buffering, the software can read the BSTS bit to monitor the FIFO buffer status on the CPU side and the INBUFM bit to monitor the FIFO buffer status on the SIE side. When write access to the FIFO port by the CPU or DMAC/DTC is slow and the buffer empty status cannot be determined using the BEMP interrupt, the software can use the INBUFM bit to confirm the end of transmission.

Table 30.23 Buffer status indicated in the BSTS flag

ISEL or DIR	BSTS	FIFO buffer status
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is disabled.
0 (receiving direction)	1	There is received data, or a zero-length packet is received. Reading from the FIFO port is allowed. When a zero-length packet is received, reading is not possible and the buffer must be cleared.
1 (transmitting direction)	0	Transmission is not complete. Writing to the FIFO port is disabled.
1 (transmitting direction)	1	Transmission is complete. CPU write is allowed.

Table 30.24 Buffer status indicated in the INBUFM bit

DIR	INBUFM	FIFO buffer status
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	Transmission is complete. There is no data waiting to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted.

30.3.8.2 FIFO buffer clearing

Table 30.25 shows the methods for clearing the FIFO buffer. The FIFO buffer can be cleared using the BCLR bit in the port control register, DnFIFOSEL.DCLRM, or the PIPEnCTR.ACLRM bit.

Single or double buffering can be selected for pipes 1 to 5 in the PIPECFG.DBLB bit.

Table 30.25 Buffer clearing methods

FIFO buffer clearing mode	Clearing the FIFO buffer on the CPU side	Mode for automatically clearing the FIFO buffer after reading the specified pipe data	Auto buffer clear mode for discarding all received packets
Register used	• CFIFOCTR • DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Bit used	BCLR	DCLRM	ACLRM
Clearing condition	Cleared by writing 1	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

Auto buffer clear mode function

The USBHS discards all received data packets if the PIPEnCTR.ACLRM bit is set to 1. If a correct data packet is received, the ACK response is returned to the host controller. The auto buffer clear mode function can only be set in the FIFO buffer reading direction.

Setting the ACLRM bit to 1 and then to 0 clears the FIFO buffer of the selected pipe regardless of the access direction. An access cycle of at least 100 ns is required for the internal hardware sequence processing between ACLRM = 1 and ACLRM = 0.

30.3.8.3 FIFO port functions

Table 30.26 shows the settings for the FIFO port functions. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, set the BVAL flag in the port control register to end writing. To send a zero-length packet, use the BCLR bit to clear the buffer, and then set the BVAL flag to end writing.

In reading, reception of new packets is automatically enabled when all data is read. Data cannot be read when a zerolength packet is received (DTLN[11:0] = 0), so the buffer must be cleared with the BCLR bit. The length of the receive data can be confirmed in the DTLN[11:0] flags in the port control register.

Table 30.26 FIFO port function settings

Register name	Bit name	Description
CFIFOSEL, DnFIFOSEL (n = 0, 1)	RCNT	Selects DTLN[11:0] read mode
	REW	FIFO buffer rewind (re-read, rewrite)
	DCLRM	Automatically clears receive data for a specified pipe after the data is read (only for DnFIFO)
	DREQE	Enables DMAC/DTC transfers (only for DnFIFO)
	MBW[1:0]	FIFO port access bit width
	BIGEND	Selects FIFO port endian
	ISEL	FIFO port access direction (only for DCP)
	CURPIPE[3:0]	Selects the current pipe
CFIFOCTR, DnFIFOCTR (n = 0, 1)	BVAL	Ends writing to the FIFO buffer
	BCLR	Clears the FIFO buffer on the CPU side
	DTLN[11:0]	Checks the length of receive data

FIFO port selection

Table 30.27 shows the pipes that can be selected with the different FIFO ports. The pipe to be accessed must be selected in the CURPIPE[3:0] bits in the port selection register. After a pipe is selected, the software must check whether the written value can be correctly read from the CURPIPE[3:0] bits. (If the previous pipe number is read, it indicates that the USBHS is modifying the pipe.) Next, the software checks that the FRDY flag in the port control register is 1.

In addition, the software must specify the bus width to be accessed in the MBW[1:0] bits in the port selection register.

The FIFO buffer access direction conforms to the PIPECFG.DIR setting. For the DCP only, the ISEL bit in the port selection register determines the direction.

Table 30.27 FIFO port access by pipe

Pipe	Access Method	Port that can be used
DCP	CPU access	CFIFO port register
Pipes 1 to 9	CPU access	CFIFO port register D0FIFO/D1FIFO port register
	DMAC/DTC access	D0FIFO/D1FIFO port register

REW bit

It is possible to temporarily stop access to a pipe being accessed, access a different pipe, and then continue processing for the first pipe again. The REW bit in the port selection register is used for this processing.

If a pipe is selected in the CURPIPE[3:0] bits in the port selection register with the REW bit set to 1, the pointer used for reading from and writing to the FIFO buffer is reset, and reading or writing can be carried out from the first byte. If a pipe is selected with the REW bit set to 0, data can be read and written in continuation from the previous selection, without the pointer being reset.

To access the FIFO port, the software must check that the FRDY bit in the port control register is 1 after selecting a pipe.

30.3.8.4 DMAC/DTC transfers (D0FIFO and D1FIFO ports)

Overview of DMAC/DTC transfer

For pipes 1 to 9, the FIFO port can be accessed using the DMAC/DTC. When buffer access for the pipe targeted for DMAC/DTC transfer is enabled, a DMAC/DTC transfer request is issued.

Select the unit of transfer to the FIFO port in the DnFIFOSEL.MBW[1:0] bits, and select the pipe targeted for the DMAC/DTC transfer in the DnFIFOSEL.CURPIPE[3:0] bits. Do not change the selected pipe during the DMAC transfer.

DnFIFO auto clear mode (D0FIFO and D1FIFO port reading direction)

If 1 is set in the DnFIFOSEL.DCLRM bit, the USBHS automatically clears the FIFO buffer of the selected pipe when reading of data from the FIFO buffer is complete.

Table 30.28 shows the packet reception and FIFO buffer clearing processing by software for each of the different settings. As shown in the table, the buffer clearing conditions depend on the value set in the PIPECFG.BFRE bit. Using the DnFIFOSEL.DCLRM bit eliminates the need for the buffer to be cleared by software in any situation that requires buffer clearing. This enables DMAC/DTC transfers without involving software.

The DnFIFO auto clear mode can only be set in the FIFO buffer reading direction.

Table 30.28 Packet reception and FIFO buffer clearing processing by software

	Register setting			
	DCLRM = 0		DCLRM = 1	
Buffer status when packet is received	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	No clearing required	No clearing required	No clearing required	No clearing required
Zero-length packet reception	Clearing required	Clearing required	No clearing required	No clearing required
Normal short packet reception	No clearing required	Clearing required	No clearing required	No clearing required
Transaction count end	No clearing required	Clearing required	No clearing required	No clearing required

30.3.8.5 Allocating the FIFO buffer

Figure 30.11 shows an example of a memory map of the FIFO buffer. The FIFO buffer is an area shared by the USBHS and the control CPU of the application. There are two situations for the FIFO buffer: (1) access rights are given to the application (CPU side), and (2) access rights are given to the USBHS (SIE side).

An independent area is set for the FIFO buffer for each pipe. A memory area is determined by the first block number and the number of blocks (specified in the BUFNMB[7:0] and BUFSIZE[4:0] bits in PIPEBUF), where 64 bytes is regarded as one block. When the continuous transfer mode is selected in the CNTMD bit in PIPECFG, set the BUFSIZE[4:0] bits to an integral multiple of the maximum packet size. When double buffering is selected in the DBLB bit in PIPECFG, twice the memory area specified in the BUFSIZE[4:0] bits in PIPEBUF is allocated to the same pipe.

Three FIFO ports are used to access (read data from and write data to) the FIFO buffer. Specify the number of the pipe to be allocated to the FIFO port in the CURPIPE[3:0] bits in C/DnFIFOSEL.

The FIFO buffer status of each pipe can be checked in the DCPCTR.BSTS, PIPEnCTR.BSTS, and PIPEnCTR.INBUFM bits. The FIFO port access rights can be checked in the FRDY flag in C/DnFIFOCTR.

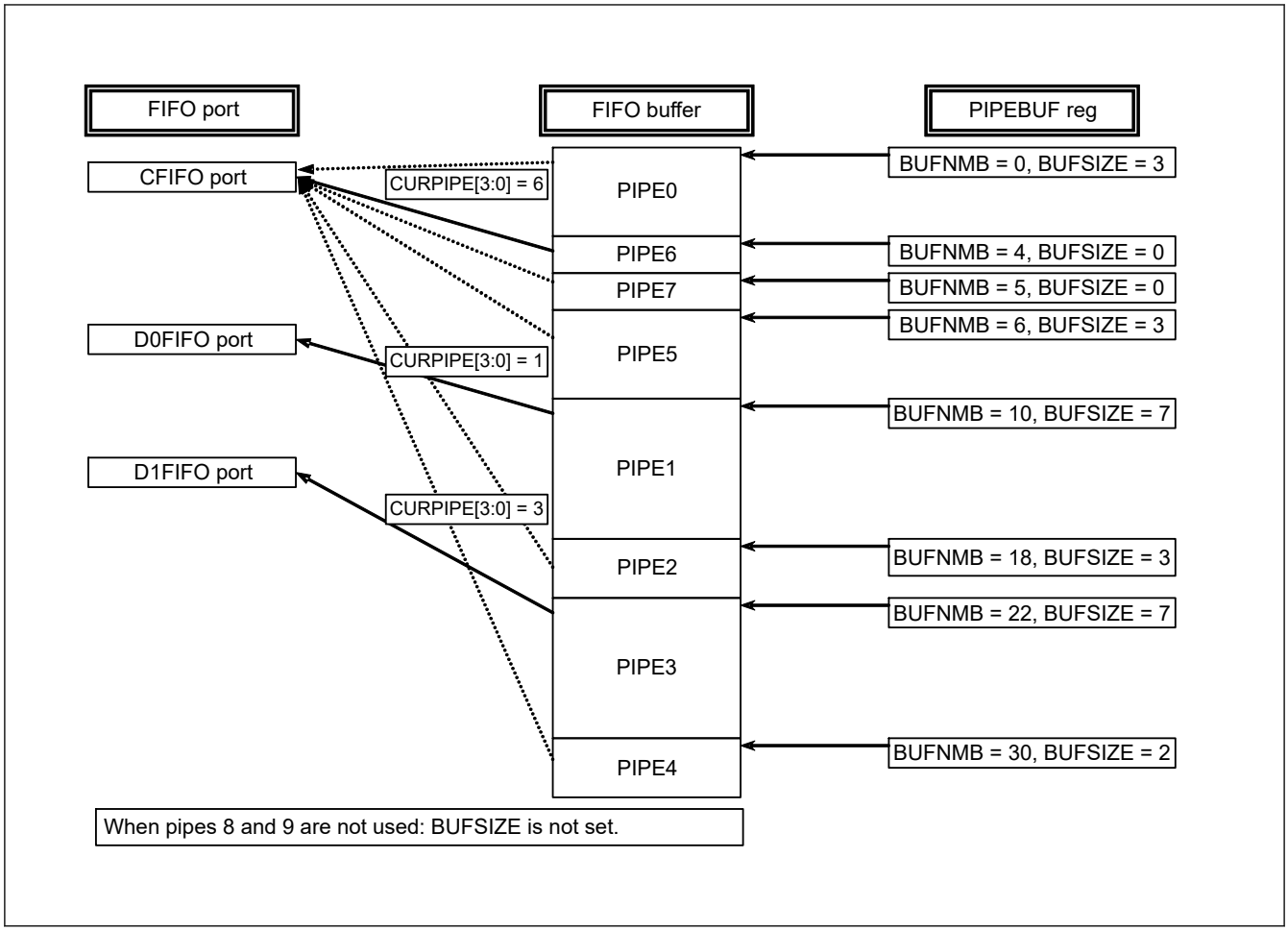


Figure 30.11 Example memory map of the FIFO buffer

30.3.9 Control Transfers Using the DCP

The Default Control Pipe (DCP) is used for data transfers in the control transfer data stage. The FIFO buffer of the DCP is a 64-byte single buffer with a fixed area for both control reads and control writes. The FIFO buffer can be accessed only through the CFIFO port.

30.3.9.1 Control transfers in host controller mode

Setup stage

The USBREQ, USBVAL, USBINDX, and USBLENG registers are used to transmit USB requests for setup transactions. Writing the setup packet data to the register and then writing 1 to the DCPCTR.SUREQ bit transmits the specified data for the setup transaction. On completion of the transaction, the SUREQ bit clears to 0. Do not change these USB request registers while SUREQ = 1.

When an attached function device is detected, the software must issue the first setup transaction for the device using this sequence with the DCPMAXP.DEVSEL[3:0] bits cleared to 0 and the DEVADD0.USBSPD[1:0] bits set appropriately.

When an attached function device is shifted to the Address state, the software must issue setup transactions using this sequence with the assigned USBAddress set in the DEVSEL[3:0] bits and the bits in DEVADDn (n = 0 to A) corresponding to the specified USBAddress set appropriately. For example, when PIPEMAXP.DEVSEL[3:0] = 0010b, make appropriate settings in DEVADD2. When PIPEMAXP.DEVSEL[3:0] = 0101b, make appropriate settings in DEVADD5.

When the setup transaction data is sent, an interrupt request is generated based on the response from the peripheral device (SIGN or SACK bit in INTSTS1). This interrupt request allows the software to check the setup transaction result.

The DATA0 data packet (USB request) for the setup transaction is always transmitted regardless of the status of the DCPCTR.SQMON flag.

Data stage

The data stage is used to transfer data using the DCP FIFO buffer.

Before accessing the DCP FIFO buffer, specify the access direction in the CFIFOSEL.ISEL bit. Specify the transfer direction in the DCPCFG.DIR bit.

For the first data packet of the data stage, the data PID must be transferred as DATA1. Set data PID to DATA1 in the DCPCTR.SQSET bit and set the PID[1:0] bits to 01b (BUF response). Completion of data transfer is detected using the BRDY or BEMP interrupt.

Data transfer of multiple packets is enabled in continuous transfer mode. However, when continuous transfer is specified in the receiving direction, the BRDY interrupt is not generated unless the buffer becomes full or a short packet is received (for 256 bytes or less, which is an integral multiple of the maximum packet size). If the transmit data size is an integral multiple of the maximum packet size, control the control write transfer through the software to transmit a zerolength packet last.

Status stage

The status stage is used for zero-length packet data transfers in the reverse direction of the data stage. As in the data stage, data is transferred using the DCP FIFO buffer. Transactions are executed using the same procedure as the data stage.

Data packets in the status stage must be transmitted and received with the data PID set to DATA1 using the DCPCTR.SQSET bit.

When a zero-length packet is received, check the receive-data length in the CFIFOCTR.DTLN[11:0] flags after a BRDY interrupt is generated, and then clear the FIFO buffer using the BCLR bit.

30.3.9.2 Control transfers in device controller mode

Setup stage

The USBHS returns an ACK response to a normal setup packet for the USBHS. The USBHS operates in the setup stage as follows:

On receiving a new setup packet, the USBHS sets the following bits:

- Sets the INTSTS0.VALID flag to 1
- Sets the DCPCTR.PID[1:0] bits to 00b (NAK response)
- Sets the DCPCTR.CCPL bit to 0

When the USBHS receives a data packet following a setup packet, it stores the USB request parameters in USBREQ, USBVAL, USBINDX, and USBLENG.

Before performing the response processing for a control transfer, set the VALID flag to 0. When the VALID flag = 1, the PID[1:0] bits cannot be set to 01b (BUF response), and the data stage cannot be terminated.

Using the VALID flag function, the USBHS can suspend a request being processed when it receives a new USB request during a control transfer and return a response to the latest request.

In addition, the USBHS automatically detects the direction bit (bmRequestType bit 8) and the request data length (wLength) in the received USB request. It distinguishes between control read transfers, control write transfers, and nodata control transfers, and it controls stage transitions. For an incorrect sequence, a sequence error occurs in the control transfer stage transition interrupt, and the interrupt is reported to the software. For a diagram of the stage control by the USBHS, see Figure 30.9.

Data stage

The DCP must be used to execute data transfers for received USB requests. Before accessing the DCP FIFO buffer, specify the access direction in the CFIFOSEL.ISEL bit.

If the transfer data is larger than the size of the DCP FIFO buffer, execute the data transfer using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

In high-speed control write transfers, a NYET handshake response is returned based on the FIFO buffer status.

Status stage

Control transfers are terminated by setting the DCPCTR.CCPL bit to 1 while the DCPCTR.PID[1:0] bits are set to 01b (BUF response).Control transfers are terminated by setting the DCPCTR.CCPL bit to 1 while the DCPCTR.PID[1:0] bits are set to 01b (BUF response).

After this setting is made, the USBHS automatically executes the status stage based on the data transfer direction determined at the setup stage. The status stage is executed as follows:

- For control read transfers:
  The USBHS receives a zero-length packet from the USB host and transmits an ACK response
- For control write transfer and no data control transfer:
  The USBHS transmits a zero-length packet and receives an ACK response from the USB host.

Control transfer auto response function

The USBHS automatically responds to a normal SET_ADDRESS request. If the SET_ADDRESS request contains any of the following errors, a response must be returned by software.

- When bmRequestType is not 0x00: except control write transfer
- When wIndex is not 0x00: request error
- When wLength is not 0x00: except no data control transfer
- When wValue is larger than 0x7F: request error
- When PL1CTRL.DVSQ[3:0] flags are 0011b (Configured): control transfer of device state error

A response by the corresponding software is required to all requests other than SET_ADDRESS.

30.3.10 Bulk Transfers (Pipes 1 to 5)

The FIFO buffer usage (setting of single buffer/double buffer or continuous/discontinuous transfer mode) is configurable for bulk transfers. The FIFO buffer size can be set up to 2 KB. The USBHS manages the FIFO buffer state and automatically responds to the PING packet and the NYET handshake.

30.3.10.1 PING packet control in host controller mode

In the OUT direction, a PING packet is automatically transmitted by the USBHS. The USBHS starts communication in the transmitting direction beginning with the PING packet. When it receives an ACK handshake in response to the PING packet, the USBHS transmits an OUT packet. The USBHS returns to the PING transmission state on receiving a NAK or NYET response during an OUT transaction. The procedure is as follows:

Starting OUT data transmission

(1) Transmit PING packet
(2) Receive NAK handshake
(3) Transmit PING packet
(4) Receive ACK handshake
(5) Transmit OUT data packet
(6) Receive ACK handshake
(7) Transmit OUT data packet
⋮
(8) Return to (1) on receiving a NAK/NYET handshake

The USBHS returns to the PING packet transmission state when a hardware reset is issued, the NYET or NAK handshake is received, the sequence toggle bit is cleared (SQCLR), or the buffer clear bit (ACLRM) is set.

30.3.10.2 NYET handshake control in device controller mode

Table 30.29 lists responses to received tokens during bulk and control transfers. The USBHS returns a NYET response when an available area for only one packet is left in the FIFO buffer when the USBHS has received an OUT token during a bulk or control transfer. When the USBHS receives a short packet, however, it returns an ACK response instead of NYET even when this condition occurs.

Table 30.29 Responses to received tokens

PID[1:0] bit setting	FIFO buffer state	Received token	Response	Note
NAK/STALL	—	SETUP	ACK	—
	—	IN/OUT/PING	NAK/STALL	—
BUF	—	SETUP	ACK	—
	RCV-BRDY	OUT/PING	ACK	When OUT token is received, data packet is received.*1
	RCV-BRDY	OUT	NYET	Data packet is received*2
	RCV-BRDY	OUT (Short)	ACK	Data packet is received*2
	RCV-BRDY	PING	ACK	*2
	RCV-NRDY	OUT/PING	NAK	—
	TRN-BRDY	IN	DATA0/1	Data packet is transmitted
	TRN-NRDY	IN	NAK	—

Note 1. RCV-BRDY: An available area for two packets is left in the FIFO buffer when an OUT token or a PING token is received.
Note 2. RCV-BRDY: An available area for only one packet is left in the FIFO buffer when an OUT token is received.
RCV-NRDY: No available area is left in the FIFO buffer when a PING token is received.
TRN-BRDY: The FIFO buffer contains transmit data when an IN token is received.
TRN-NRDY: The FIFO buffer contains no transmit data when an IN token is received.

30.3.11 Interrupt Transfers (Pipes 6 to 9)

In device controller mode, the USBHS performs interrupt transfers based on the timing dictated by the host controller. In the interrupt transfer, the USBHS ignores PING packets (no response) and does not transmit the NYET handshake, but returns an ACK, NAK, or STALL response.

In host controller mode, the software can set the timing for issuing tokens using the interval counter. The USBHS does not issue a PING token but issues an OUT token, including for transfers in the OUT direction.

The USBHS does not support high-bandwidth interrupt transfers.

30.3.11.1 Interval counter for interrupt transfers in host controller mode

Specify the transaction interval for interrupt transfers in the PIPEPERI.IITV[2:0] bits. The USBHS issues interrupt transfer tokens based on this interval.

(1) Initializing the counter

The USBHS initializes the interval counter under the following conditions:

- Power-on reset:
  This initializes the IITV[2:0] bits
- FIFO buffer initialization using the PIPEnCTR.ACLRM bit:
  This does not initialize the IITV[2:0] bits, but does initialize the count value. Setting the PIPEnCTR.ACLRM bit to 0 starts counting from the value set in IITV[2:0].

(2) Operation when tokens cannot be transmitted or received even on token generation

No token is generated in the following cases even at token generation time. In these cases, the USBHS tries to execute the transaction in the next interval.

- When the PID[1:0] bits are set to NAK or STALL
- When the FIFO buffer is full at token transmit time in the receiving (IN) direction
- When there is no data to be transmitted in the FIFO buffer at token transmit time in the transmitting (OUT) direction

30.3.12 Isochronous Transfers (Pipes 1 and 2)

The USBHS does not support high-bandwidth isochronous transfers but provides the following functions for isochronous transfers:

- Notification of isochronous transfer error

- Interval counter (specified in the PIPEPERI.IITV[2:0] bits)
- Isochronous IN transfer data setup control (IDLY function)
- Isochronous IN transfer buffer flush function (specified in the PIPEPERI.IFIS bit)
- SOF pulse output function

30.3.12.1 Error detection in isochronous transfers

The USBHS provides a function for detecting the errors described in this section, so that when errors occur in isochronous transfers, they can be controlled by software. Table 30.30 and Table 30.31 show the priority order for errors detected by the USBHS and the associated interrupts.

PID errors

- The PID value of the received packet is invalid

CRC errors and bit stuffing errors

- A CRC error is found in a received packet or the bit stuffing is illegal

Maximum packet size exceeded

- The data size of the received packet exceeds the specified maximum packet size

Overrun and underrun errors

In host controller mode:

- The FIFO buffer is full at token transmit time in the IN (receiving) direction
- There is no data to be sent in the FIFO buffer at token transmit time in the OUT (transmitting) direction

In device controller mode:

- There is no data to be sent in the FIFO buffer at token receive time in the IN (transmitting) direction
- The FIFO buffer is full at token receive time in the OUT (receiving) direction

Interval error

In device controller mode, the following cases are treated as an interval error:

- Failure to receive an IN token in the interval frame during an isochronous IN transfer
- Failure to receive an OUT token in the interval frame during an isochronous OUT transfer

Table 30.30 Error detection for token transmission and reception

Detection priority	Error type	Interrupt generated at error detection and status
1	PID error	No interrupts are generated in either host or device controller mode. (Ignored as a corrupted packet.)
2	CRC or bit stuffing error	No interrupts are generated in either host or device controller mode. (Ignored as a corrupted packet.)
3	Overrun or underrun error	An NRDY interrupt is generated to set the OVRN flag to 1 in both host and device controller modes. In device controller mode, a zero-length packet is transmitted in response to an IN token. No data packets are received in response to the OUT token.
4	Interval error	An NRDY interrupt is generated in device controller mode. No interrupt is generated in host controller mode.

Table 30.31 Error detection for data packet reception (1 of 2)

Detection priority	Error type	Interrupt generated at error detection and status
1	PID error	No interrupt is generated. (Ignored as a corrupted packet.)
2	CRC or bit stuffing error	An NRDY interrupt is generated and the FRMNUM.CRCE bit sets to 1 in both host and device controller modes.

Table 30.31 Error detection for data packet reception (2 of 2)

Detection priority	Error type	Interrupt generated at error detection and status
3	Maximum packet size exceeded error	A BEMP interrupt is generated and the PID[1:0] bits set to STALL in both host and device controller modes.

30.3.12.2 DATA PID

The USBHS does not support high-bandwidth transfers. In device controller mode, the USBHS responds as follows to a received PID:

IN direction

- DATA0: Transmitted as data packet PID
- DATA1: Not transmitted
- DATA2: Not transmitted
- mDATA: Not transmitted

OUT direction (full-speed operation)

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Packets ignored
- mDATA: Packets ignored

OUT direction (high-speed operation)

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Received normally as data packet PID
- mDATA: Received normally as data packet PID

30.3.12.3 Interval counter

The isochronous transfer interval can be set in the PIPEPERI.IITV[2:0] bits. In device controller mode, the interval counter enables functions as shown in Table 30.32. In host controller mode, the USBHS generates the token issuance timing, and the interval counter operation is the same as that for interrupt transfers.

Table 30.32 Interval counter functions in device controller mode

Transfer direction	Function	Conditions for detection
IN	Transmit buffer flush	Failure to receive an IN token successfully in the interval frame during an isochronous IN transfer
OUT	Notification of no reception of token	Failure to receive an OUT token successfully in the interval frame during an isochronous OUT transfer

The interval count is performed when an SOF is received or for complemented SOFs, so the isochronism can be maintained even if an SOF is corrupt. The frame interval can be set to 2IITV (μ) frames.

(1) Counter initialization in device controller mode

The USBHS initializes the interval counter under the following conditions:

- Power-on reset:
  This initializes the PIPEPERI.IITV[2:0] bits
- FIFO buffer initialization using the ACLRM bit:
  This does not initialize the IITV[2:0] bits, but does initialize the count value.

After the interval counter is initialized, the interval count starts under either of the following conditions when a packet is transferred successfully:

- An SOF is received after data is transmitted in response to an IN token, with the PID[1:0] bits set to 01b (BUF response)
- An SOF is received after data is received in response to an OUT token, with PID[1:0] bits set to 01b (BUF response)

The interval counter is not initialized under the following conditions:

- When the PID[1:0] bits are set to NAK or STALL
  This does not stop the interval timer. The USBHS attempts the transaction in the next interval.
- USB bus reset and USB suspension
  This does not initialize the IITV[2:0] bits. When an SOF is received, the interval counter starts counting from the value set before SOF was received.

(2) Interval counting and transfer control in host controller mode

The USBHS controls the interval between token issuance operations based on the PIPEPERI.IITV[2:0] bit settings. Specifically, the USBHS issues a token for a selected pipe once every 2IITV frames.

The USBHS starts counting the token issuance interval at the frame following the frame in which the PID[1:0] bits are set to 01b (BUF response) by software.

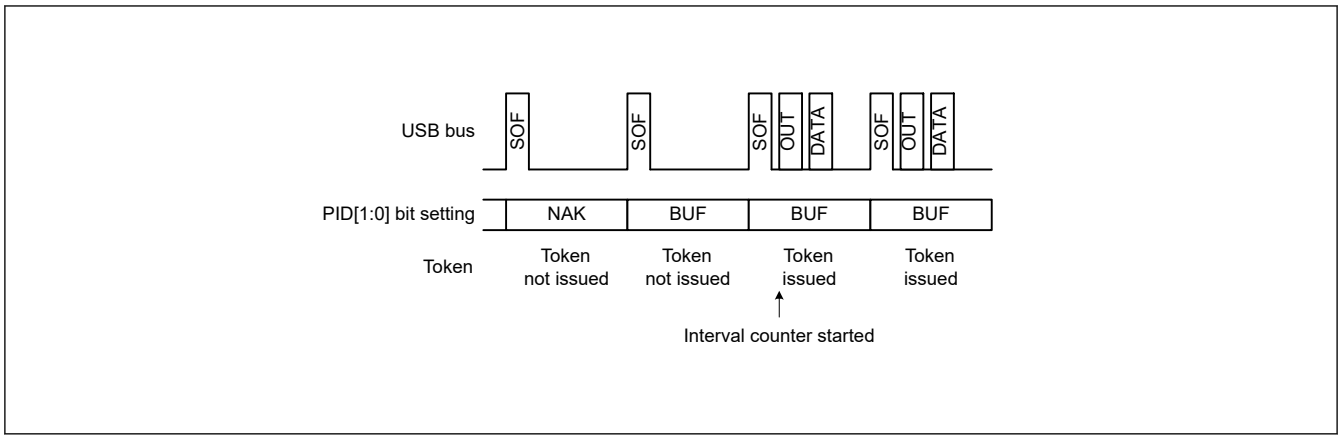


Figure 30.12 Token issuance when IITV[2:0] = 0

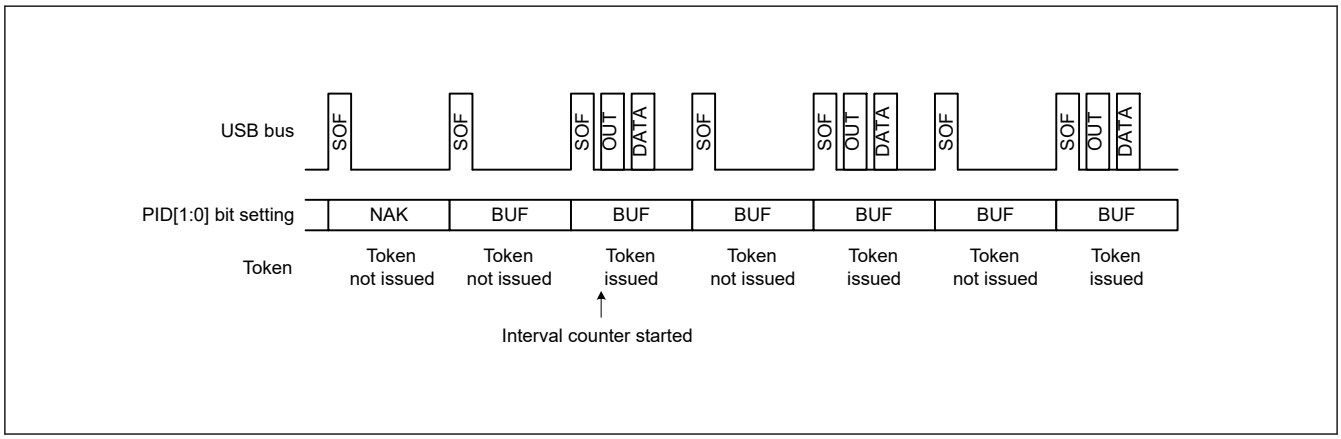


Figure 30.13 Token issuance when IITV[2:0] = 1

When the selected pipe is set for isochronous transfers, the USBHS carries out the following operation in addition to controlling the token issuance interval. The USBHS issues a token even when the NRDY interrupt generation condition is satisfied.

When the selected pipe is for isochronous IN transfers

The USBHS generates an NRDY interrupt when the USBHS issues an IN token but does not receive a packet successfully from a peripheral device (no response or packet error).

When the selected pipe is for isochronous OUT transfers

The USBHS sets the OVRN flag to 1, generating an NRDY interrupt and transmitting a zero-length packet, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer, because the CPU or DMAC/DTC is too slow in writing data to the FIFO buffer.

The token issuance interval is reset on any of the following conditions:

- When the MCU is reset
  This initializes the IITV[2:0] bits
- When the PIPEnCTR.ACLRM bit is set to 1 by software

(3) Interval counting and transfer control in device controller mode

When the selected pipe is for isochronous OUT transfers

The USBHS generates an NRDY interrupt when it fails to receive a data packet within the interval set in the PIPEPERI.IITV[2:0] bits.

The USBHS also generates an NRDY interrupt when it fails to receive data because of a CRC error or other errors contained in the data packet or because of the FIFO buffer is full.

The NRDY interrupt is generated on SOF packet reception. Even if the SOF packet is corrupted, internal complementation allows the interrupt to be generated when the SOF packet is received. However, when the IITV[2:0] bits are set to a value other than 0, the USBHS generates an NRDY interrupt on receiving an SOF packet for every interval after interval counting starts.

When the PID[1:0] bits are set to 00b (NAK response) by software after starting the interval timer, the USBHS does not generate an NRDY interrupt on receiving an SOF packet.

The timing for starting interval counting depend on the IITV[2:0] setting as follows:

- When the IITV[2:0] bits = 0:
  The interval counting starts when the PID[1:0] bits of the selected pipe are changed to BUF
- When the IITV[2:0] bits ≠ 0:
  The interval counting starts on completion of successful reception of the first data packet after the PID[1:0] bits for the selected pipe are changed to 01b (BUF response)

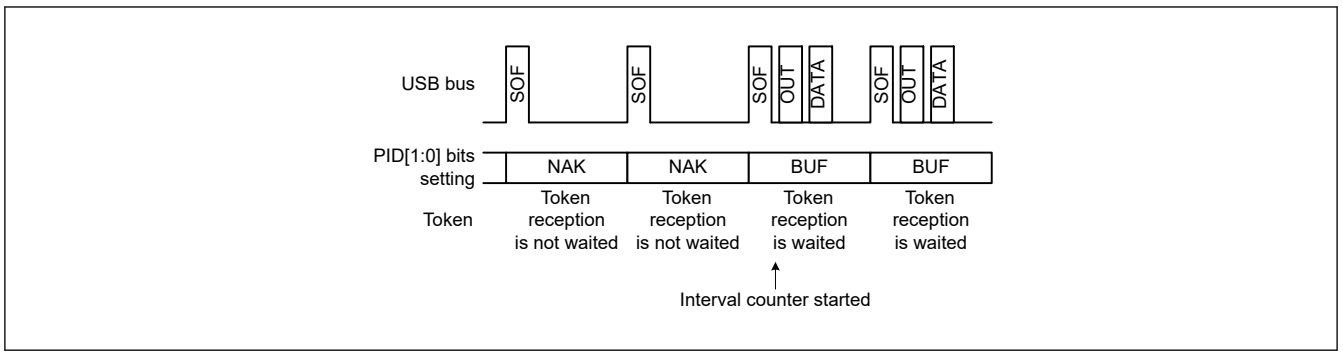


Figure 30.14 Relationship between frames and expected token reception when IITV[2:0] = 0

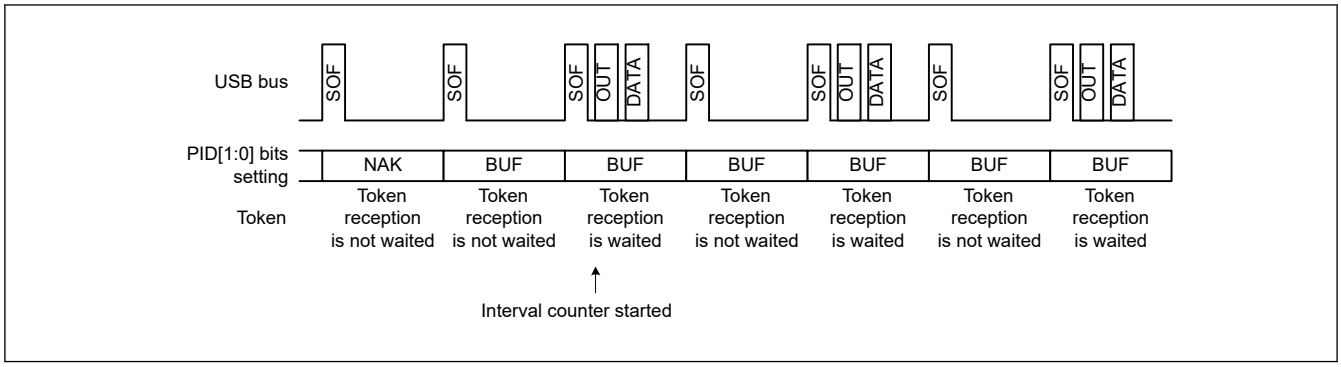


Figure 30.15 Relationship between frames and expected token reception when IITV[2:0] ≠ 0

When the selected pipe is for isochronous IN transfers

The PIPEPERI.IFIS bit must be 1 for this use case. When the IFIS bit is cleared to 0, the USBHS transmits a data packet in response to a received IN token, regardless of the PIPEPERI.IITV[2:0] setting.

When IFIS is 1 and there is data to be transmitted in the FIFO buffer, the USBHS clears the FIFO buffer when it fails to receive an IN token in the frame at the interval set in the IITV[2:0] bits.

The USBHS also clears the FIFO buffer when it fails to receive an IN token successfully because of a bus error, such as a CRC error, contained in the IN token.

The FIFO buffer is cleared on SOF packet reception. Even if the SOF packet is corrupted, the internal complementation allows the FIFO buffer to be cleared when the SOF packet is received.

The timing to start interval counting depends on the IITV[2:0] setting, as with OUT transfers.

The interval is counted on any of the following conditions in device controller mode:

- When a hardware reset is applied to the USBHS (which also sets the IITV[2:0] bits to 000b)
- When the PIPEnCTR.ACLRM bit is set to 1 by software
- When the USBHS detects a USB bus reset

(4) Transmit data setup for isochronous transfers in device controller mode

With isochronous data transmission using the USBHS in device controller mode, after data is written to the FIFO buffer, a data packet can be transmitted in the first frame after the SOF packet is detected. This isochronous transfer transmit data setup function can identify the frame that started transmission.

When the double buffering is used, transmission is only enabled for the buffer in which data writing was completed first, even after the data write to both buffers is complete. Accordingly, even if multiple IN tokens are received, only the one packet of FIFO buffer data is transmitted.

When the FIFO buffer is ready to transmit data when an IN token is received, the data is transferred and a normal response is returned. However, if the FIFO buffer cannot transmit data, a zero-length packet is transmitted and an underrun error occurs.

Figure 30.16 shows an example of transmission using the isochronous transfer transmit data setup function when the IITV[2:0] bits are set to 0 (every frame).

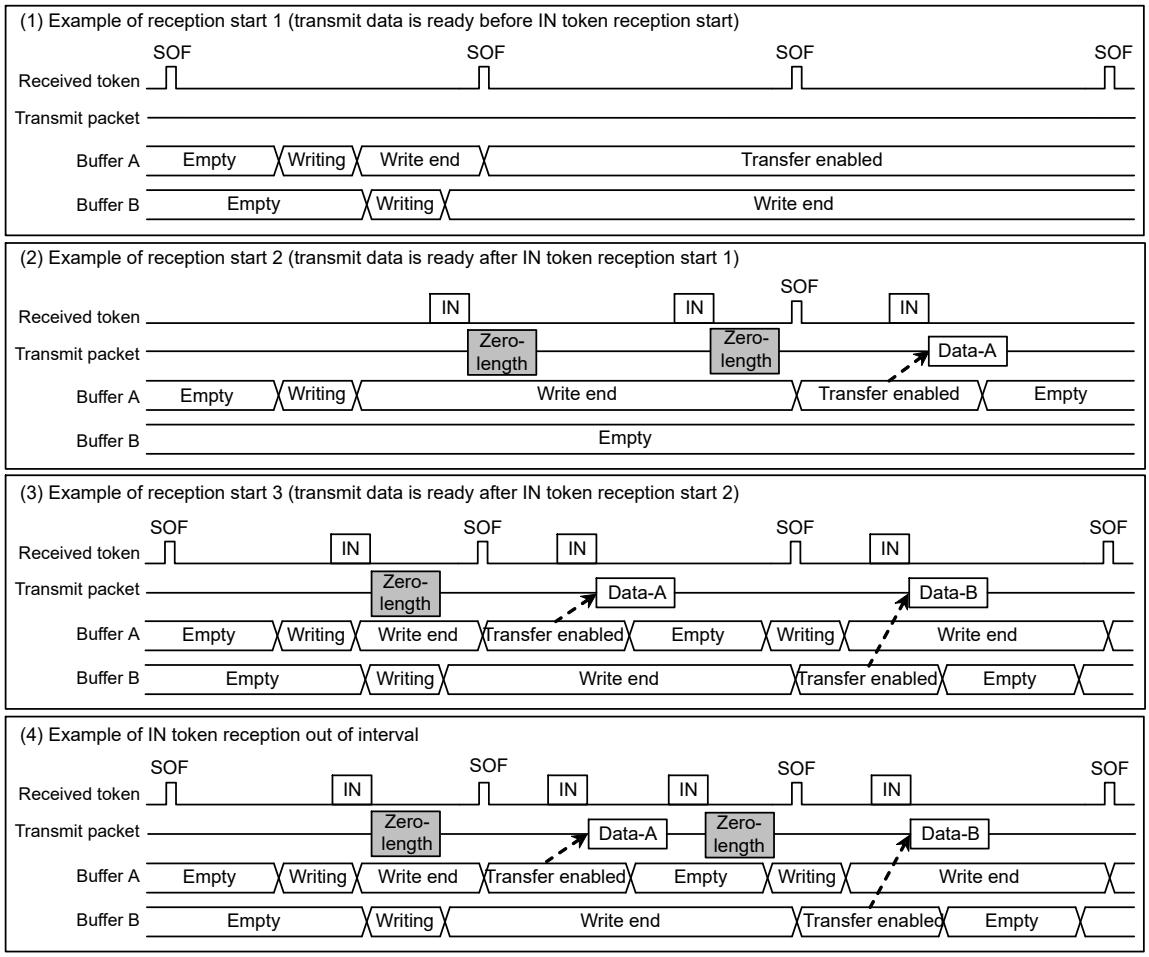


Figure 30.16 Example data setup operation

(5) Isochronous transfer transmit buffer flush in device controller mode

In device controller mode during isochronous data transmission, if the USBHS receives an SOF packet for the next frame without receiving an IN token in the interval frame, it operates as if the IN token is corrupt and clears the buffer that is enabled for transmission, putting that buffer in the writing enabled state.

When double buffering is used and writing to both buffers is complete, the cleared FIFO buffer is assumed to be the one where the data was transmitted in the interval frame, and transmission is enabled for the FIFO buffer that was not cleared on SOF packet reception.

The timing of the buffer flush function depends on the PIPEPERI.IITV[2:0] setting as follows:

- When IITV[2:0] = 0:
  The buffer flush operation starts from the first frame after the pipe is enabled
- When IITV[2:0] ≠ 0:
  The buffer flush operation starts after the first normal transaction

Figure 30.17 shows an example buffer flush. When an unanticipated token is received before the interval frame, the USBHS sends the write data or a zero-length packet as an underrun error, depending on the data setup status.

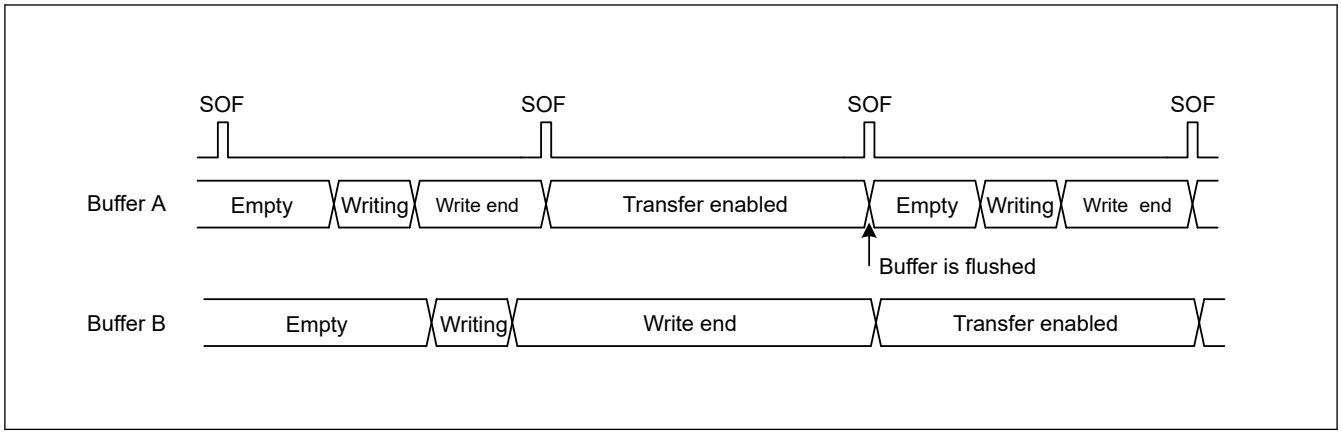


Figure 30.17 Example buffer flush operation

Figure 30.18 shows an example interval error occurrence. There are five types of interval errors, as shown in the figure. An interval error occurs at timing ①, and the buffer flush function is activated.

If an interval error occurs during an IN transfer, the buffer flush function is activated. If it occurs during an OUT transfer, an NRDY interrupt is generated. Use the FRMNUM.OVRN bit to distinguish between this and NRDY interrupts triggered by received packet errors and overrun errors.

For tokens that are shaded in the figure, responses are returned based on the FIFO buffer status.

- IN direction:
  - If the buffer is ready to transfer data, the data is transferred and a normal response is returned
  - If the buffer is not ready to transfer data, a zero-length packet is transmitted and an underrun error occurs
- OUT direction:
  - If the buffer is ready to receive data, the data is received and a normal response is returned
  - If the buffer is not ready to receive data, the received data is discarded and an overrun error occurs

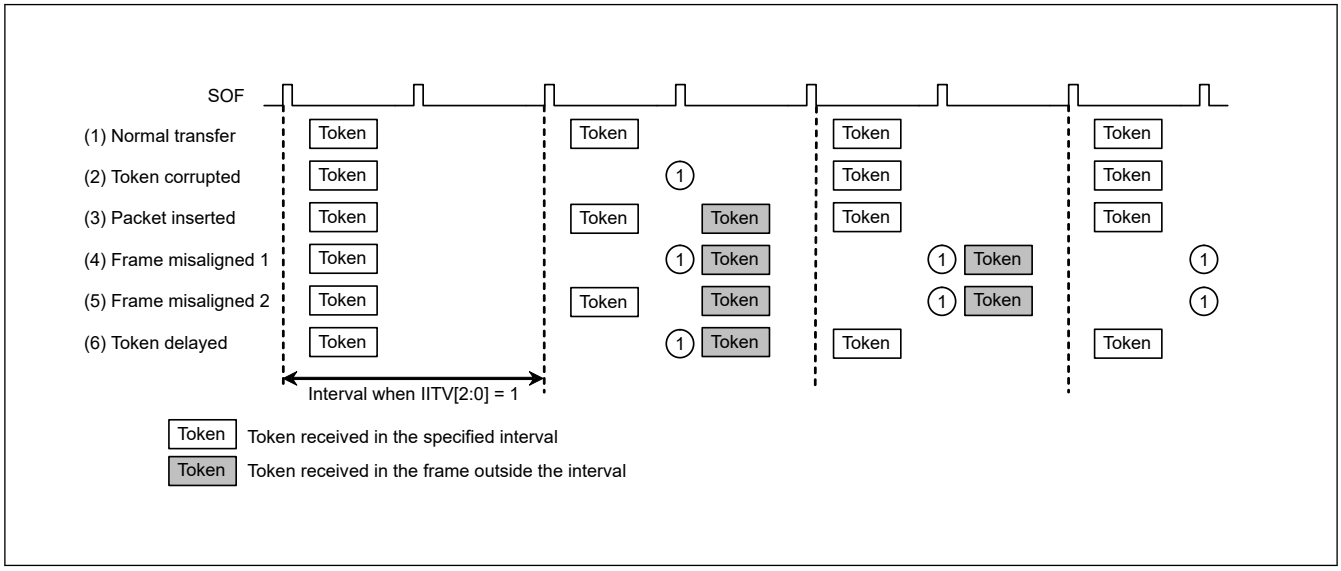


Figure 30.18 Example interval error occurrence when PIPEPERI.IITV[2:0] = 1

30.3.13 SOF Complementation Function

In device controller mode, if packet reception is disabled at intervals of 125 µs in high-speed mode or 1 ms in full-speed mode because the SOF packet is missing or corrupted, the USBHS complements the SOF. SOF complementation begins when the SYSCFG.USBE and LPSTS.SUSPENDM bits are set to 1 and an SOF packet is received. The complementation function is initialized under the following conditions:

- Power-on reset

- USB bus reset
- Suspend state detection

The SOF complementation function operates as follows:

- The frame interval (125 µs or 1 ms) is determined by the reset handshake protocol result
- The complementation function is not activated until an SOF packet is received
- When the first SOF packet is received, complementation is performed by counting 125 µs or 1 ms on the 48-MHz internal clock
- When the second or subsequent SOF packets are received, complementation is performed at the previous reception interval
- Complementation is not performed in the Suspend state or on reception of a USB bus reset. During high-speed operation, complementation continues for 3 ms from the last packet on transition to the Suspend state

The USBHS supports the following functions controlled by SOF packet reception. These functions operate normally with SOF complementation if the SOF packet is missing:

- Updating of the frame number and micro frame number
- SOFR interrupt and micro-SOF lock
- SOF pulse output
- Isochronous transfer interval count

If an SOF packet is missing during full-speed operation, the FFRMNUM.FRNM[10:0] flags are not updated. If a micro-SOF packet is missing during high-speed operation, the URMNUM.UFRNM[2:0] bits are updated.

However, if a micro-SOF packet is missing while the UFRNM[2:0] bits are set to 000b, the FRNM bits are not updated. In this case, even if a subsequent micro-SOF packet with a value other than UFRNM[2:0] bits = 000b is received successfully while UFRNM[2:0] bits are set to the value other than 000b, the FRNM bits are not updated.

30.3.14 Pipe Schedule

30.3.14.1 Conditions for generating transactions

In host controller mode and when the DVSTCTR0.UACT bit is set to 1, the USBHS generates transactions under the conditions as shown in Table 30.33.

Table 30.33 Conditions for generating transactions

Transaction	Conditions for generation				
	DIR	**PID[1:0]**	**IITV0**	**Buffer state**	**SUREQ**
Setup	—*1	—*1	—*1	—*1	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	—*1	Receive area exists	—*1
	OUT	BUF	—*1	Transmit data exists	—*1
Interrupt transfer	IN	BUF	Valid*2	Receive area exists	—*1
	OUT	BUF	Valid*2	Transmit data exists	—*1
Isochronous transfer	IN	BUF	Valid*2	*3	—*1
	OUT	BUF	Valid*2	*4	—*1

Note 1. An em dash (—) in the table indicates that the condition is unrelated to the generating of tokens.
Note 2. “Valid” indicates that, for interrupt transfers and isochronous transfers, a transaction is generated only in transfer frames that are based on the interval counter.
Note 3. This indicates that a transaction is generated regardless of whether there is a receive area. If there is no receive area, however, the received data is discarded.
Note 4. This indicates that a transaction is generated regardless of whether there is any data to be transmitted. If there is no data to be transmitted, however, a zero-length packet is transmitted.

30.3.14.2 Transfer schedule

This section describes the transfer scheduling within a frame of the USBHS. After the USBHS sends an SOF, the transfer is carried out in the following sequence:

1. Execution of periodic transfers:
   A pipe is searched for in the order of pipe 1 → pipe 2 → pipe 6 → pipe 7 → pipe 8 → pipe 9, and then if there is a pipe for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.
2. Setup transactions for control transfers:
   The DCP is checked, and if a setup transaction is possible, it is sent.
3. Execution of bulk transfers, control transfer data stages, and control transfer status stages:
   A pipe is searched for in the order of DCP → pipe 1 → pipe 2 → pipe 3 → pipe 4 → pipe 5, and then if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated.
   When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

30.3.14.3 Enabling USB communication

Setting the DVSTCTR0.UACT bit to 1 initiates an SOF transmission, and transaction generation is enabled. Setting the UACT bit to 0 stops SOF transmission, and the Suspend state is invoked. If the UACT setting is changed from 1 to 0, processing stops after the next SOF is sent.

30.3.15 Battery charging detection processing

The USBHS provides control over the data contact detection processing (D+ line contact checking), primary detection processing (charger detection processing), and secondary detection processing (charger determination processing) as defined in the Battery Charging Specification.

This section describes operations required in device and host controller modes.

30.3.15.1 Processing in device controller mode

To operate a function device as a battery charging portable device:

1. Start primary detection processing after detecting contact with the D+ and D- lines. The Battery Charging Specification describes two processing methods for Data Contact Detection. The USBHS supports both methods as follows:
   - Software processing
     After a VBINT interrupt or polling of the VBSTS flag indicates a change in the state of the USBHS_VBUS input pin, software controls a wait from 300 to 900 ms. The BCCTRL.VDPSRCE and IDMSINKE bits are then both set to 1, enabling the VDP_SRC and IMP_SINK circuits, respectively, to start primary detection processing.
   - Hardware processing
     Apply 7 to 13 µA of current to the D+ line to hold the D+ line at the logical high level. This is done to detect the D+ and D- lines going to the logical low level because of pull-down resistors on the host device side when the D+ and D- lines come in contact with those of the host. Monitor the SYSSTS0.LNST[1:0] flags while the BCCTRL.IDPSRCE bit is set to 1, enabling the IDP_SRC circuit, to see when the level on the D+ line changes from high to low. After detecting a low level on the D+ line, clear the BCCTRL.IDPSRCE bit to 0, disabling the IDP_SRC circuit, and set both the BCCTRL.VDPSRCE and IDMSINKE bits to 1, enabling the VDP_SRC and IDM_SINK circuits, respectively, to start primary detection processing. The VDPSRCE and IDMSINKE bits must be set to 1 simultaneously.
2. After the start of primary detection processing followed by a software-controlled wait of 40 ms, check the BCCTRL.CHGDETSTS flag. A value of 1 indicates detection of a charger, and secondary detection processing starts.[*1]
3. To start secondary detection processing, clear both the BCCTRL.VDPSRCE and IDMSINKE bits to 0, disabling the VDP_SRC and IDM_SINK circuits, respectively. Next, set both the BCCTRL.VDMSRCE and IDPSINKE bits to 1, enabling the VDM_SRC and IDP_SINK circuits, respectively.
4. After the start of secondary detection processing followed by a software-controlled wait of 40 ms, check the BCCTRL.PDDETSTS flag. A value of 1 indicates that secondary detection processing is complete.

Note 1. In primary detection processing, detection of a voltage above the range from 0.25 to 0.4 V and below the range from 0.8 to 2.0 V on the D-Line indicates that the other device is a host device that supports battery charging (charging downstream port). The BCCTRL.CHGDETSTS flag in the PHY block only indicates whether the voltage on the D-line is higher than the range from 0.25 to 0.4 V, so add processing as required to read the SYSSTS0.LNST[1:0] flags and confirm that the voltage on the D- line is also below the range from 0.8 to 2.0 V.

Figure 30.19 illustrates this processing flow.

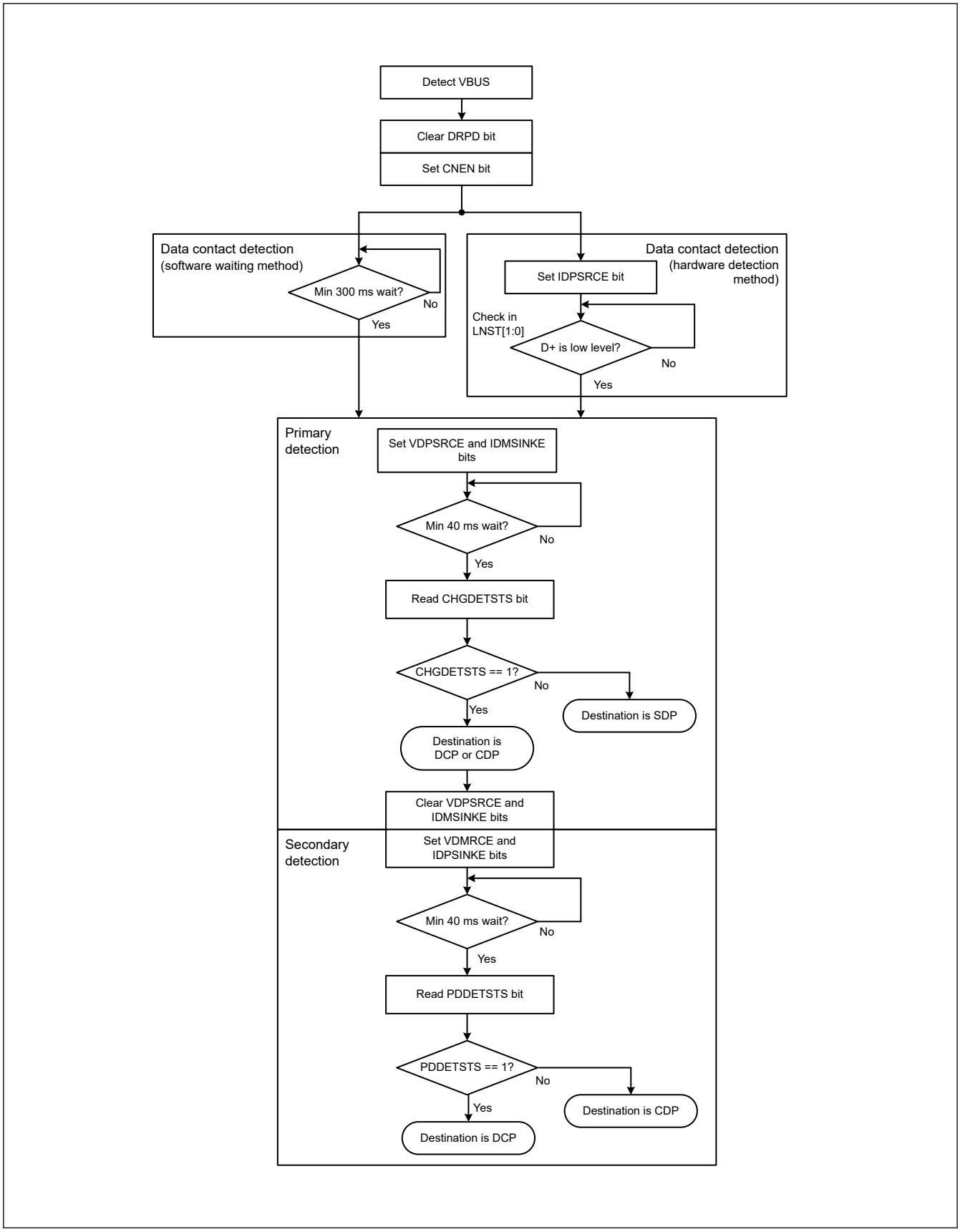


Figure 30.19 Processing flow as portable device

30.3.15.2 Processing in host controller mode

In host controller mode, driving the D- line is required for a portable device to perform primary detection. The USBHS supports the following two primary detection methods:

- When the hardware has a portable device detection function
- When the hardware does not have the function or the function is present but not used

Figure 30.20 and Figure 30.21 show the processing flows for these methods.

(1) When the hardware has a portable device detection function

a. Start driving the USBHS_VBUS input pin.

b. Set the BCCTRL.IDMSINKE bit to 1 to enable the portable device detection circuit.

c. Monitor the portable device detection signal and start driving the D-line when the level of the portable device detection signal is high*1.

d. Stop driving the D-line when the portable device detection signal is at the low level*1.

Note 1. The PDDETINT interrupt indicates a change in the level of the portable device detection signal (EUH_CPDDET), and the current level can be obtained by reading the PDDETSTS flag.

(2) When the hardware does not have a portable device detection function or the function is not used

Software handles the timing of steps a. and b.

a. After a disconnect is detected, start driving the D-line within 200 ms.

b. After a connect is detected, stop driving the D-line within 10 ms.

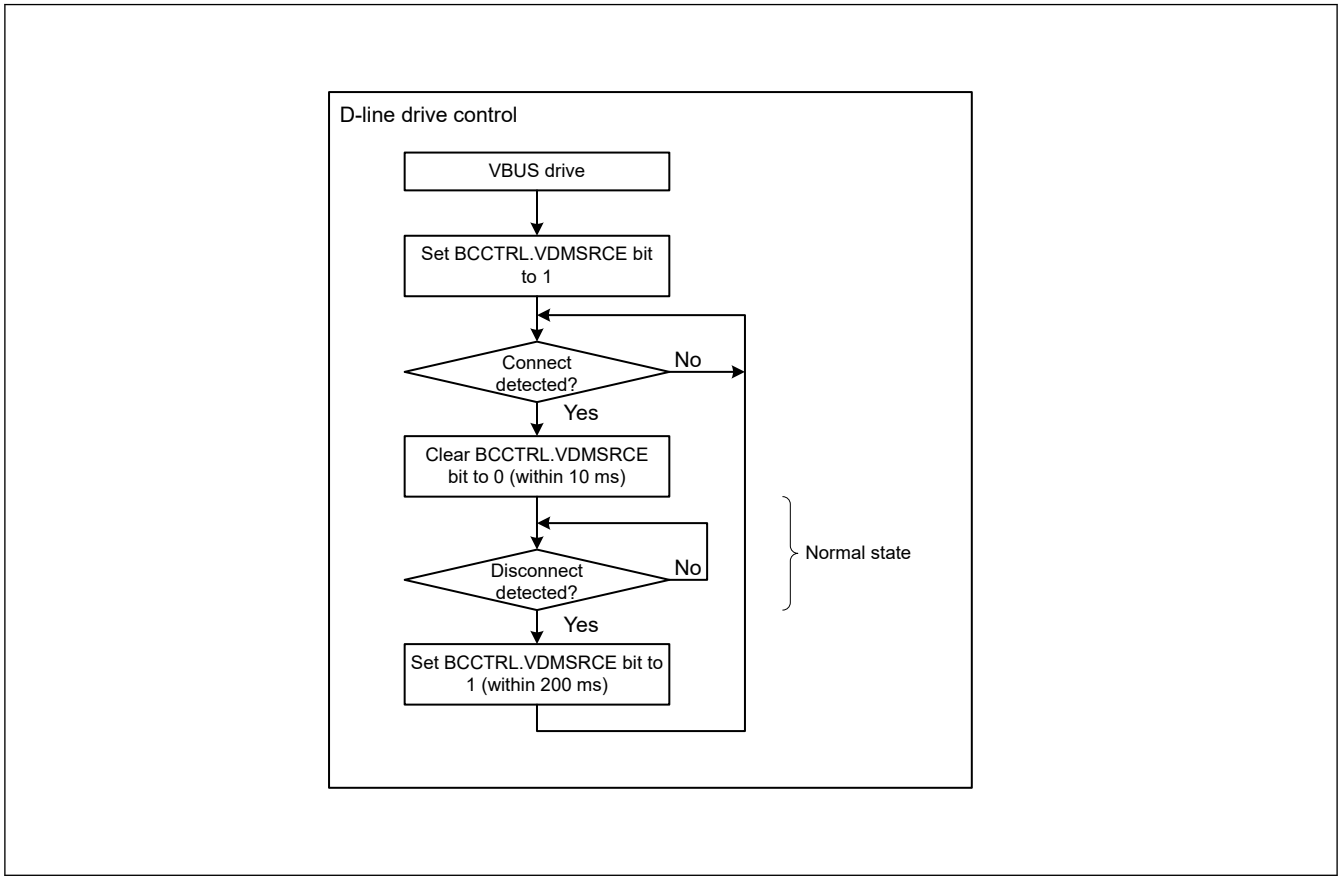


Figure 30.20 Processing flow as charging downstream port without hardware portable device detectionfunction or when function is not used

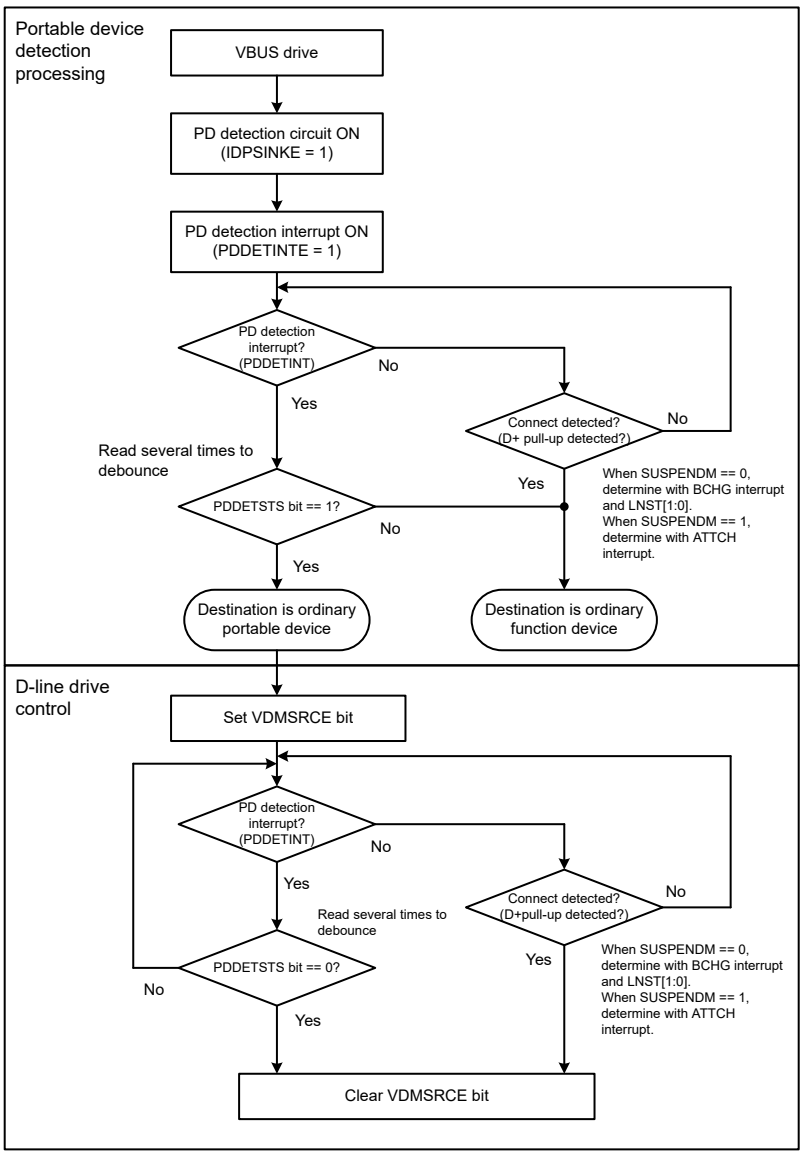


Figure 30.21 Processing flow as charging downstream port with hardware portable device detection function

30.3.16 Link Power Management Processing

The Link Power Management standard defines the existing Suspend state as the L2 state and also defines the L1 state as a state that allows transition and return with lower latency than the L2 state (Suspend). Table 30.34 provides a comparison between the L2 (Suspend) and L1 states.

Table 30.34 Comparison between L2 (Suspend) state and L1 state (1 of 2)

Parameter	L1 state	L2 (Suspend) state
Transition	LPM transaction	Idle for 3 ms

Table 30.34 Comparison between L2 (Suspend) state and L1 state (2 of 2)

Parameter	L1 state	L2 (Suspend) state
Return caused by host	Host: Minimum drive period (75 µs to 1.175 ms) can be specified by the host. Function: 10-µs K drive	Host: Minimum 20-ms K drive Function: 10-ms K drive
Return caused by function	Device: 50-µs K drive Function: 60- to 990-µs K drive Device: 10-µs K drive	Function: 1- to 15-ms K drive Host: Minimum 20-ms K drive Function: 10-ms K drive
Signaling	Full-speed and low-speed idle	Full-speed and low-speed idle

30.3.16.1 Processing in device controller mode

(1) Descriptor contents

In device controller mode, the USBHS must return its descriptor on receiving the GetDescriptor command.

Change the content of the descriptor to be returned depending on whether the transition to and return from the L1 state corresponds to the processing for the LPM transaction. The following table shows the relationship between LPM correspondence and the descriptor.

Table 30.35 Relationship between LPM correspondence and descriptor

Correspondence with LPM	bcdUSB field	USB2.0 extension descriptor		Response to received LPM request	Notes
		Provided/ not provided	Value of LPM bit		
Does not correspond	0x0200	Not provided	—	No response	Normal operation when the LPM is not supported
	0x0201	Provided	0	STALL	Setting for clear non-correspondence to LPM. In this case, a STALL response must be returned.
Corresponds	0x0201	Provided	1	ACK or NYET	Normal operation when the LPM is supported

Declare whether to correspond to the transition to and return from L1 in the LPM bit in the USB 2.0 extension descriptor. To provide the USB2.0 extension descriptor, the bcdUSB field of the device descriptor must be set to a value of 0x0201 or larger.

When the LPM is not supported, the USB2.0 extension descriptor is not provided and the bcdUSB field value must be 0x0200. If an LPM token is received in this case, it must be ignored. It is also possible to set the bcdUSB field value to 0x0201 and the LPM bit in the USB2.0 extension descriptor to 0 (LPM tokens not supported). In this case, the LPM token cannot be ignored and a STALL response must be returned.

When the LPM token is supported, set the bcdUSB field value to 0x0201 and set the LPM bit in the USB 2.0 extension descriptor to 1 (LPM tokens supported). This allows acknowledgment when returning a NYET or ACK response to the LPM token.

(2) Processing during LPM token reception

Transition to and return from the L1 state in device controller mode is as follows:

a. When the USBHS receives an LPM token from the host, the L1RESPEN, L1RESPMD[1:0], and L1NEGOMD settings in PL1CTRL1 determine whether a response packet is sent or the token is ignored and, if a response is to be sent, whether it is an ACK, NYET, or STALL packet.

b. If an ACK response to the LPM token is sent and the host does not transmit another LPM token in 8 µs, the USBHS enters the L1 state. The USBHS handles detection of the newly transmitted packet and the transition to the L1 state. The DVST interrupt can be used to detect the transition.

c. Two types of processing can return the USBHS from the L1 state:

- When the host drives the D-line in the K-state:
  The function device detects the K-state and starts processing the return from the L1 state in response to an RESM interrupt request
- When the function device outputs a remote wakeup signal:
  If the software on the function device sets the DVSTCTR0.WKUP bit to 1, it sends a remote wakeup signal to the host

The software clears the DVSTCTR0.WKUP bit to 0 on returning from the L2 (Suspend) state, and the USBHS clears the DVSTCTR0.WKUP bit to 0 for return from the L1 state.

(3) HIRD field value negotiation function

The HIRD field value included in the LPM token indicates the host K-drive period on return from the L1 state. The HIRD field value can be adjusted according to the requirements of the target system. For example, a small HIRD field value is better for systems focusing on higher transfer efficiency, while a large HIRD field value is better for systems focusing on low power consumption.

Based on the L1NEGOMD and HIRDTHR[3:0] settings in PL1CTRL1, an ACK response is returned when the received HIRD field value is in the expected range, and otherwise a NYET response is returned, requesting the host to change the HIRD field value.

Note: This HIRD field value negotiation function at the host must also support negotiation processing.

30.3.16.2 Processing in host controller mode

(1) Processing during LPM token transmission

Transition to and return from the L1 state in host controller mode is as follows:

a. When the HL1CTRL.L1REQ bit is set to 1, an LPM token is sent to the function device from the host device.

b. If an ACK response is received from the function device, a transition to the L1 state starts within 10 µs and is complete within 50 µs. If a transaction error is detected, another LPM token is transmitted within 8 µs. Retransmission can proceed up to two times. The USBHS handles all of this processing.

c. Two types of processing can return the USBHS from the L1 state:

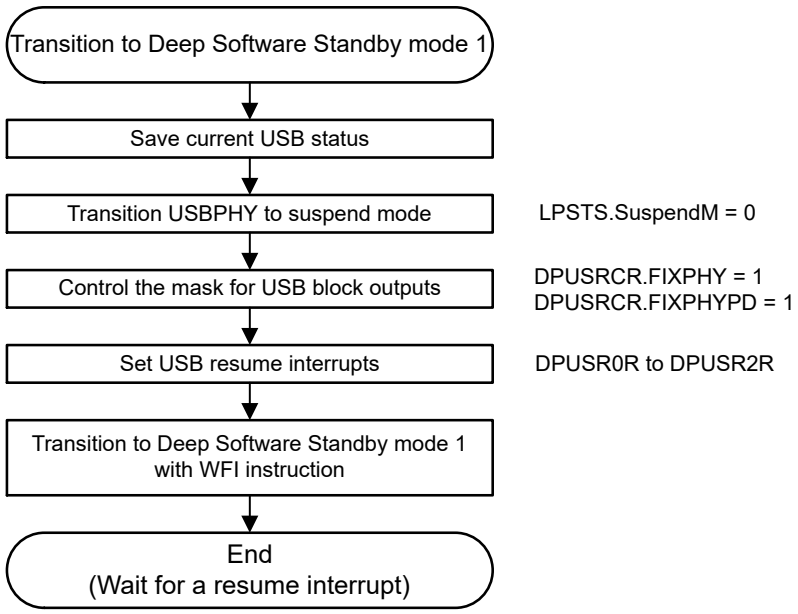
- When the host drives the D-line for the K state:
  When the DVSTCTR0.RESUME bit is set to 1, the host device starts driving the D-line for the K-state and starts processing the return
- When the function device generates a remote wakeup signal:
  When the host device detects a remote wakeup signal from the function device, it sets the DVSTCTR0.RESUME bit to 1 and starts driving the D-line for the K-state

Unlike when returning from the Suspend (L2) state, the USBHS clears the DVSTCTR0.RESUME bit to 0. After clearing the RESUME bit, it sets the DVSTCTR0.UACT bit to 1 and issues an L1RSMEND interrupt request.

30.3.17 Deep Software Standby Mode 1 Because of USB Suspend/Resume Interrupts

Deep Software Standby mode 1 can be canceled by a USB suspend/resume interrupt. USB suspend/resume interrupts are detected by the USB resume detecting unit, which controls and monitors the USB I/O pins to detect the interrupts.

Figure 30.22 shows the flow for setting the USBHS when entering Deep Software Standby mode 1 from either host or device controller mode. Figure 30.23 and Figure 30.24 show the flows for setting the USBHS when canceling Deep Software Standby mode 1 from host controller mode. Figure 30.25 shows the flow for setting the USBHS when canceling Deep Software Standby mode 1 from device controller mode.



Note 1. When connected to an external device, transition to Deep Software Standby mode 1 is enabled only on USB Suspend.
Note 2. Never invoke Deep Software Standby mode 1 in the following cases:
- In an L1 Suspend status with the USB-LPM protocol
- When remote wakeup is enabled in host controller mode.

Figure 30.22 USBHS setup flow for transition to Deep Software Standby mode 1 as a host or device controller

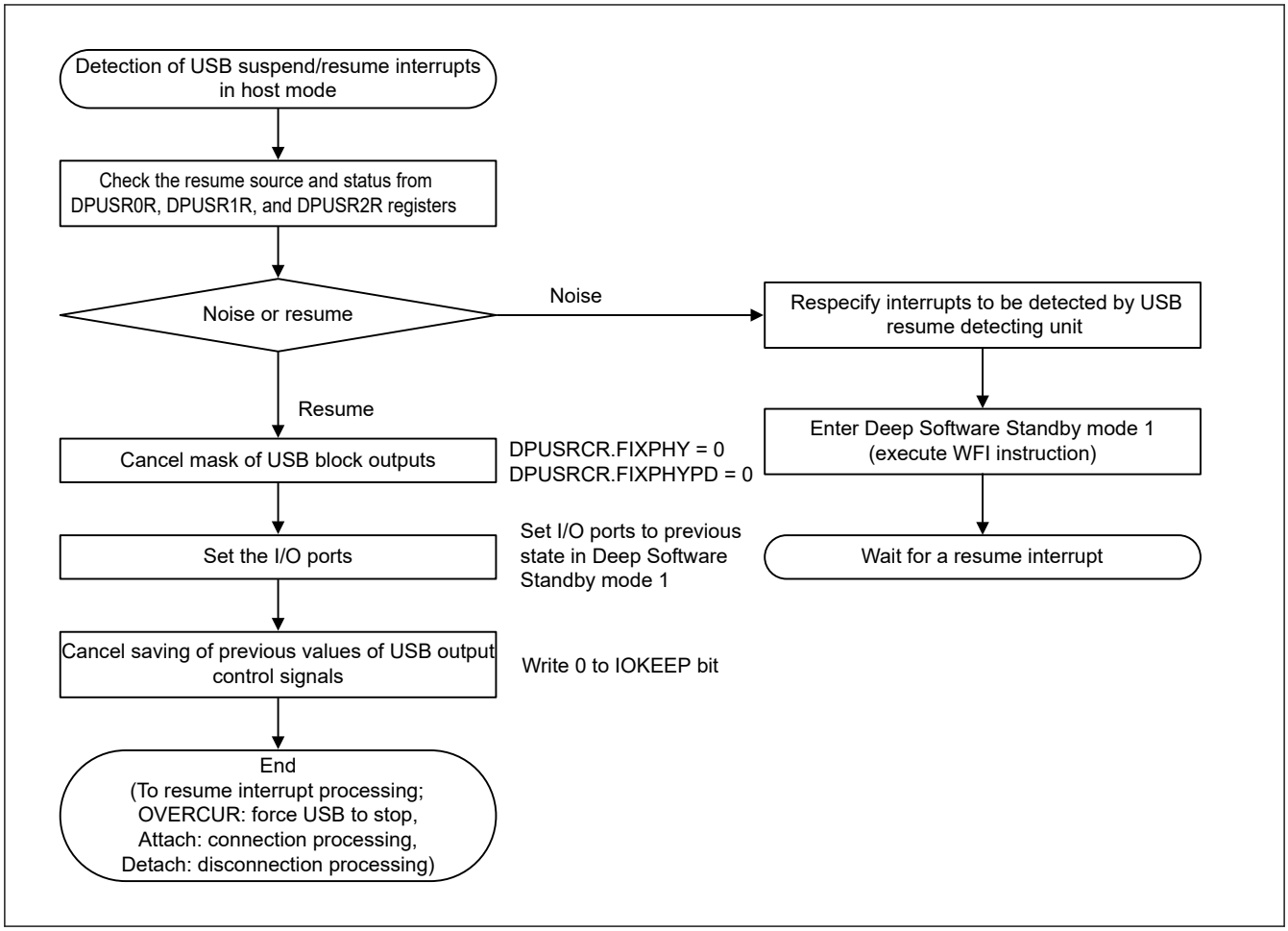


Figure 30.23 USBHS setup flow for canceling Deep Software Standby mode 1 as a host controller (1)

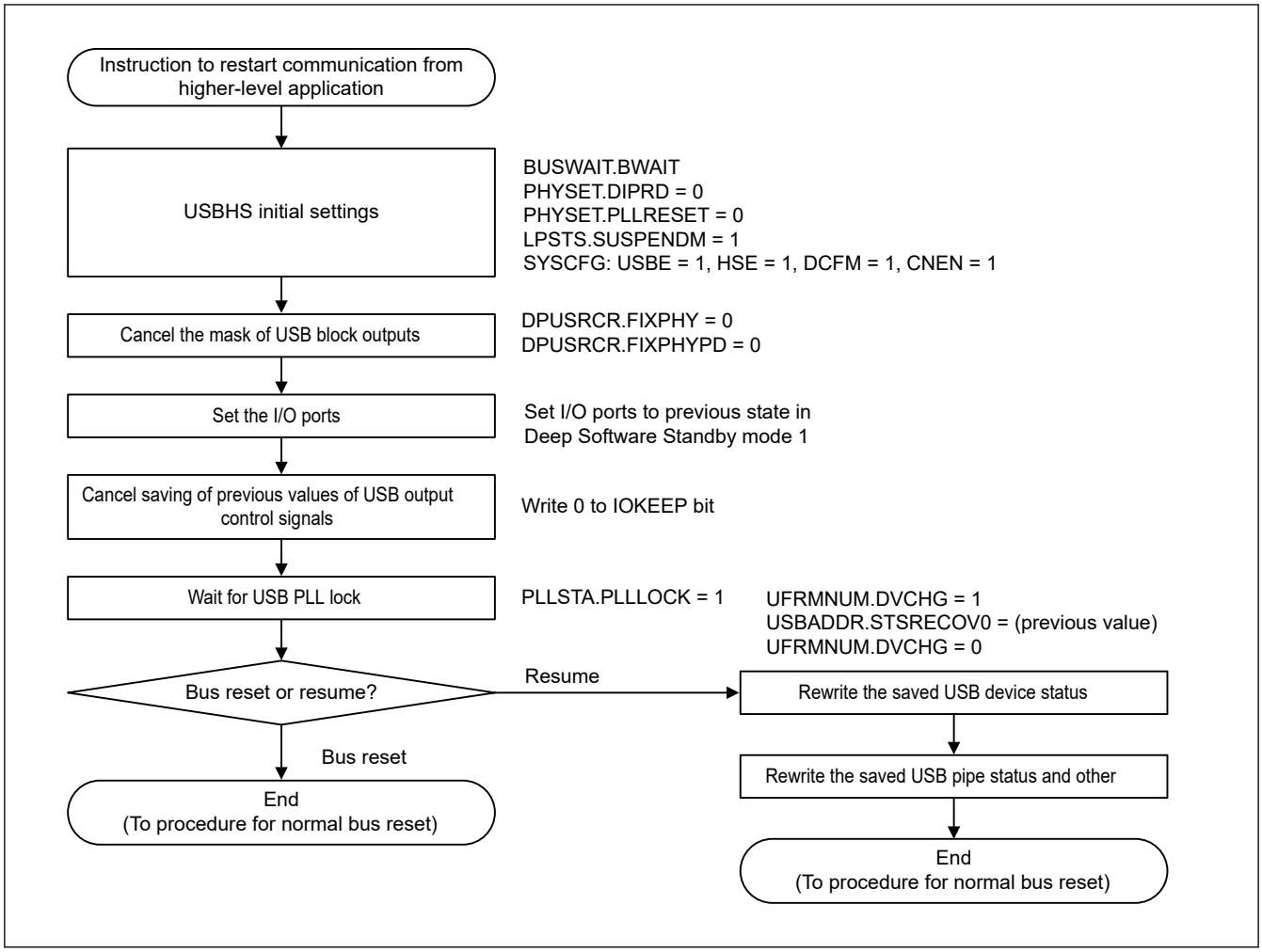
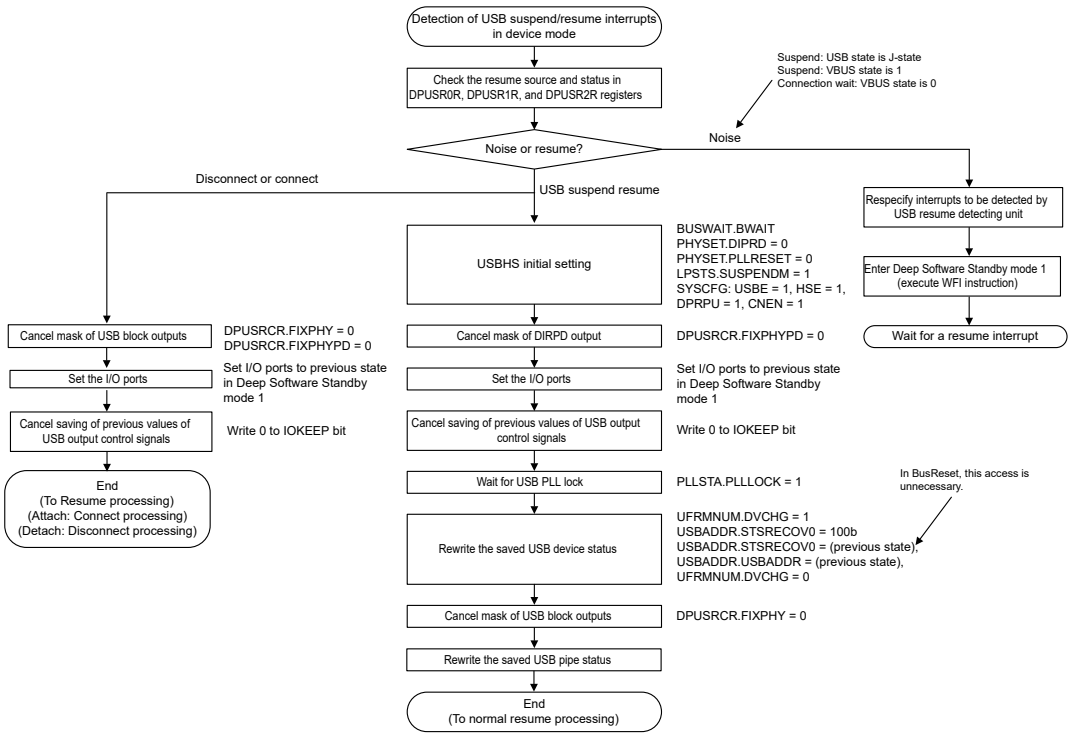


Figure 30.24 USBHS setup flow for canceling Deep Software Standby mode 1 as a host controller (2)



Note 1. In BusReset return, the write to the STSRECOV0 bit must be within 2.5 µS of USBHS-PHY PLL oscillation start

Figure 30.25 USBHS setup flow for transition to Deep Software Standby mode 1 as a host or device controller

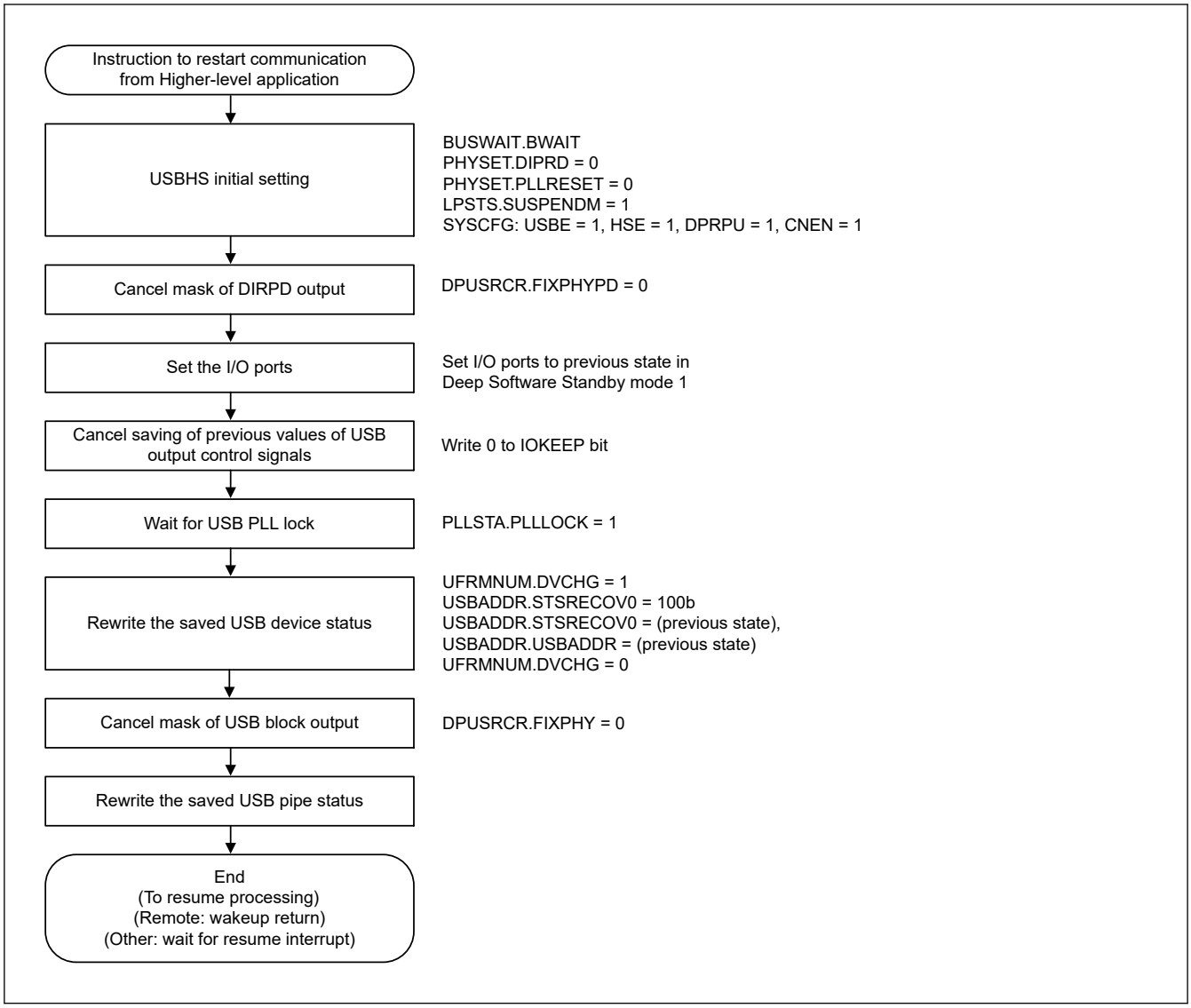


Figure 30.26 USBHS setup flow for canceling Deep Software Standby mode 1 as a device controller (2)

30.3.18 Example External Connection Circuits

Figure 30.27 shows an example OTG connection in a self-powered system. The USBHS controls the pull-up resistor of the D+ line and the pull-down resistor of D+ and D- lines. Select pull-up and pull-down for the lines in the SYSCFG.DPRPU and SYSCFG.DRPD bits. In device controller mode, the pull-up resistor of USB data line is disabled if SYSCFG.DPRPU bit is set to 0 while communicating with the USB host. The USBHS can use this to notify the USB host of a device disconnect.

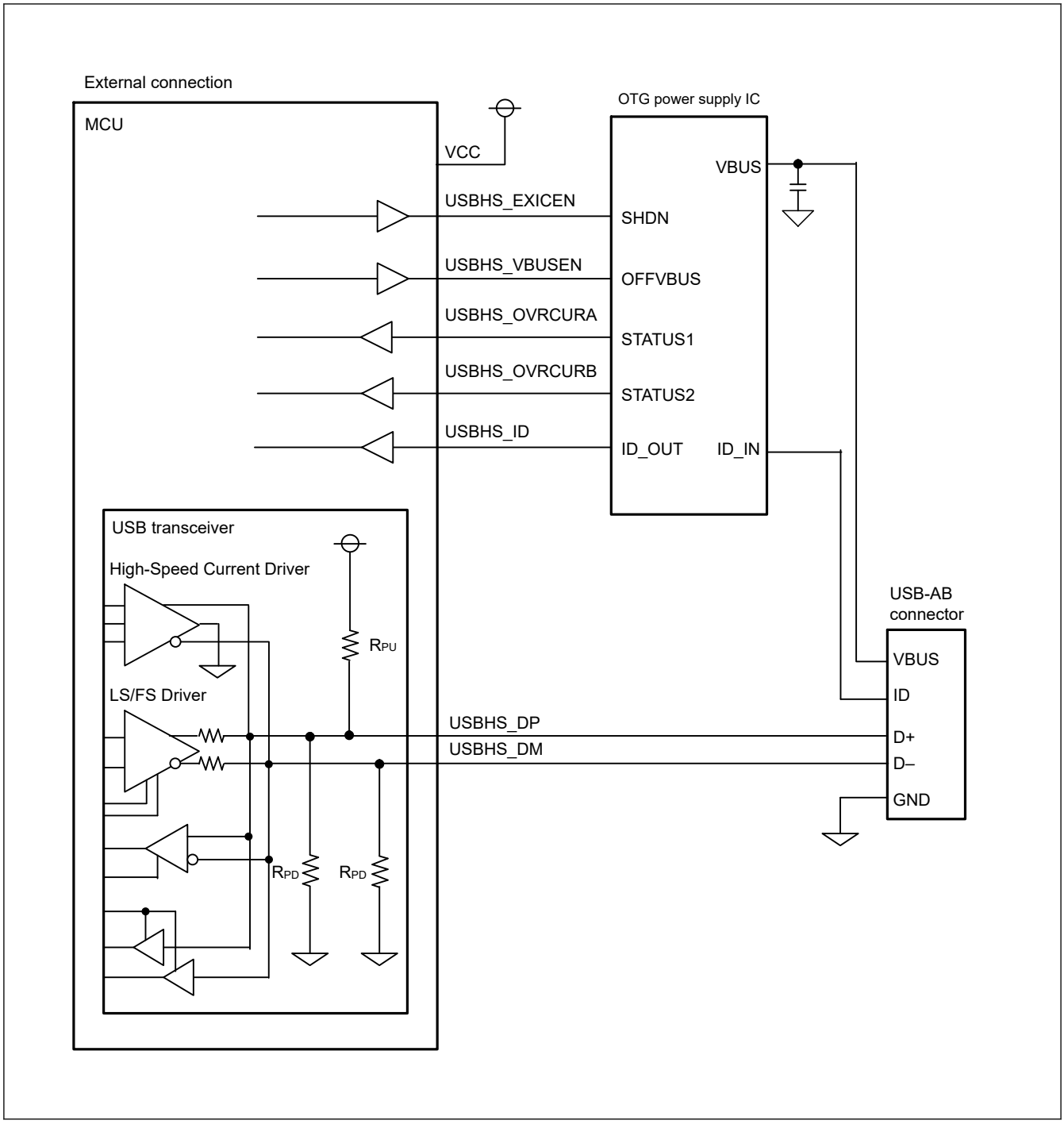
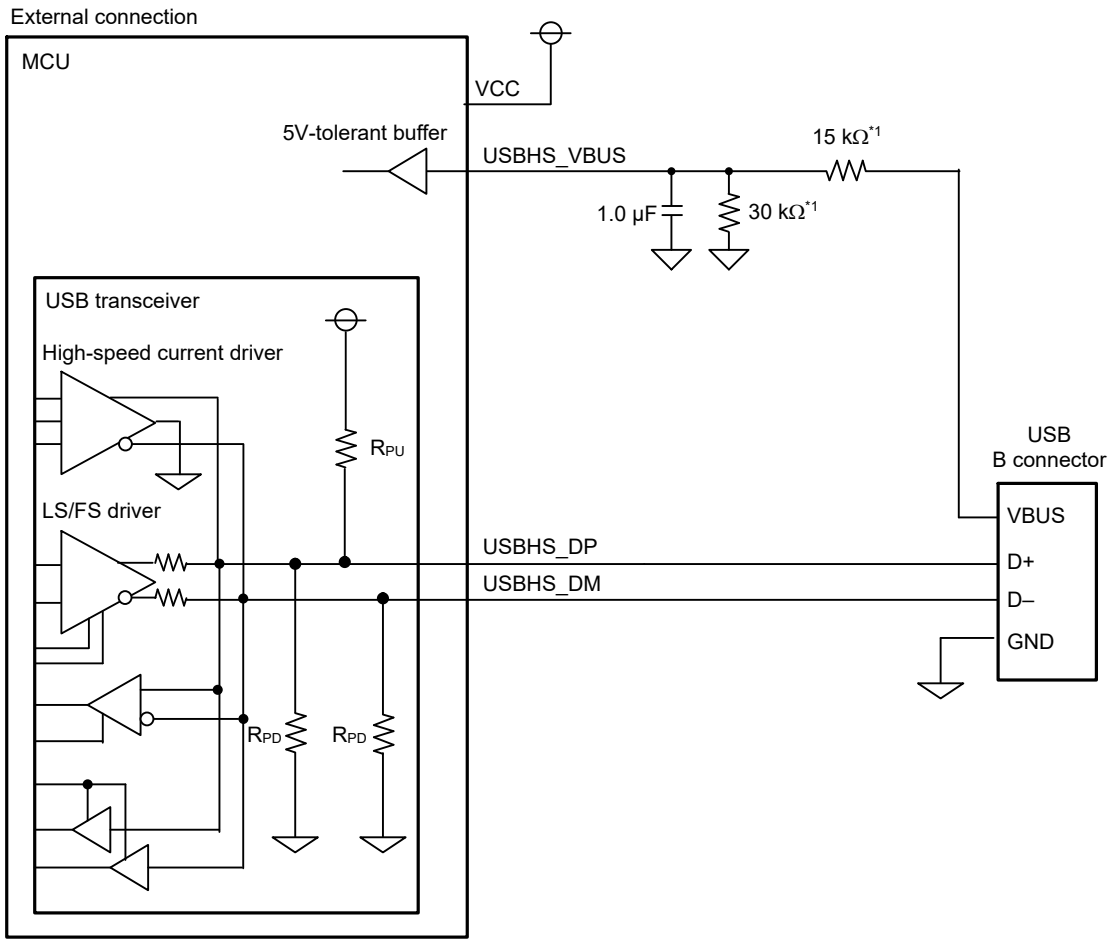


Figure 30.27 Example OTG connection in a self-powered system

Figure 30.28 shows an example USB device connection in a self-powered system.



Note 1. The VBUS (5 V) can be directly connected to the MCU if the VCC power supply of the MCU is not turned off when the USB is connected. If the VCC power supply of the MCU is turned off, the VBUS should be less than 3.6 V.

Figure 30.28 Example device connection in self-powered system

Figure 30.29 shows an example USB device connection in a bus-powered system.

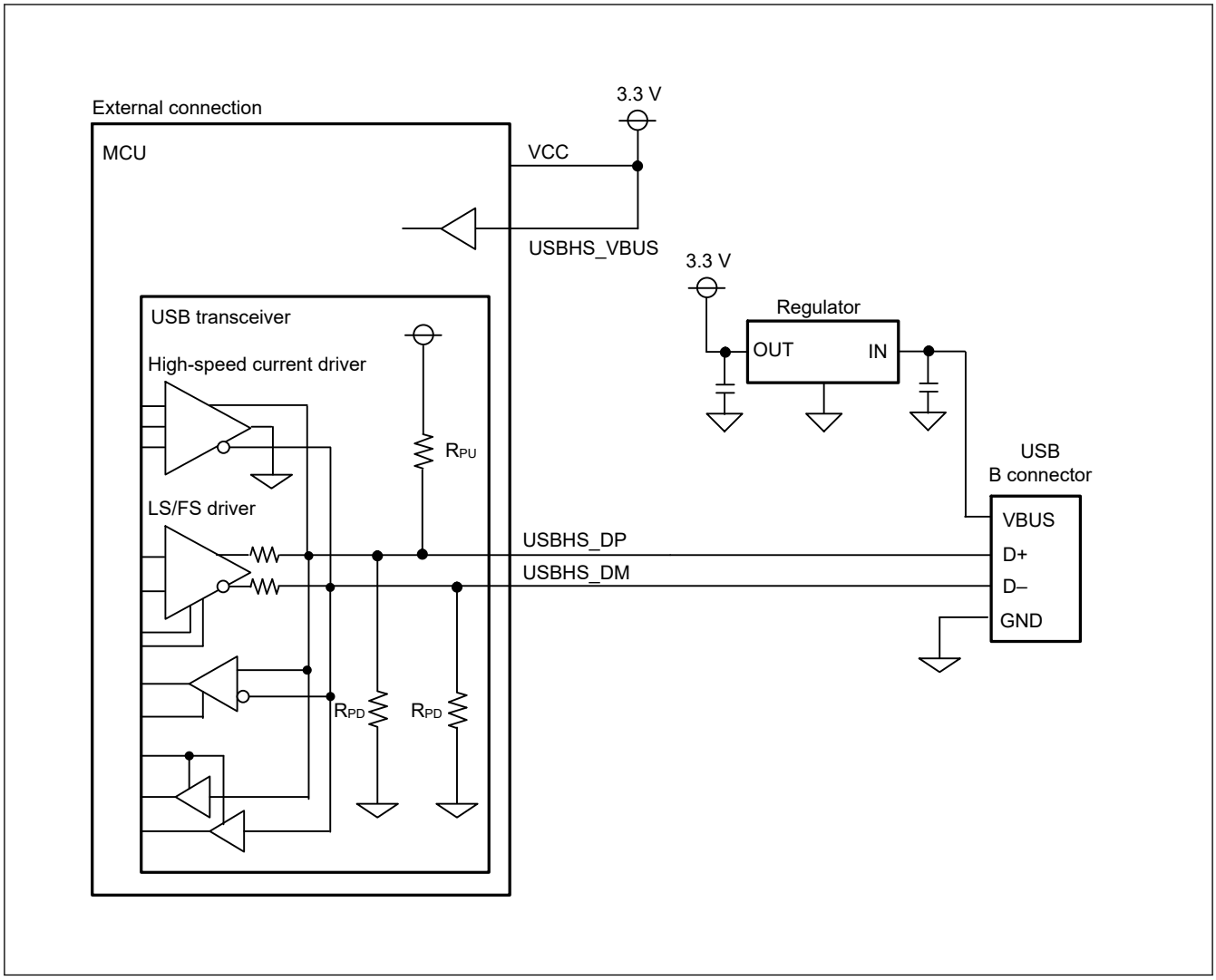


Figure 30.29 Example device connection in a bus-powered system

Figure 30.30 shows an example USB host connection.

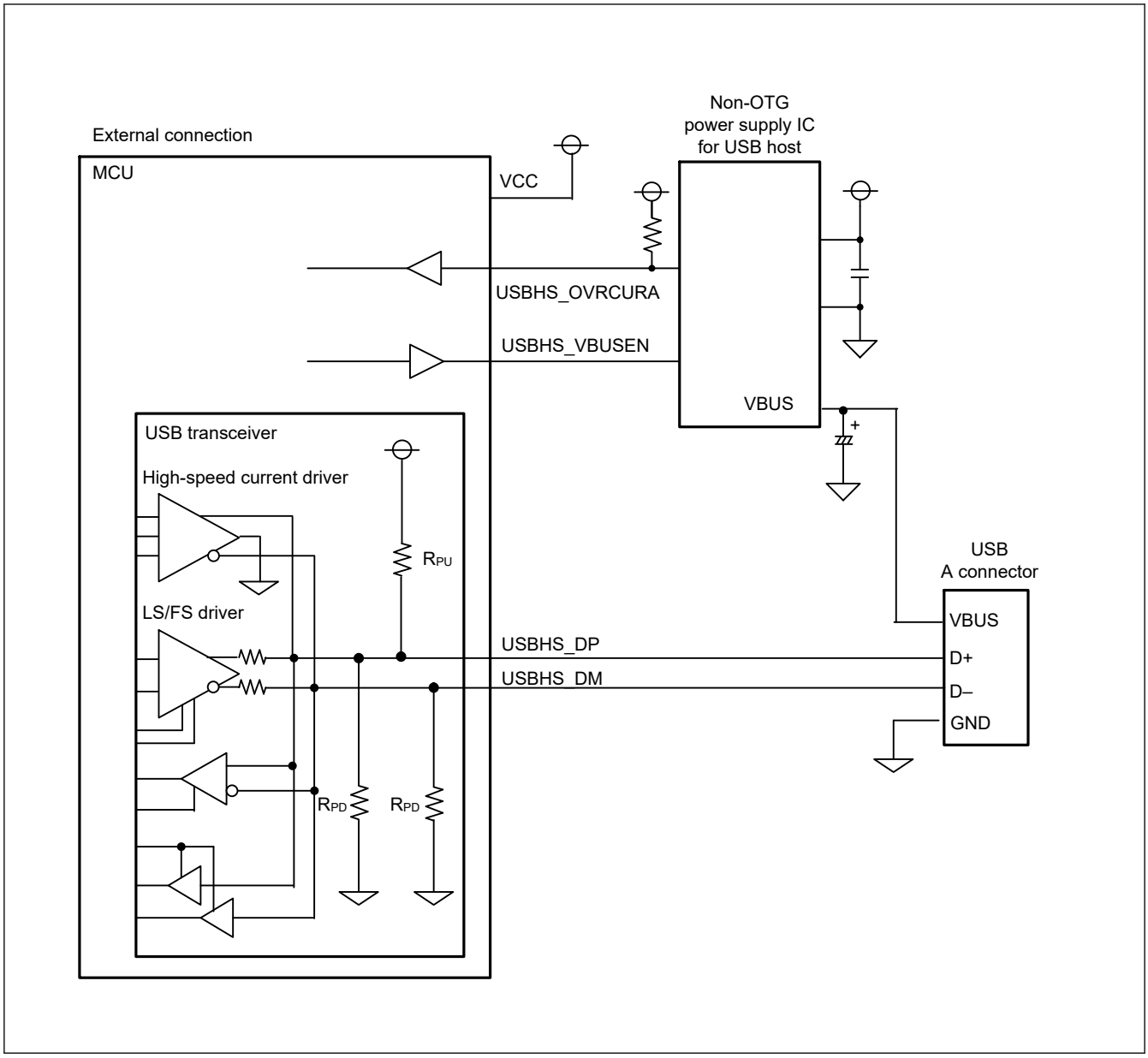


Figure 30.30 Example USB host connection

30.4 Usage Notes

30.4.1 Settings for the Module-Stop Function

USBHS operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The USBHS is initially stopped after reset. Releasing the module-stop state enables access to the registers. After releasing module stop, make settings required to activate the PHY circuit, including the input clock frequency setting, and then clear the PHYSET.DIRPD bit to 0. For details, see section 10, Low Power Modes .

30.4.2 Setup for Transitioning to Deep Software Standby Mode 1

Before transitioning to Deep Software Standby mode 1, clear the DVSTCTR0.VBUSEN bit to 0.

30.4.3 Clearing the Interrupt Status Register on Exiting Software Standby Mode

Because the input buffer is always enabled in Software Standby mode, an unexpected interrupt might occur under the following conditions:

- When the interrupt is enabled in Normal mode

- When the interrupt is disabled in Software Standby mode
- When the input level of the pin that cancels Software Standby mode is changed in Software Standby mode

These conditions might cause the associated interrupt flag in the Interrupt Status Register to set unexpectedly. After the MCU exits Software Standby mode, the unexpected interrupt might be sent to the interrupt controller. To avoid this, always clear the INTSTS0 and INTSTS1 registers in the canceling sequence.

30.4.4 Clearing the Interrupt Status Register after Setting Up the Port Function

The input buffer is disabled before the PmnPFS.PSEL and PmnPFS.PMR ports are set up, so the internal signal is fixed high or low. The input buffer is enabled after the port is set so that the external pin state is propagated to the MCU. An unexpected interrupt might occur at this time, causing the VBINT and OVRCR bits in INTSTS0 and INTSTS1, or other interrupt status flags to set to 1. To avoid a malfunction, always clear the INTSTS0 and INTSTS1 registers after setting up the port.

30.4.5 Restriction of register access

When accesing USBHS register consecutively and writing to Deep Software Standby-related register (DPUSR0R, DPUSR1R, DPUSR2R and DPUSRCR), there are constraints between Deep Software Standby-related register and last accessed USBHS register. At least, one of the following two constraints must be performed.

- Set an interval specified in the Table 30.36 between the last accessed USBHS register and write access to the Deep Software Standby-related register.

Table 30.36 Required interval before write access to Deep Software Standby related registers.

Last accessed USBHS register	Write	Read
Other registers than following	BWAIT[3:0] + 3	BWAIT[3:0] + 3
DPUSR0R, DPUSR1R	130	BWAIT[3:0] + 3
DPUSR2R, DPUSRCR	130	7

Note: Unit: PCLKA cycles

[Examples]

1. Performing read access to other register, then write access to Deep Software Standby-related register. Required interval is BWAIT[3:0] + 3 cycles or longer.
2. Performing write access to Deep Software Standby-related register, then write access to Deep Software Standby-related register. Required interval is 130 cycles or longer.

- Before performing write access to Deep Software Standby-related register, when the read access to USBHS register is performed, confirm the read value. When the write access to USBHS register is performed, read it and confirm the written value.

31. Serial Communications Interface (SCI)

This is the SCI_B version of the SCI peripheral module.

SCI_B is referred to as SCI in this chapter.

31.1 Overview

The Serial Communications Interface (SCI) × 6 channels have asynchronous and synchronous serial interfaces:

- Asynchronous interfaces (UART and Asynchronous Communications Interface Adapter (ACIA))
- 8-bit clock synchronous interface
- Simple IIC (master-only)
- Simple SPI
- Simple LIN
- Smart card interface
- Manchester interface

The smart card interface complies with the ISO/IEC 7816-3 standard for electronic signals and transmission protocol. SCIn (n = 0 to 4, 9) has FIFO buffers to enable continuous and full-duplex communication, and the data transfer speed can be configured independently using an on-chip baud rate generator.

In this section, PCLK refers to PCLKA.

Table 31.1 lists the SCI specifications, Table 31.2 shows the functions of each SCI channel, Figure 31.1 shows a block diagram of SCI, Figure 31.2 shows a clock source selector block diagram, and Table 31.3 lists the I/O pins.

Table 31.1 SCI specifications (1 of 4)

Parameter	Specifications
Number of modules	6 (SCIn (n = 0 to 4, 9))
Serial communication modes	• Asynchronous • Clock synchronous • Simple IIC • Simple SPI • Simple LIN (SCIn (n = 0, 1)) • Smart card interface • Manchester interface (SCIn (n = 0))
Operation clock (TCLK)	Synchronized clock (PCLK) or independent clock (SCICLK) can be selected
Transfer speed	Bit rate specifiable with the on-chip baud rate generator
Full-duplex communications	• Transmitter: Continuous transmission possible using double-buffering • Receiver: Continuous reception possible using double-buffering
Half-duplex communications	Half-duplex communication is possible by using only TXDn pins
Data transfer	Selectable as LSB-first or MSB-first transfer
Inverter for communication pins (RXDn, TXDn)	Selectable inverter for each pins (RXDn, TXDn)
Interrupt sources	Transmit end, transmit data empty, receive data full, receive error, receive data ready, address match. Break Field detection/output, Bus collision detection, Active edge detection. (SCIn (n = 0, 1)). Completion of generation of a start condition, restart condition, or stop condition. (for simple IIC mode)
Loop Back function	Self-diagnosis of communication function by SCI internal transmission / reception is possible
Synchronizer Bypass function	Ability to bypass synchronization circuit between bus clock (PCLK) and operation clock (TCLK)
Module-stop function	Module-stop state can be set for each channels to reduce power consumption
Reception in Software Standby mode	Only SCI0 is available

Table 31.1 SCI specifications (2 of 4)

Parameter		Specifications
Clock synchronous mode	Data length	8 bits
	Adjustment of receive sampling timing	Adjustable receive sampling timing after the default timing in master mode only when using internal clock
	Receive error detection	Overrun error
	Clock source	Selectable to internal clock (master mode) or external clock (slave mode)
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Hardware flow control	Transmission and reception controllable with CTSn_RTSn pins
	Transmission and reception	Selectable to 1-stage register or 16-stage FIFO
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Adjustment of receive sampling timing	Adjustable receive sampling timing before/after the default timing
	Adjustment of transmit timing	Adjustable edge timing of transmit waveform controlled by the setting value of registers.
	Parity	Even parity, odd parity, or no parity
	Receive error detection	• Parity error • Overrun error • Framing error
	Hardware flow control	Transmission and reception controllable with CTSn_RTSn pin and CTSn pin
	Transmission and reception	Selectable to 1-stage register or 16-stage FIFO
	Address match	Interrupt request/event output can be issued upon detecting a match between received data and the value in the compare match register
	Start-bit detection	Selectable to low level or falling edge detection
	Break detection	Breaks from framing errors detectable by read from CSR register
	Clock source	Selectable to internal or external clock. Transfer rate clock input from the GPT can be used. (SCIn (n = 1, 2))
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Multi-processor communications function	Serial communication enabled among multiple processors
	RS-485 driver control function	Output DEn signal to enable external transceiver transmit mode
	Noise cancellation	Digital noise filters included on signal paths from the RXDn pin inputs
Smart card interface mode	Error processing	Error signal can be automatically transmitted upon detecting a parity error during reception
		Data can be automatically retransmitted upon receiving an error signal during transmission
	Data type	Both direct and inverse convention supported
Manchester mode	Communication format	Manchester code with the preface and the Start Bit added
	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity function	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, framing, Manchester errors
	Hardware flow control	Transmission and reception controllable with CTSn_RTSn pin and CTSn pin
	Clock source	Only internal clock can be used.

Table 31.1 SCI specifications (3 of 4)

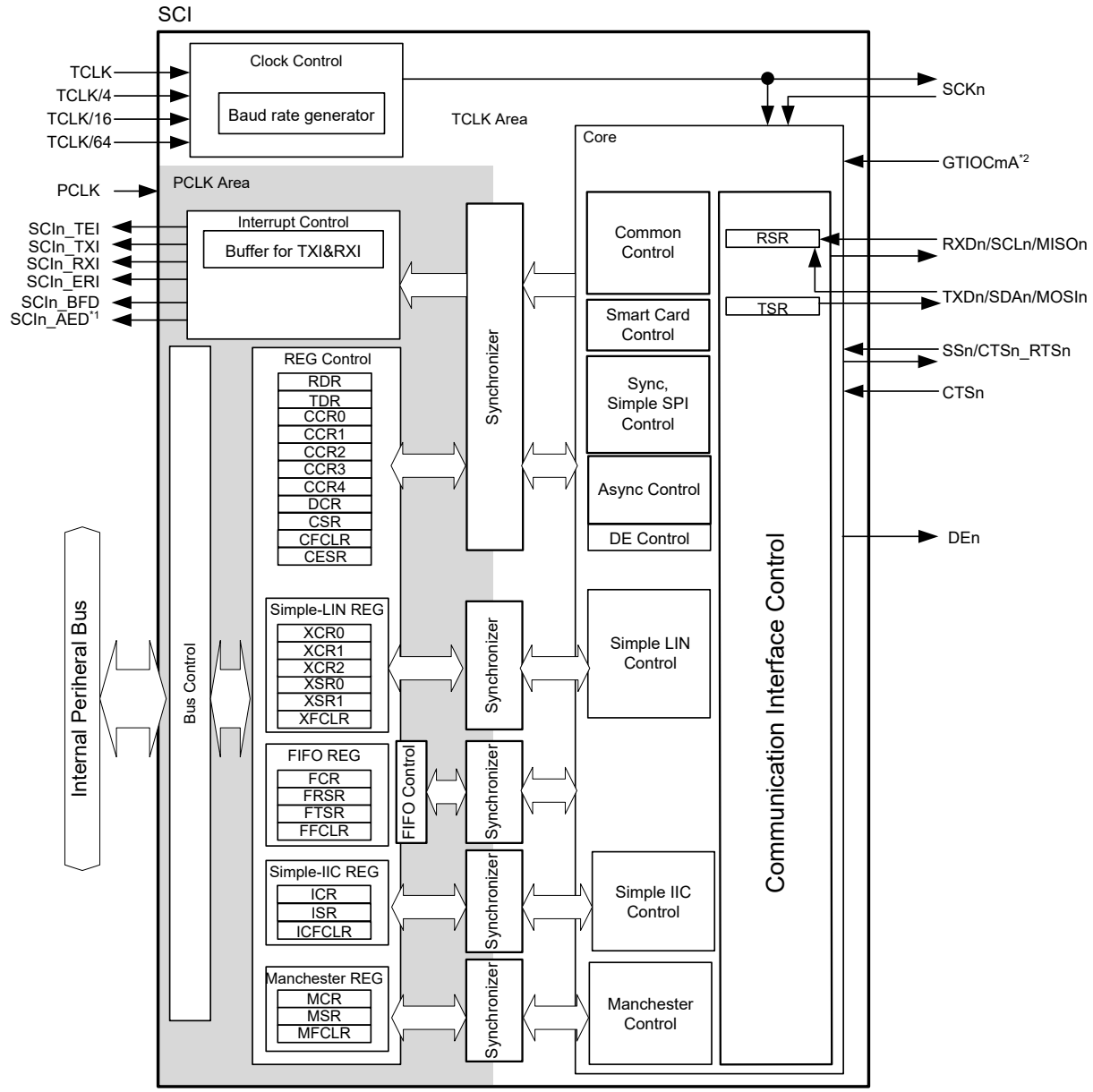
Parameter		Specifications
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Multi-processor communication function	Serial communication among multiple processors
	Manchester encoding / decoding function	Function to perform Manchester encoding / decoding of transmission / reception data and communicate using Manchester code
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters
	Preface setting / detection function	The function outputs the configured the preface pattern and detects it.
	Start Bit setting / detection function	The function outputs the configured the Start Bit pattern and detects it.
	Reception retiming function	Timing correction is performed for each bit of the received signal
Simple IIC mode	Transfer format	I^2C bus format (MSB-first only)
	Operating mode	Master (single-master operation only)
	Transfer rate	Up to 400 kbps
	Noise cancellation	The signal paths from input on the SCLn and SDAn pins incorporate digital noise filters and provide an adjustable interval for noise cancellation
Simple SPI mode	Data length	8 bits
	Error detection	Overrun error
	Clock source	Selectable to internal clock (master mode) or external clock (slave mode)
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Transmission / Reception	Selectable either 1 stage register or 16-stage FIFO
	Adjustment of receive sampling timing	Adjustable receive sampling timing after the default timing in master mode only when using internal clock
	SSn input pin function	High impedance state can be invoked on the output pins by driving the SSn pin high.
	Clock settings	Configurable among four clock phase and clock polarity settings
Simple LIN mode	Start Frame Transmission	• Break Field output possible, Break Field output complete interrupt output possible • Bus collision detection possible, bus collision detection interrupt output possible
	Start Frame Reception	• Break Field detectable, Break Field detected interrupt output possible • Control Field 0/1 data comparison function • Control Field 1 can set two types of comparison data of primary and secondary • Priority interrupt bit can be set in Control Field 1 • Handling of Start Frames that do not include a Break Field • Handling of Start Frames that do not include a Control Field 0 • Bit rate measurement function
	Input/Output control function	• Selectable polarity for TXDn and RXDn signals • Selection of a digital filter for the RXDn signal • Half-duplex operation employing RXDn and TXDn signals multiplexed on the same pin • Selectable timing for the sampling of data received through RXDn
Bit rate modulation function		Error reduction through correction of outputs from the on-chip baud rate generator
Event link function		Error event output for receive error or error signal detection (SCIn_ERI) (n = 0 to 4, 9)
		Receive data full event output (SCIn_RXI) (n = 0 to 4, 9)
		Transmit data empty event output (SCIn_TXI) (n = 0 to 4, 9)
		Address match event output (SCIn_AM) (n = 0 to 4, 9)

Table 31.1 SCI specifications (4 of 4)

Parameter	Specifications
	Active edge detection event output (SCIn_AED) (n = 0, 1)
	Transmit end event output (SCIn_TEI) (n = 0 to 4, 9)
TrustZone Filter	Security and Privilege attribution can be set for each channels

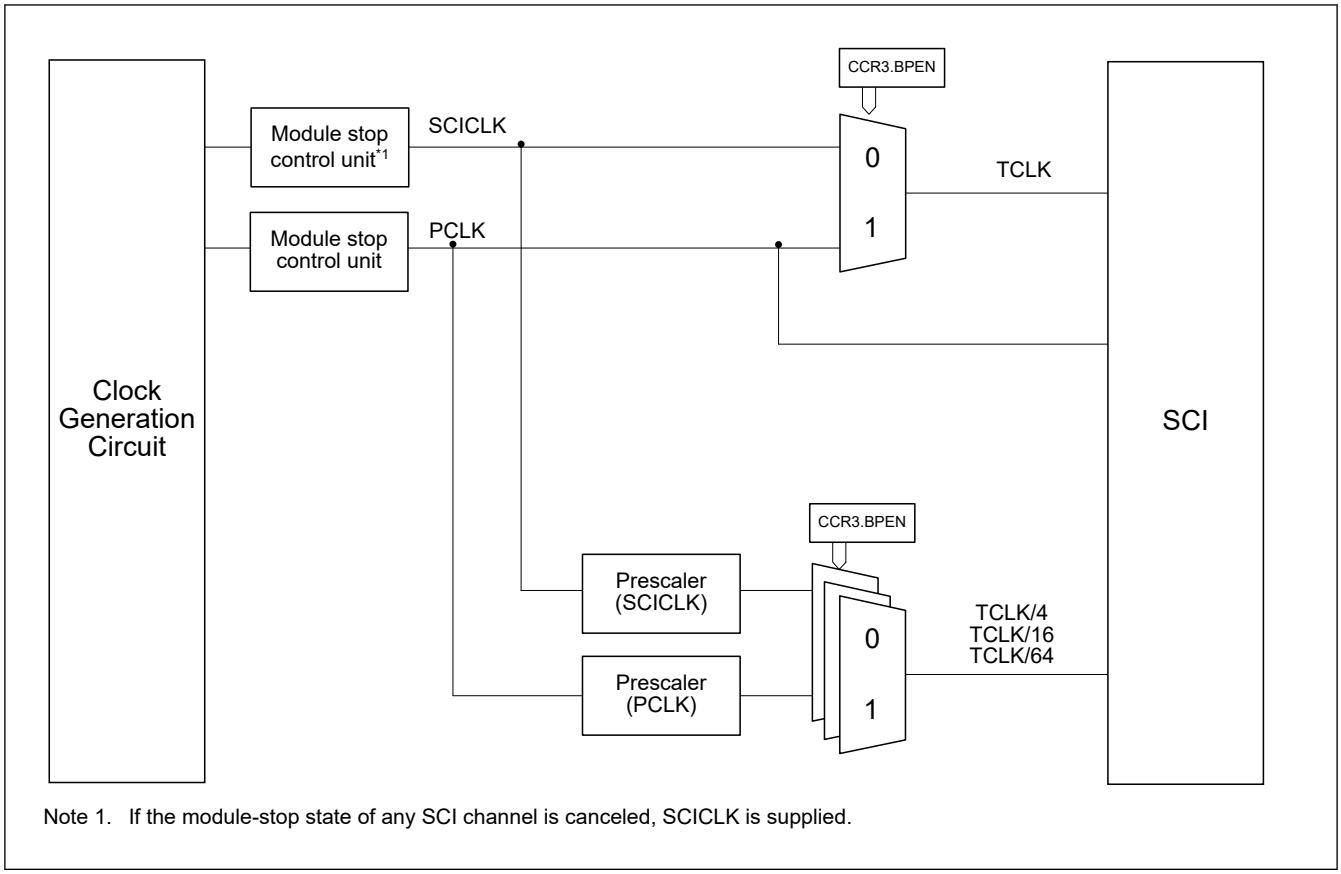
Table 31.2 Functions of each SCI channel

Item	SCI0	SCI1	SCI2	SCI3	SCI4	SCI9
Asynchronous mode	Available					
Clock synchronous mode	Available					
Smart card interface mode	Available					
Simple IIC mode	Available					
Simple SPI mode	Available					
FIFO mode	Available (16-stage)					
Address match	Available					
Manchester mode	Available	Not Available	Not Available	Not Available	Not Available	Not Available.
Simple LIN mode	Available	Available	Not Available	Not Available	Not Available	Not Available
GPT clock input	Not Available	GTIOC3A GTIOC4A	GTIOC6A GTIOC7A	Not Available	Not Available	Not Available



Note: n = 0 to 4, 9
Note: Only SCI0 and SCI1 support Simple LIN and only SCI0 supports Manchester
Note 1. n = 0, 1
Note 2. m = 3, 4, 6, 7

Figure 31.1 SCI block diagram



Note 1. If the module-stop state of any SCI channel is canceled, SCICLK is supplied.

Figure 31.2 Clock source selector block diagram

Table 31.3 SCI I/O pins

Function	Pin name	Input/ Output	Description
SCIn (n = 0 to 4, 9)	RXDn/SCLn/MISOn	Input/ Output	SCIn receive data input SCIn I^2C clock input/output SCIn slave transmit data input/output
	TXDn/SDAn/MOSIn	Input/ Output	SCIn transmit data output SCIn I^2C data input/output SCIn master transmit data input/output
	SSn/CTSn_RTSn	Input/ Output	SCIn chip select input, active-low SCIn transfer start control input/output, active-low
	CTSn	Input	SCIn transfer start control input, active-low
	DEn	Output	Driver Enable signal output
	SCKn	Input/ Output	SCIn clock input/output

31.2 Register Descriptions

31.2.1 RSR : Receive Shift Register

RSR is a shift register that receives serial data input from the RXDn pin and converts it into parallel data. When one frame of data is received, the data is automatically transferred to the RDR register. The RSR register cannot be directly accessed by the CPU.

31.2.2 RDR/RDR_BY : Receive Data Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 4, 9)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	FER	PER	—	—	ORER	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	FFER	FPER	DR	MPB	RDAT[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	RDAT[8:0]	Serial receive data RDAT is a 9-bit register for storing received data. Received data is stored in [6:0] when 7-bit data is selected, in [7:0] when 8-bit data is selected, and in [8:0] when 9-bit data is selected. And 0 is stored in the unused bit. Use RDR_BY for byte access to RDR[7:0].	R
9	MPB	Multi-processor flag 0: Data transmission cycles 1: ID transmission cycles	R
10	DR	Receive data ready flag FRSR.DR can be read.	R
11	FPER	FIFO parity error flag Valid only in Asynchronous mode 0: There is no parity error in the data read from the receive-FIFO 1: There is parity error in the data read from the receive-FIFO	R
12	FFER	FIFO framing error flag Valid only in Asynchronous mode 0: There is no framing error in the data read from the receive-FIFO 1: There is framing error in the data read from the receive-FIFO	R
23:13	—	These bits are read as 0.	R
24	ORER	Overrun Error flag CSR.ORER can be read.	R
26:25	—	These bits are read as 0.	R
27	PER	Parity error flag CSR.PER can be read.	R
28	FER	Framing error flag CSR.FER can be read.	R
31:29	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

In FIFO mode (CCR3.FM = 1), the RDR/RDR_BY is 16-stage FIFO buffer configuration.

When using FIFO mode, use RDR for 32-bit access.

RDAT[8:0] bits (Serial receive data)

After one frame of data is received, the received data is transferred from the RSR register to this register, thus allowing the RSR register to receive the next data.

The RSR and RDR registers have a double-buffered construction to enable continuous reception.

For Non-FIFO mode, read RDR only once when a receive data full interrupt (SCIn_RXI) request is issued. Without reading received data from RDR, if the next one frame is received, an overrun error occurs.

For FIFO mode, continuous reception is executed until 16 stages are stored. If data is read when there is no received data in the receive-FIFO(RDR), the value is undefined. When the receive-FIFO (RDR) are full of received data, subsequent serial receive data is lost.

The CPU cannot write to RDR.

0 is stored in the bit position which isn't received (RDR.bit8 or RDR.bit7) at the time of 7bit or 8bit communication of Asynchronous and Manchester mode.

MPB bit (Multi-processor flag)

In Asynchronous mode and Manchester mode, during multi-processor communication (CCR3.MP = 1), the value of the multi-processor bit corresponding to the received data (RDAT[8:0]) can be read.

FPER bit (FIFO parity error flag)

Indicates whether the data read from the receive-FIFO has a parity error.

0 is stored for non-FIFO mode.

FFER bit (FIFO framing error flag)

Indicates whether the data read from receive-FIFO has a framing error.

0 is stored for non-FIFO mode.

31.2.3 TDR/TDRLL/TDRLH : Transmit Data Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 4, 9)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x04(TDR/TDRLL)
0x05(TDRLH)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TSYNC	—	—	MPBT	TDAT[8:0]								
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
8:0	TDAT[8:0]	Serial transmit data TDAT is a 9-bit register for setting transmit data. Transmit data is set in [6:0] when 7-bit data is selected, in [7:0] when 8-bit data is selected, and in [8:0] when 9-bit data is selected. When byte access, write TDR [15:8] and then write TDR [7:0]. Use TDRLL for byte access to TDR[7:0], TDRLH for byte access to TDR[15:8].	R/W
9	MPBT	Multi-processor transfer bit flag Value of the multi-processor bit in the transmission frame. This bit is use in Asynchronous and Manchester mode. When writing to this bit when not used, write the initial value. 0: Data transmission cycles 1: ID transmission cycles	R/W
11:10	—	These bits are read as 1. The write value should be 1.	R/W
12	TSYNC	Transmit SYNC data It is valid when MCR.SBSEL = 1 and MCR.SYNSEL = 1 in Manchester mode. When this bit is not used, write the initial value. 0: The Start Bit is transmitted as DATA SYNC. 1: The Start Bit is transmitted as COMMAND SYNC.	R/W
31:13	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-3, P-TYPE-3

In FIFO mode (CCR3.FM = 1), the TDR/TDRLL/TDRLH is 16-stage FIFO buffer configuration.

When using FIFO mode, use TDR for 32-bit access.

TDAT[8:0] bits (Serial transmit data)

The is a 9-bit register for storing transmit data.

When empty space is detected in the TSR register, the transmit data stored in the TDR/TDRLL/TDRLH registers is transferred to TSR, and transmitting is started.

The TSR and TDR/TDRLL/TDRLH registers have a double-buffered construction to realize continuous transmission. When the next data to be transmitted is stored in TDR/TDRLL/TDRLH after one frame of data has been transmitted, the transmitting operation is continued by transfer to the TSR register.

When the SCI detects that the transmit shift register (TSR) is empty, it transmits data written in the transmit-FIFO (TDR/TDRLL/TDRLH) into TSR and starts serial transmission. Continuous serial transmission is executed until there is no transmit data left in the transmit-FIFO (TDR/TDRLL/TDRLH).

For non-FIFO mode, when a transmit data empty interrupt (SCIn_TXI) request is issued and CCR0.TE is 1, write transmit data to the TDR only once.

For FIFO mode, when transmit-FIFO is full of transmit data 16 frames, no more data can be written. If writing of new data is attempted, the data is ignored.

The TDR/TDRLL/TDRLH register can always be read / written from the CPU. And when byte access, write TDR/TDRLL/TDRLH[15:8] and then write TDR/TDRLL/TDRLH[7:0].

MPBT bit (Multi-processor transfer bit flag)

Selects the multi processor bit of transmit frame.

TSYNC bit (Transmit SYNC data)

When Manchester mode and MCR.SBSEL = 1 and MCR.SYNSEL = 1, the type of SYNC selected according to the TSYNC bit becomes the Start Bit of the transmission frame.

31.2.4 TSR : Transmit Shift Register

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI first automatically transfers transmit data from TDR to TSR, then sends the data to the TXDn pin. The CPU cannot directly access the TSR.

31.2.5 CCR0 : Common Control Register 0

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 4, 9)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	SSE	—	—	TEIE	TIE	—	—	—	RIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	IDSEL	DCME	MPIE	—	—	—	TE	—	—	—	RE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RE	Receive Enable 0: Serial reception is disabled 1: Serial reception is enabled	R/W*1 *3
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	TE	Transmit Enable 0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*1

Bit	Symbol	Function	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	MPIE	Multi-Processor Interrupt Enable Valid in Asynchronous mode and Manchester mode when CCR3.MP is 1. This bit should set 0 in Smart card interface mode. 0: Non-multi-processor reception 1: Multi-processor reception When the data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and non-multi-processor reception is resumed. If you want to continue receiving operation using the multiprocessor function, set this bit to 1 sufficiently earlier than receiving the STOP bit of the next received frame (consider the synchronization delay time).	R/W*2
9	DCME	Data Compare Match Enable Valid only in Asynchronous mode 0: Address match function is disabled 1: Address match function is enabled	R/W*2
10	IDSEL	ID Frame Select Valid only in Asynchronous mode with multi-processor 0: Compare data irrespective of the value of the MPB bit 1: Compare data only when the MPB bit is 1 (ID frame)	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
16	RIE	Receive Interrupt Enable 0: SCIn_RXI and SCIn_ERI interrupt requests are disabled 1: SCIn_RXI and SCIn_ERI interrupt requests are enabled	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	TIE	Transmit Interrupt Enable 0: SCIn_TXI interrupt request is disabled 1: SCIn_TXI interrupt request is enabled	R/W
21	TEIE	Transmit End Interrupt Enable This bit should set 0 in Smart card interface mode. 0: SCIn_TEI interrupt request is disabled 1: SCIn_TEI interrupt request is enabled	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
24	SSE	SSn Pin Function Enable Valid in Simple SPI mode. In slave mode (CCR3.CKE[1:0] = 1x), set 1 to this bit. 0: SSn pin function is disabled 1: SSn pin function is enabled	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. In Clock synchronous mode (CCR3.MOD[2:0] = 010b), Simple SPI mode (CCR3.MOD[2:0] = 011b), and Simple IIC mode (CCR3.MOD[2:0] = 100b), 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written in TE and RE. In other mode, writing is enabled under any condition.

Note 2. This bit is a bit that is cleared by hardware. Note that writing to a bit other than this bit with a bit manipulation instruction may cause this bit to be unintentionally set to 1 by a read-modify-write operation.

Note 3. In Clock synchronous mode or Simple SPI mode, and internal clock (master mode), receive-only setting is prohibited in the internal clock (master mode) (TE = 0 and RE = 1 setting prohibited).

RE bit (Receive Enable)

Enables or disables serial receive operation.

When the RE bit is set to 1, serial reception becomes possible after the synchronization delay time has elapsed in Asynchronous mode or the synchronous clock input in Clock synchronous mode or the neg-edge of RXDn in Manchester mode or start bit in Smart card interface mode.

Note that CCR3 should be set prior to setting the RE bit to 1 in order to designate the reception format.

Except Smart card interface mode, even if reception is halted by setting the RE bit to 0, the CSR. RDRF, FER, PER, ORER, MSR.MER, SBER, SYER, PFER, FRSR. DR flags are not affected, and the previous values is retained. In Smart card interface mode, even if reception is halted by setting the RE bit to 0, the CSR.FER, PER, ORER flags are not affected and

the previous value is retained. Also, to stop the reception operation, synchronization delay time is required from when the RE bit is set to 0 until the reception operation is stopped.

TE bit (Transmit Enable)

Enables or disables serial transmission.

When the TE bit is set to 1, serial transmission becomes possible after the synchronization delay time has elapsed. After the synchronization delay time, transmission is started by writing transmit data to TDR. Note that CCR3 should be set prior to setting the TE bit to 1 in order to designate the transmission format. In addition, the synchronization delay time is required until the transmission control circuit is stopped after the TE bit is set to 0.

MPIE bit (Multi-Processor Interrupt Enable)

When the MPIE bit is set to 1 and data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags (CSR.RDRF, ORER, FER, FRSR, DR, MSR.MER, SYER, PFER, SBER) are disabled.

When data with the multi-processor bit set to 1 is received, the MPIE is automatically cleared to 0, and normal reception is resumed. For details, see section 31.4. Multi-Processor Communication Function. If you want to continue receiving operation using the multi-processor function, set this bit to 1 sufficiently earlier than receiving the STOP bit of the next received frame.

When the receive data includes the MPB bit set to 0, the receive data is not transferred from the RSR to the RDR, a receive error is not detected, and setting the flags ORER, FER, MER, SYER, PFER, and SBER to 1 is disabled.

When the receive data includes the MPB bit set to 1, the MPIE bit is automatically cleared to 0, the SCIn_RXI and SCIn_ERI interrupt requests are enabled (if CCR0.RIE is set to 1), and setting the flags ORER, FER, MER, SYER, PFER, and SBER to 1 is enabled.

MPIE should be set to 0 if multi-processor communications function is not to be used.

DCME bit (Data Compare Match Enable)

The DCME bit selects whether the address match function (data compare match function) is enabled or not.

When DCME is 1, if SCI detects the match to the comparison data (CCR4.CMPD) with receive data, DCME is cleared automatically, and after that, SCI operation mode is in receive mode without data compare match function.

See section 31.3.6. Address Match (Receive Data Match Detection) Function.

The write value should be 0 other than Asynchronous mode.

IDSEL bit (ID Frame Select)

The IDSEL bit selects whether to compare data irrespective of the value of MPB bit or to compare data only when the MPB bit = 1 (ID frame) when the address match function is valid. Set at the same time as DCME.

RIE bit (Receive Interrupt Enable)

Enables or disables SCIn_RXI and SCIn_ERI interrupt requests.

SCIn_RXI and SCIn_ERI interrupt request is disabled by setting the RIE bit to 0.

An SCIn_ERI interrupt request can be canceled by reading 1 from the CSR.ORER, FER, or PER and then setting the flag to 0 or setting the RIE bit to 0.

In Manchester mode, the MER, SYER, PFER and SBER flags are also the cause of SCIn_ERI interrupt request, so the same processing is necessary. For details of these flags, see section 31.2.12. MCR : Manchester Control Register and section 31.2.21. MSR : Manchester Status Register.

TIE bit (Transmit Interrupt Enable)

Enables or disables SCIn_TXI interrupt request.

An SCIn_TXI interrupt request is disabled by setting the TIE bit to 0. At the beginning of transmission, set 1 to CCR0.TE and CCR0.TIE simultaneously. The SCIn_TXI interrupt request is then generated.

TEIE bit (Transmit End Interrupt Enable)

Enables or disables a SCIn_TEI interrupt request. A SCIn_TEI interrupt request is disabled by setting the TEIE bit to 0.

In simple IIC mode, the SCIn_TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STIn). In this case, the TEIE bit can be used to enable or disable the STIn.

SSE bit (SSn Pin Function Enable)

Set the SSE bit to 1 if the SSn pin is to be used in control of transmission and reception (in simple SPI mode).

Set the SSE bit to 0 in any other mode. Do not set both the SSE and CTSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

In the slave mode (CCR3.CKE[1:0] = 10 or 11), SSE should be set 1.

In the master mode (CCR3.CKE[1:0] = 00 or 01) and single-master, the SSn pin on the master side is not required to control reception and transmission, so SSE should be set 0.

31.2.6 CCR1 : Common Control Register 1

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 4, 9)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	NFM	NFEN	—	NFCS[2:0]			—	—	—	SHARPS	—	—	—	SPLP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	RINV	TINV	—	—	PM	PE	—	—	SPB2IO	SPB2DT	—	—	CTSPEN	CTSE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	CTSE	CTS Enable 0: CTS function is disabled (RTS output function is enabled) 1: CTS function is enabled	R/W
1	CTSPEN	CTS External Pin Enable 0: Alternate setting to use CTS and RTS functions as either 1 pin 1: Dedicated setting for separately using CTS and RTS functions with 2 pins	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	SPB2DT	Serial Port Break Data Select The output level of TXDn pin is selected when CCR0.TE = 0 and SPB2IO = 1.*1 0: When TINV is 0, low level is output in TXDn pin. When TINV is 1, high level is output in TXDn pin. 1: When TINV is 0, high level is output in TXDn pin. When TINV is 1, low level is output in TXDn pin.	R/W
5	SPB2IO	Serial Port Break I/O This bit selects whether the value of SPB2DT is output to TXDn pin when CCR0.TE = 0.*1 0: The value of SPB2DT bit is not output in TXDn pin 1: The value of SPB2DT bit is output in TXDn pin	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	PE	Parity Enable Valid only in Asynchronous mode and Manchester mode. In Smart Card Interface mode, set 1 to this bit. 0: When transmitting: Do not add parity bit When receiving: Do not check parity bit 1: When transmitting: Add parity bit When receiving: Check parity bit	R/W
9	PM	Parity Mode Valid only when the PE bit is 1 0: Selects even parity 1: Selects odd parity	R/W

Bit	Symbol	Function	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	TINV	TXD Invert 0: Transmit data is not inverted and output to TXDn*2 1: Transmit data is inverted and output to TXDn	R/W
13	RINV	RXD Invert 0: Received data from RXDn is not inverted and input*2 1: Received data from RXDn is inverted and input	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	SPLP	Loopback Control This bit can be used for internal clock operation in Asynchronous mode, internal mode operation in Manchester mode, internal clock operation in Clock synchronous mode. 0: Normal mode 1: Loopback mode	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	SHARPS	Half-Duplex Communication Select In Simple IIC mode, Smart Card Interface mode, or Simple SPI mode, this bit should be set to 0. 0: TXDn pin, RXDn pin independent 1: TXDn/RXDn pin combination use (half-duplex communication using TXDn pin)	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
26:24	NFCS[2:0]	Noise Filter Clock Select Valid in Asynchronous mode and Manchester mode, Simple LIN mode, and Simple IIC mode. In Simple IIC mode, 000 setting is prohibited. The on-chip baud rate generator source clock means the clock selected by CCR2.CKS [1:0]. Select for the clock source of the noise filter. 0 0 0: The base clock signal divided by 1 0 0 1: The on-chip baud rate generator source clock divided by 1 0 1 0: The on-chip baud rate generator source clock divided by 2 0 1 1: The on-chip baud rate generator source clock divided by 4 1 0 0: The on-chip baud rate generator source clock divided by 8 others: Setting prohibited	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
28	NFEN	Digital Noise Filter Function Enable Valid in Asynchronous mode, Manchester mode, Simple LIN mode, and Simple IIC mode 0: In Asynchronous, Manchester, Simple LIN modes: Disable noise cancellation function for RXDn input signal In Simple IIC mode: Disable noise cancellation function for SCLn and SDAn input signals 1: In Asynchronous, Manchester, Simple LIN modes: Enable noise cancellation function for RXDn input signal In Simple IIC mode: Enable noise cancellation function for SCLn and SDAn input signals	R/W
29	NFM	Noise Filter Mode Valid in asynchronous mode and ABCSE2 = 1 0: 3-point matching mode 1: Majority vote mode	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Use this bit in Asynchronous mode and Manchester mode only. Operation in other mode is not guaranteed.
Note 2. RINV/TINV should be set to 0 in Smart card interface mode and Simple IIC mode.

CTSE bit (CTS Enable)

Set the CTSE bit to 1 if the SSn pin is to be used for inputting of the CTSn control signal to control of transmission and reception. The RTSn signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode, simple SPI mode, Simple LIN mode, and simple IIC mode. Do not set both the CTSE and SSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

CTSPEN bit (CTS External Pin Enable)

When CTSE is 1, select the pins usage method when using the CTS and RTS functions. Set this bit to 1 when assigning the CTS/RTS function to 2 pins and using them at the same time. Set it to 0 except in Asynchronous and Manchester modes.

Table 31.4 shows the relationship between the CTSE bit and CTSPEN bit settings and the functions of the CTSn_RTSn pin and CTSn pin.

Table 31.4 CTSE bit and CTSPEN bit settings and pin functions

CTSE bit	CTSPEN bit	CTSn_RTSn pin	CTSn pin
0	0	RTSn signal output	Not use
1	0	CTSn signal input	Not use
1	1	RTSn signal output	CTSn signal input
0	1	prohibition	prohibition

SPB2DT bit (Serial Port Break Data Select), SPB2IO bit (Serial Port Break I/O)

The TXDn pin status determined by combination of CCR0.TE bit, CCR1.SPB2IO bit, and CCR1.SPB2DT bit is indicated in Table 31.5.

Table 31.5 TXDn pin status

Value of CCR0.TE	Value of CCR1.SPB2IO	Value of CCR1.SPB2DT	TXDn pin status (when TINV is 0)
0	0	—	Hi-Z (initial value)
0	1	0	Low level output
0	1	1	High level output
1	—	—	Serial transmission data is output

Note: —: Don't care

PE bit (Parity Enable)

When the PE bit to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

In the multi-processor format, the parity bit is not added or checked regardless of the PE bit setting.

PM bit (Parity Mode)

The PM bit selects the parity mode for transmission and reception (even or odd). In Multi-processor mode, the PM bit is invalid.

For details on the usage of this bit in Smart card interface mode, see section 31.7.2. Data Format (Except in Block Transfer Mode).

TINV bit (TXD Invert), RINV bit (RXD Invert)

The data of RDR is controlled by RINV and CCR3.SINV. And the data from TXDn pin is controlled by TINV and CCR3.SINV. The control by RINV/TINV are done to communication pins (RXDn / TXDn), so they can control not only data-bits but also other bits (start bit, stop bit, parity bit). For details, see Figure 31.3.

During half-duplex communication and slave operation in Simple SPI mode, use the TXDn pin for reception, so set the inversion control of the received data with the TINV bit.

Note: Description and a timing chart of the SCI operation are provided when TINV = 0 and RINV = 0 (when the value of TINV and the value of RINV are not specified).

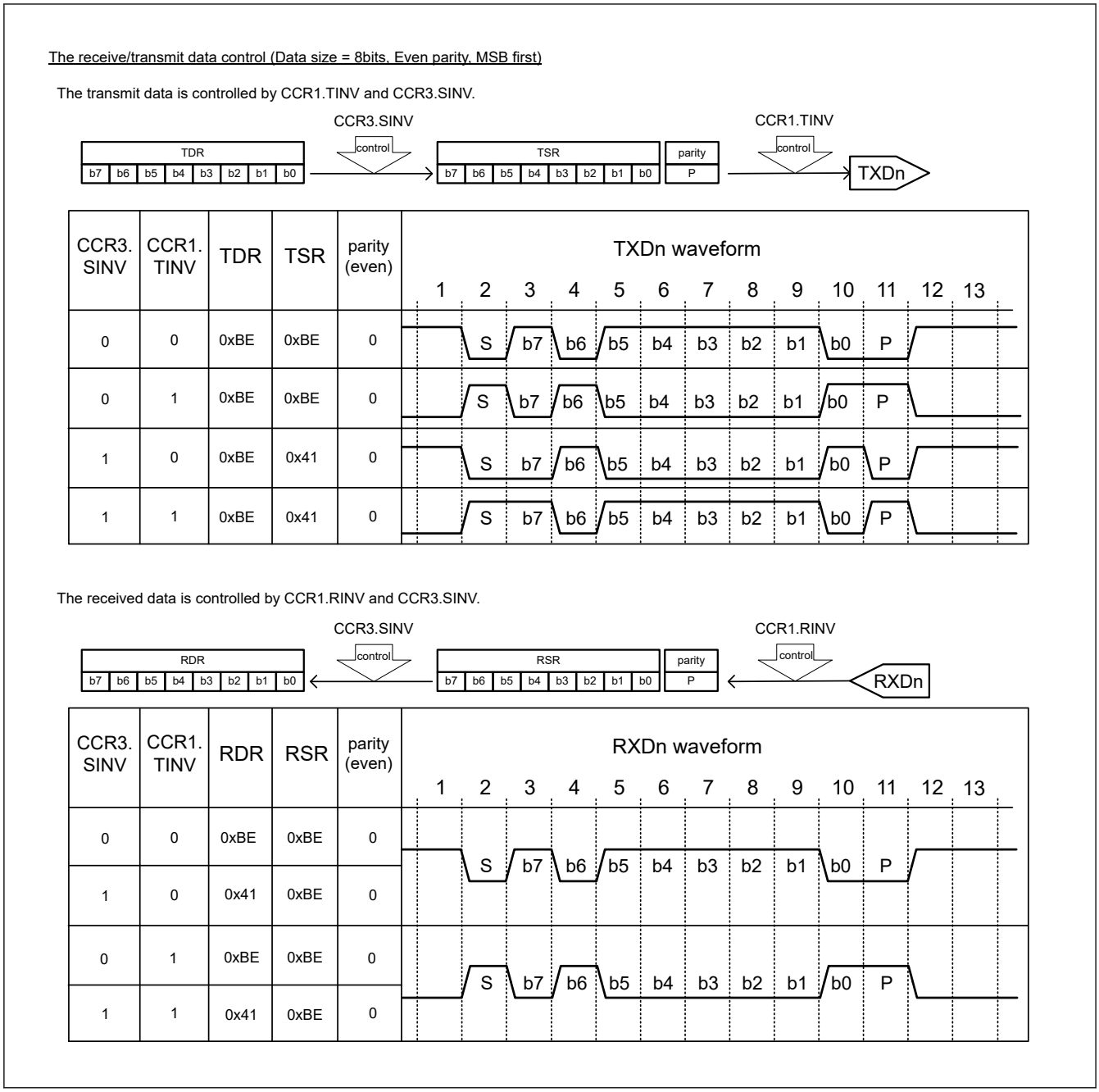


Figure 31.3 Example of receive or transmit data control

SPLP bit (Loopback Control)

When the SPLP bit is 1, SCI blocks the input path from RXDn and connects the output path to TXDn to the reception data register.

Transmit data can be inverted and received by combining it with TINV bit.

Set this bit to 0 for slave operation in Clock synchronous mode, when using an external clock in Asynchronous mode, and when in Simple LIN mode.

SHARPS bit (Half-Duplex Communication Select)

Setting the SHARPS bit to 1 enables half-duplex communication using the TXDn pin. However, the SHARPS bit cannot be used in Simple SPI mode, Simple IIC mode and Smart card interface mode.

If the SHARPS bit is set to 1 and CCR0.TE = 1, CCR0.RE = 0, the TXDn pin becomes the communication output. If this bit is set to 1 and CCR0.TE = 0, CCR0.RE = 1, the TXDn pin becomes the communication input. For details, see section 31.17. Half-Duplex communication Function.

NFCS[2:0] bits (Noise Filter Clock Select)

The NFCS[2:0] bits select the sampling clock for the digital noise filter.

To use the noise filter in Asynchronous mode, Manchester mode, and Simple LIN mode set the NFCS[2:0] bits from 000b to 100b. In Simple IIC mode, set the bits to a value in the range from 001b to 100b. In particular, if the ABCSE or ABCSE2 bit is set to 1 in asynchronous mode, set it to 000b or 001b.

NFEN bit (Digital Noise Filter Function Enable)

The NFEN bit enables or disables the digital noise filter function. When the function is enabled, noise cancellation is applied to the RXDn input signal in Asynchronous mode, Manchester mode, Simple LIN mode, and noise cancellation is applied to the SDAn and SCLn input signals in Simple IIC mode. In any mode other than above, set the NFEN bit to 0 to disable the digital noise filter function. When the function is disabled, input signals are transferred as is, as internal signals.

NFM bit (Noise Filter Mode)

The NFM bit selects the operation mode of the digital noise filter function. For details, see section 31.14. Noise Cancellation Function.

31.2.7 CCR2 : Common Control Register 2

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 4, 9)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MDDR[7:0]								—	—	CKS[1:0]		—	—	—	BRME
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BRR[7:0]								ABCSE2	ABCSE	ABCS	BGDM	—	BCP[2:0]		
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
2:0	BCP[2:0]	Base Clock Pulse Selects the number of base clock cycles in Smart card interface mode. 0 0 0: 93 clock cycles (S = 93)*1 0 0 1: 128 clock cycles (S = 128)*1 0 1 0: 186 clock cycles (S = 186)*1 0 1 1: 512 clock cycles (S = 512)*1 1 0 0: 32 clock cycles (S = 32)*1 (Initial value) 1 0 1: 64 clock cycles (S = 64)*1 1 1 0: 372 clock cycles (S = 372)*1 1 1 1: 256 clock cycles (S = 256)*1	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	BGDM	Baud Rate Generator Double-Speed Mode Select Valid in Asynchronous/Manchester/Clock-synchronous/Simple SPI mode and when CCR3.CKE[1] = 0. 0: Baud rate generator outputs the clock with single frequency 1: Baud rate generator outputs the clock with doubled frequency	R/W
5	ABCS	Asynchronous Mode Base Clock Select Valid only in Asynchronous mode, Manchester mode, and Simple LIN mode 0: Selects 16 base clock cycles for 1-bit period 1: Selects 8 base clock cycles for 1-bit period	R/W

Bit	Symbol	Function	R/W
6	ABCSE	Asynchronous Mode Extended Base Clock Select Valid only in Asynchronous mode and when CCR3.CKE[1] = 0 0: Clock cycles for 1-bit period is determined by combination between CCR2.BGDM and CCR2.ABCS. 1: Baud rate is 6 base clock cycles for 1-bit period and the clock of a double frequency is output from the baud rate generator.	R/W
7	ABCSE2	Asynchronous Mode Extended Base Clock Select 2 Valid only in asynchronous mode and CCR3.CKE[1] = 0 0: Clock cycles for 1-bit period is decided with combination between CCR2.BGDM and CCR2.ABCS. 1: Baud rate is 4 base clock cycles for 1-bit period and the clock of a double frequency is output from the baud rate generator.	R/W
15:8	BRR[7:0]	Bit Rate Setting BRR is an 8-bit register that adjusts the bit rate.	R/W
16	BRME	Bit Rate Modulation Enable 0: Bit rate modulation function is disabled 1: Bit rate modulation function is enabled	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
21:20	CKS[1:0]	Clock Select 0 0: TCLK clock (n = 0)*2 0 1: TCLK/4 clock (n = 1)*2 1 0: TCLK/16 clock (n = 2)*2 1 1: TCLK/64 clock (n = 3)*2	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
31:24	MDDR[7:0]	Modulation Duty Setting MDDR corrects the bit rate adjusted by the BRR[7:0] bits.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. S is the value of S in BRR[7:0] bits explanation.
Note 2. n is the decimal notation of the value of n in BRR[7:0] bits explanation.

BCP[2:0] bits (Base Clock Pulse)

The BCP[2:0] bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.

For details, see section 31.7.4. Receive Data Sampling Timing and Reception Margin.

BGDM bit (Baud Rate Generator Double-Speed Mode Select)

The BGDM bit is valid when the on-chip baud rate generator is selected as the clock source (CCR3.CKE[1] = 0) in Asynchronous mode, Manchester mode, Clock synchronous mode, Simple SPI mode. When external clock is selected (CCR3.CKE[1] = 1), set it to 0. For the clock output from the baud rate generator, either single or doubled frequency can be selected. The base clock is generated by the clock output from the baud rate generator. When the BGDM bit is set to 1, the base clock cycle is halved, and the bit rate is doubled.

Set this bit to 0 in modes other than Asynchronous mode, Manchester mode, Clock synchronous mode, or Simple SPI mode.

ABCS bit (Asynchronous Mode Base Clock Select)

Selects the clock cycles for 1-bit period.

Set the ABCS bit to 0 in modes other than Asynchronous mode, Manchester mode, or Simple LIN mode.

ABCSE bit (Asynchronous Mode Extended Base Clock Select)

The pulse number for a base clock at 1-bit period is 6 and the clock of a double frequency is output from baud rate generator. Only when the bit rate is set to 6 dividing frequency of the bus clock can this bit be used to set CCR2.CKS [1:0] = 00b and BRR[7:0] = 0x00.

Set it to 0 in modes other than Asynchronous mode. Even in Asynchronous mode, set it to 0 when using external clock.

ABCSE2 bit (Asynchronous Mode Extended Base Clock Select 2)

The pulse number for a base clock at 1-bit period is 4 and the clock of a double frequency is output from baud rate generator. Only when the bit rate is set to 4 dividing frequency of the bus clock, please use this bit and set CCR2.CKS [1:0] = 00 and BRR[7:0] = 0.

Set it to 0 in modes other than asynchronous mode. Even in asynchronous mode, set it to 0 when using external clock.

Table 31.6 Base clock cycle number per 1-bit

ABCSE2	ABCSE	ABCS	BGDM	The base clock cycles /1bit	The frequency of the baud rate generator
0	0	0	0	16	×1
0	0	0	1	16	×2
0	0	1	0	8	×1
0	0	1	1	8	×2
0	1	— (don't care)	— (don't care)	6	×2
1	0	— (don't care)	— (don't care)	4	×2

BRR[7:0] bit (Bit Rate Setting)

BRR is an 8-bit register that adjusts the bit rate.

SCI has independent baud rate generator control, different bit rates can be set for each. Table 31.7 shows the relation-ship between the setting (N) in the BRR and the bit rate (B) for Asynchronous mode, multiprocessor transfer, Manchester mode, clock synchronous mode, smart card interface mode, simple SPI mode, and simple IIC mode.

Table 31.7 Relationship between N Setting in BRR and Bit Rate B

Mode	CCR2 settings				BRR[7:0] setting	Error*4
	BGDM bit	ABCS bit	ABCSE bit	ABCSE 2 bit		
Asynchronous, multi-processor, Manchester*2, Simple-LIN*3	0	0	0	0	$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \left\{\frac{TCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1\right\} \times 100$
	1	0	0	0	$N = \frac{TCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \left\{\frac{TCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1\right\} \times 100$
	0	1	0	0		
	1	1	0	0	$N = \frac{TCLK \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \left\{\frac{TCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1\right\} \times 100$
	Don't care	Don't care	1	0	$N = \frac{TCLK \times 10^6}{12 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \left\{\frac{TCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times (N+1)} - 1\right\} \times 100$
	Don't care	Don't care	0	1	$N = \frac{TCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \left\{\frac{TCLK \times 10^6}{B \times 8 \times 2^{2n-1} \times (N+1)} - 1\right\} \times 100$
Clock synchronous, simple SPI	0	0 (Initial value)	0 (Initial value)	0 (Initial value)	$N = \frac{TCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	—
	1	0 (Initial value)	0 (Initial value)	0 (Initial value)	$N = \frac{TCLK \times 10^6}{4 \times 2^{2n-1} \times B} - 1$	—
Smart card interface					$N = \frac{TCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error\ (\%) = \left\{\frac{TCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1\right\} \times 100$
Simple IIC*1					$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	—

Note: B: Bit rate (bps)
N: BRR setting for on-chip baud rate generator (0 ≤ N ≤ 255)
TCLK: Operating frequency (MHz)
n and S: Determined by the settings of the CCR2 registers as listed in the Table 31.9 and Table 31.10. Please be careful about $2^{(2n+1)}$ is used in the expression for Smart card interface, $2^{(2n-1)}$ is used in other mode.
Note 1. Adjust the bit rate so that the widths at high and low level of the SCL output in simple IIC mode satisfy the IIC standard.
Note 2. In Manchester mode, only ABCSE = 0 and ABCSE2 = 0 can be selected.
Note 3. In Simple-LIN mode, BGDM = 0, ABCSE = 0 and ABCSE2 = 0 can be selected.
Note 4. Error means that if the N value calculated by the above formula is not an integer for the baud rate value B to be set, the BRR is set to an approximate integer value in which case it represents the error between the baud rate value and B.

Table 31.8 Calculating widths at high and low Level for SCL

Mode	SCLn	Formula (result in seconds)
IIC	Width at high level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{TCLK \times 10^6}$
	Width at low level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{TCLK \times 10^6}$

Table 31.9 Clock source settings

CCR2 setting		
CKS[1:0] bit	Clock source	n
00	TCLK clock	0
01	TCLK/4 clock	1
10	TCLK/16 clock	2
11	TCLK/64 clock	3

Table 31.10 Base clock settings in Smart card interface mode

CCR2 setting		
BCP[2:0] setting	Base clock cycles for 1-bit period	S
0 0 0	93 clock cycles	93
0 0 1	128 clock cycles	128
0 1 0	186 clock cycles	186
0 1 1	512 clock cycles	512
1 0 0	32 clock cycles	32
1 0 1	64 clock cycles	64
1 1 0	372 clock cycles	372
1 1 1	256 clock cycles	256

Table 31.11 and Table 31.12 list examples of N settings in BRR in Asynchronous mode and Manchester mode. Table 31.13 lists the maximum bit rate settable for each operating frequency. Examples of BRR (N) settings in Clock synchronous mode and Simple SPI mode are listed in Table 31.16. Examples of BRR (N) settings in Smart card interface mode are listed in Table 31.18. Examples of BRR (N) settings in Simple IIC mode are listed in Table 31.20. In Smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, see section 31.7.4. Receive Data Sampling Timing and Reception Margin. Table 31.14 and Table 31.17 list the maximum bit rates with external clock input. Table 31.15 lists the maximum bit rates with GPT clock input.

When either the Asynchronous mode base clock select bit (ABCS) or the baud rate generator double-speed mode select bit (BGDM) is set to 1 in Asynchronous mode and Manchester mode, the bit rate becomes twice that listed in Table 31.11 and Table 31.12. When both of those registers are set to 1, the bit rate becomes four times the listed value.

Table 31.11 Examples of BRR Settings for various Bit Rates (Asynchronous Mode and Manchester Mode) (1) (1 of 2)

Bit rate (bps)	Operating Frequency TCLK (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	103	0.16	2	127	0	2	129	0.16	2	155	0.16	2	159	0
300	1	207	0.16	1	255	0	2	64	0.16	2	77	0.16	2	79	0
600	1	103	0.16	1	127	0	1	129	0.16	1	155	0.16	1	159	0
1200	0	207	0.16	0	255	0	1	64	0.16	1	77	0.16	1	79	0
2400	0	103	0.16	0	127	0	0	129	0.16	0	155	0.16	0	159	0
4800	0	51	0.16	0	63	0	0	64	0.16	0	77	0.16	0	79	0
9600	0	25	0.16	0	31	0	0	32	–1.36	0	38	0.16	0	39	0
19200	0	12	0.16	0	15	0	0	15	1.73	0	19	–2.34	0	19	0
31250	0	7	0	0	9	–1.7	0	9	0	0	11	0	0	11	2.4
38400	—	—	—	0	7	0	0	7	1.73	0	9	–2.34	0	9	0

Table 31.11 Examples of BRR Settings for various Bit Rates (Asynchronous Mode and Manchester Mode) (1) (2 of 2)

Bit rate (bps)	Operating Frequency TCLK (MHz)														
	14			16			17.2032			18			19.6608		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	70	0.03	3	75	0.48	3	79	–0.12	3	86	0.31
150	2	181	0.16	2	207	0.16	2	223	0	2	233	0.16	2	255	0
300	2	90	0.16	2	103	0.16	2	111	0	2	116	0.16	2	127	0
600	1	181	0.16	1	207	0.16	1	223	0	1	233	0.16	1	255	0
1200	1	90	0.16	1	103	0.16	1	111	0	1	116	0.16	1	127	0
2400	0	181	0.16	0	207	0.16	0	223	0	0	233	0.16	0	255	0
4800	0	90	0.16	0	103	0.16	0	111	0	0	116	0.16	0	127	0
9600	0	45	–0.93	0	51	0.16	0	55	0	0	58	–0.69	0	63	0
19200	0	22	–0.93	0	25	0.16	0	27	0	0	28	1.02	0	31	0
31250	0	13	0	0	15	0	0	16	1.2	0	17	0	0	19	–1.7
38400	—	—	—	0	12	0.16	0	13	0	0	14	–2.34	0	15	0

Note: This is an example when the CCR2.ABCS = 0, CCR2.BGDM = 0 and CCR2.ABCSE = 0.
When either ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS = 1 and BGDM = 1, the bit rate increases four times.

Table 31.12 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode and Manchester Mode) (2) (1 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	20			25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	–0.25	3	110	–0.02	3	132	0.13	3	145	0.33	3	177	–0.25
150	3	64	0.16	3	80	0.47	3	97	–0.35	3	106	0.39	3	129	0.16
300	2	129	0.16	2	162	–0.15	2	194	0.16	2	214	–0.07	3	64	0.16
600	2	64	0.16	2	80	0.47	2	97	–0.35	2	106	0.39	2	129	0.16
1200	1	129	0.16	1	162	–0.15	1	194	0.16	1	214	–0.07	2	64	0.16

Table 31.12 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode and Manchester Mode) (2) (2 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	20			25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
2400	1	64	0.16	1	80	0.47	1	97	–0.35	1	106	0.39	1	129	0.16
4800	0	129	0.16	0	162	–0.15	0	194	0.16	0	214	–0.07	1	64	0.16
9600	0	64	0.16	0	80	0.47	0	97	–0.35	0	106	0.39	0	129	0.16
19200	0	32	–1.36	0	40	–0.76	0	48	–0.35	0	53	–0.54	0	64	0.16
31250	0	19	0	0	24	0	0	29	0	0	32	0	0	39	0
38400	0	15	1.73	0	19	1.73	0	23	1.73	0	26	–0.54	0	32	–1.36

Table 31.12 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode and Manchester Mode) (2) (3 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	50			60			100			120		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	221	–0.02	—	—	—	—	—	—	—	—	—
150	3	162	–0.15	3	194	0.16	—	—	—	—	—	—
300	3	80	0.47	3	97	–0.35	3	162	–0.15	3	194	0.16
600	2	162	–0.15	2	194	0.16	3	80	0.47	3	97	–0.35
1200	2	80	0.47	2	97	–0.35	2	162	–0.15	2	194	0.16
2400	1	162	–0.15	1	194	0.16	2	80	0.47	2	97	–0.35
4800	1	80	0.47	1	97	–0.35	1	162	–0.15	1	194	0.16
9600	0	162	–0.15	0	194	0.16	1	80	0.47	1	97	–0.35
19200	0	80	0.47	0	97	–0.35	0	162	–0.15	0	194	0.16
31250	0	49	0	0	59	0	1	24	0	0	119	0
38400	0	40	–0.76	0	48	–0.35	0	80	0.47	0	97	–0.35

Note: This is an example when the CCR2.ABCS = 0, CCR2.BGDM = 0, CCR2.ABCSE = 0 and CCR2.ABCSE2 = 0.
When either ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS = 1 and BGDM = 1, the bit rate increases four times.

Table 31.13 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode and Manchester mode) (1 of 3)

TCLK (MHz)	CCR2 settings				Maximum bit rate (bps)	TCLK (MHz)	CCR2 settings				Maximum bit rate (bps)
	BGDM	ABCS	ABCSE	ABCSE2			BGDM	ABCS	ABCSE	ABCSE2	
8	0	0	0	0	250000	16	0	0	0	0	500000
		1	0	0	500000			1	0	0	1000000
	1	0	0	0			1	0	0	0	
		1	0	0	1000000			1	0	0	2000000
	Don't care	Don't care	1	0	1333333		Don't care	Don't care	1	0	2666666
	Don't care	Don't care	0	1	2000000		Don't care	Don't care	0	1	4000000

Table 31.13 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode and Manchester mode) (2 of 3)

TCLK (MHz)	CCR2 settings				Maximum bit rate (bps)	TCLK (MHz)	CCR2 settings				Maximum bit rate (bps)
	BGDM	ABCS	ABCSE	ABCSE2			BGDM	ABCS	ABCSE	ABCSE2	
9.8304	0	0	0	0	307200	17.2032	0	0	0	0	537600
		1	0	0	614400			1	0	0	1075200
	1	0	0	0			1	0	0	0	
		1	0	0	1228800			1	0	0	2150400
	Don't care	Don't care	1	0	1638400		Don't care	Don't care	1	0	2867200
	Don't care	Don't care	0	1	2457600		Don't care	Don't care	0	1	4300800
10	0	0	0	0	312500	18	0	0	0	0	562500
		1	0	0	625000			1	0	0	1125000
	1	0	0	0			1	0	0	0	
		1	0	0	1250000			1	0	0	2250000
	Don't care	Don't care	1	0	1666666		Don't care	Don't care	1	0	3000000
	Don't care	Don't care	0	1	2500000		Don't care	Don't care	0	1	4500000
12	0	0	0	0	375000	19.6608	0	0	0	0	614400
		1	0	0	750000			1	0	0	1228800
	1	0	0	0			1	0	0	0	
		1	0	0	1500000			1	0	0	2457600
	Don't care	Don't care	1	0	2000000		Don't care	Don't care	1	0	3276800
	Don't care	Don't care	0	1	3000000		Don't care	Don't care	0	1	4915200
12.288	0	0	0	0	384000	20	0	0	0	0	625000
		1	0	0	768000			1	0	0	1250000
	1	0	0	0			1	0	0	0	
		1	0	0	1536000			1	0	0	2500000
	Don't care	Don't care	1	0	2048000		Don't care	Don't care	1	0	3333333
	Don't care	Don't care	0	1	3072000		Don't care	Don't care	0	1	5000000
14	0	0	0	0	437500	25	0	0	0	0	781250
		1	0	0	875000			1	0	0	1562500
	1	0	0	0			1	0	0	0	
		1	0	0	1750000			1	0	0	3125000
	Don't care	Don't care	1	0	2333333		Don't care	Don't care	1	0	4166666
	Don't care	Don't care	0	1	3500000		Don't care	Don't care	0	1	6250000

Table 31.13 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode and Manchester mode) (3 of 3)

TCLK (MHz)	CCR2 settings				Maximum bit rate (bps)	TCLK (MHz)	CCR2 settings				Maximum bit rate (bps)
	BGDM	ABCS	ABCSE	ABCSE2			BGDM	ABCS	ABCSE	ABCSE2	
30	0	0	0	0	937500	50	0	0	0	0	1562500
		1	0	0	1875000			1	0	0	3125000
	1	0	0	0			1	0	0	0	
		1	0	0	3750000			1	0	0	6250000
	Don't care	Don't care	1	0	5000000		Don't care	Don't care	1	0	8333333
	Don't care	Don't care	0	1	7500000		Don't care	Don't care	0	1	12500000
33	0	0	0	0	1031250	60	0	0	0	0	1875000
		1	0	0	2062500			1	0	0	3750000
	1	0	0	0			1	0	0	0	
		1	0	0	4125000			1	0	0	7500000
	Don't care	Don't care	1	0	5500000		Don't care	Don't care	1	0	10000000
	Don't care	Don't care	0	1	8250000		Don't care	Don't care	0	1	15000000
40	0	0	0	0	1250000	120	0	0	0	0	3750000
		1	0	0	2500000			1	0	0	7500000
	1	0	0	0			1	0	0	0	
		1	0	0	5000000			1	0	0	15000000
	Don't care	Don't care	1	0	6666666		Don't care	Don't care	1	0	20000000
	Don't care	Don't care	0	1	10000000		Don't care	Don't care	0	1	30000000

Note: For maximum bit rate, the CKS and BRR register values should be n = 0 and N = 0, respectively. In Manchester mode, use ABCSE = ABCSE2 = 0.

Table 31.14 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

TCLK(MHz)	External clock (MHz)	Maximum bit rate (bps)		TCLK(MHz)	External clock (MHz)	Maximum bit rate (bps)	
		CCR2.ABCS bit = 0	CCR2.ABCS bit = 1			CCR2.ABCS bit = 0	CCR2.ABCS bit = 1
8	2	125000	250000	25	6.25	390625	781250
9.8304	2.4576	153600	307200	30	7.5	468750	937500
10	2.5	156250	312500	33	8.25	515625	1031250
12	3	187500	375000	40	10	625000	1250000
12.288	3.072	192000	384000	50	12.5	781250	1562500
14	3.5	218750	437500	60	15	937500	1875000
16	4	250000	500000	120	30	1875000	3750000
17.2032	4.3008	268800	537600				
18	4.5	281250	562500				
19.6608	4.9152	307200	614400				
20	5	312500	625000				

Table 31.15 Maximum Bit Rate with GPT Clock Input (Asynchronous Mode)

TCLK (MHz)	GPT clock (MHz)	Maximum bit rate (bps)		TCLK (MHz)	GPT clock (MHz)	Maximum bit rate (bps)	
		CCR2.ABCS bit = 0	CCR2.ABCS bit = 1			CCR2.ABCS bit = 0	CCR2.ABCS bit = 1
8	4	250000	500000	25	12.5	781250	1562500
9.8304	4.9152	307200	614400	30	15	937500	1875000
10	5.0	312500	625000	33	16.5	1031250	2062500
12	6	375000	750000	40	20	1250000	2500000
12.288	6.144	384000	768000	50	25	1562500	3125000
14	7.0	437500	875000	60	30	1875000	3750000
16	8	500000	1000000	120	60	3750000	7500000
17.2032	8.6016	537600	1075200				
18	9.0	562500	1125000				
19.6608	9.8304	614400	1228800				
20	10	625000	1250000				

Table 31.16 BRR Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	8			10			30			60			120		
	BGDM	n	N	BGDM	n	N	BGDM	n	N	BGDM	n	N	BGDM	n	N
250	0	3	124	0	3	155	—	—	—	—	—	—	—	—	—
500	0	2	249	0	3	77	0	3	233	—	—	—	—	—	—
1k	0	2	124	0	3	38	0	3	116	0	3	233	—	—	—
2.5k	0	2	49	0	1	249	0	3	46	0	3	93	0	3	187
5k	0	2	24	0	1	124	0	2	93	0	3	46	0	3	93
10k	0	1	49	0	0	249	0	2	46	0	2	93	0	3	46
25k	0	2	4	0	1	24	0	1	74	0	1	149	0	2	74
50k	0	1	9	0	0	49	0	0	149	0	1	74	0	1	149
100k	0	1	4	0	0	24	0	0	74	0	0	149	0	1	74
250k	0	1	1	0	0	9	0	0	29	0	1	14	0	1	29
500k	0	1	0	0	0	4	0	0	14	0	0	29	0	1	14
1M	0	0	1	1	0	4	1	0	14	0	0	14	0	0	29
2.5M	—	—	—	0	0	0	0	0	2	0	0	5	0	1	2
5M	—	—	—	1	0	0	1	0	2	0	0	2	0	0	5
7.5M	—	—	—	—	—	—	0	0	0	0	0	1	0	1	0
60M	—	—	—	—	—	—	—	—	—	—	—	—	1	0	0

Note: —: Can be set, but an error over 10% will occur.

Table 31.17 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode) (1 of 2)

TCLK (MHz)	External clock (MHz)	MAX Bit rate (Mbps)	TCLK (MHz)	External clock (MHz)	MAX Bit rate (Mbps)
8	4	4	25	12.5	12.5
10	5	5	30	15	15
12	6	6	33	16.5	16.5
14	7	7	40	20	20

Table 31.17 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode) (2 of 2)

TCLK (MHz)	External clock (MHz)	MAX Bit rate (Mbps)	TCLK (MHz)	External clock (MHz)	MAX Bit rate (Mbps)
16	8	8	50	25	25
18	9	9	60	30	30
20	10	10	120	60	60

Table 31.18 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372) (1 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	7.1424			10.00			10.7136			13.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	0	0.00	0	1	–30	0	1	–25	0	1	–8.99

Table 31.18 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372) (2 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	14.2848			16.00			18.00			20.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	1	0.00	0	1	12.01	0	2	–15.99	0	2	–6.66

Table 31.18 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372) (3 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	25.00			30.00			33.00			40.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	3	–12.49	0	3	5.01	0	4	–7.59	0	5	–6.66

Table 31.18 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372) (4 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)								
	50.00			60.00			120.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	6	0.01	0	7	5.01	0	16	–1.17

Table 31.19 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 32)

TCLK (MHz)	MAX Bit rate (bps)	n	N	TCLK (MHz)	MAX Bit rate (bps)	n	N
10	156250	0	0	30	468750	0	0
10.7136	167400	0	0	33	515625	0	0
13	203125	0	0	40	625000	0	0
16	250000	0	0	50	781250	0	0
18	281250	0	0	60	937500	0	0
20	312500	0	0	120	1875000	0	0
25	390625	0	0				

Table 31.20 BRR Settings for Various Bit Rates (Simple IIC Mode) (1 of 3)

Bit rate (bps)	Operating Frequency TCLK (MHz)											
	8			10			16			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10k	0	24	0	0	30	0.8	0	49	0	0	62	-0.8
25k	0	9	0	0	12	-3.8	1	4	0	0	24	0
50k	0	4	0	0	5	4.2	0	9	0	0	12	-3.8
100k	0	2	-16.7	0	2	4.2	0	4	0	0	5	4.2
250k	0	0	0	0	0	25	0	1	0	0	2	-16.7
350k										0	1	-10.7
400k										0	1	-21.9

Table 31.20 BRR Settings for Various Bit Rates (Simple IIC Mode) (2 of 3)

Bit rate (bps)	Operating Frequency TCLK (MHz)											
	25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10k	0	77	0.2	0	93	-0.3	0	102	0.1	0	124	0
25k	0	30	0.8	0	37	-1.3	0	40	0.6	0	49	0
50k	2	0	-2.3	0	18	-1.3	0	20	-1.8	0	24	0
100k	1	1	-2.3	0	8	4.2	0	9	3.1	0	12	-3.8
250k	0	2	4.2	1	0	-6.3	1	0	3.1	0	4	0
350k	0	2	-25.6	0	2	-10.7	0	2	-1.8	1	0	-10.7
400k	0	2	-34.9	0	2	-21.9	0	2	-14.1	1	0	-21.9

Table 31.20 BRR Settings for Various Bit Rates (Simple IIC Mode) (3 of 3)

Bit rate (bps)	Operating Frequency TCLK (MHz)								
	50			60			120		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10k	1	38	-0.2	1	46	-0.3	1	93	-0.3
25k	0	62	-0.8	0	74	0	0	149	0
50k	0	30	0.8	0	37	-1.3	0	74	0
100k	2	0	-2.3	0	18	-1.3	0	37	-1.3
250k	0	5	4.2	1	1	-6.3	0	14	0
350k	0	4	-10.7	0	4	7.1	0	10	-2.6
400k	0	4	-21.9	0	4	-6.3	0	9	-6.3

Table 31.21 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple IIC Mode) (1 of 4)

Bit rate (bps)	Operating Frequency TCLK (MHz)											
	8			10			16			20		
	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)
10k	0	24	43.75 / 50.00	0	30	43.40 / 49.60	0	49	43.75 / 50.00	0	62	44.10 / 50.40
25k	0	9	17.50 / 20.00	0	12	18.20 / 20.80	1	4	17.50 / 20.00	0	24	17.50 / 20.00

Table 31.21 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple IIC Mode) (2 of 4)

Bit rate (bps)	Operating Frequency TCLK (MHz)											
	8			10			16			20		
	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)
50k	0	4	8.75 / 10.00	0	5	8.40 / 9.60	0	9	8.75 / 10.00	0	12	9.10 / 10.40
100k	0	2	5.25 / 6.00	0	2	4.20 / 4.80	0	4	4.37 / 5.00	0	5	4.20 / 4.80
250k	0	0	1.75 / 2.00	0	1	1.40 / 1.60	0	1	1.75 / 2.00	0	2	2.10 / 2.40
350k										0	1	1.40 / 1.60
400k										0	1	1.40 / 1.60

Table 31.21 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple IIC Mode) (3 of 4)

Bit rate (bps)	Operating Frequency TCLK (MHz)											
	25			30			33			40		
	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)
10k	0	77	43.68 / 49.92	0	93	43.87 / 50.13	0	102	43.70 / 49.94	0	124	43.75 / 50.00
25k	0	30	17.36 / 19.84	0	37	17.73 / 20.27	0	40	17.39 / 19.88	0	49	17.50 / 20.00
50k	2	0	8.96 / 10.24	1	4	8.87 / 10.13	0	20	8.91 / 10.18	0	24	8.75 / 10.00
100k	1	1	4.48 / 5.12	0	8	4.20 / 4.80	0	9	4.24 / 4.85	0	12	4.55 / 5.20
250k	0	2	1.68 / 1.92	1	0	1.87 / 2.13	1	0	1.70 / 1.94	0	4	1.75 / 2.00
350k	0	2	1.68 / 1.92	0	2	1.40 / 1.60	0	2	1.27 / 1.45	1	0	1.40 / 1.60
400k	0	2	1.68 / 1.92	0	2	1.40 / 1.60	0	2	1.27 / 1.45	1	0	1.40 / 1.60

Table 31.21 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple IIC Mode) (4 of 4)

Bit rate (bps)	Operating Frequency TCLK (MHz)								
	50			60			120		
	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)
10k	1	38	43.68 / 49.92	1	46	43.87 / 50.03	1	93	43.87 / 50.13
25k	0	62	17.64 / 20.16	0	74	17.50 / 20.00	0	149	17.50 / 20.00
50k	0	30	8.68 / 9.92	0	37	8.87 / 10.13	0	74	8.75 / 10.00
100k	2	0	4.48 / 5.12	0	18	4.43 / 5.07	0	37	4.43 / 5.07
250k	0	5	1.68 / 1.92	1	1	1.87 / 2.13	0	14	1.75 / 2.00
350k	0	4	1.40 / 1.60	0	4	1.17 / 1.33	0	10	1.28 / 1.47
400k	0	4	1.40 / 1.60	0	4	1.17 / 1.33	0	9	1.17 / 1.33

BRME bit (Bit Rate Modulation Enable)

Enables and disables the bit rate modulation function. The bit rate generated by on-chip baud rate generator is evenly corrected when this function is enabled.

Set to 0 in Clock-synchronous mode, Simple SPI mode, Smart Card Interface mode, Manchester mode and Simple LIN mode.

Even in asynchronous mode, set this bit to 0 when ABCSE2 = 1.

CKS[1:0] bits (Clock Select)

The CKS[1:0] bits select the clock source for the on-chip baud rate generator.

For the relationship between the settings of the CKS[1:0] bits and the baud rate, see BRR[7:0] bits explanation.

MDDR[7:0] bits (Modulation Duty Setting)

When the BRME bit is set to 1, the bit rate generated by the on-chip baud rate generator is evenly corrected according to the settings of MDDR (M/256). The relationship between the MDDR setting (M) and the bit rate (B) is given in Table 31.22.

The initial value of MDDR is FFh. Bit 7 in the CCR2 is fixed to 1.

Table 31.22 Relationship between MDDR Setting (M) and Bit Rate (B) When Bit Rate Modulation Function is Used

Mode*1	CCR2 settings			BRR setting	Error
	BGDM bit	ABCS bit	ABCSE bit		
Asynchronous, Multiprocessor transfer	0	0	0	$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	$Error\ (\%) = \left\{\frac{TCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\right\} \times 100$
	1	0	0	$N = \frac{TCLK \times 10^6}{32 \times 2^{2n-1} \times (256/M) \times B} - 1$	$Error\ (\%) = \left\{\frac{TCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\right\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{TCLK \times 10^6}{16 \times 2^{2n-1} \times (256/M) \times B} - 1$	$Error\ (\%) = \left\{\frac{TCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\right\} \times 100$
	x (Arbitrarily)	x (Arbitrarily)	1*2	$N = \frac{TCLK \times 10^6}{12 \times 2^{2n-1} \times (256/M) \times B} - 1$	$Error\ (\%) = \left\{\frac{TCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\right\} \times 100$
Simple IIC*2	$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$				

Note: B: Bit rate (bps)
M: MDDR setting (128 ≤ M ≤ 255)
N: BRR setting for bound rate generator (0 ≤ N ≤ 255)
TCLK: Operating frequency (MHz)
n: Determined by the settings of the CKS[1:0] as listed in Table 31.9.

Note 1. Do not use this function in Clock-synchronous mode, Simple SPI mode, Smart card Interface mode, Manchester mode and Simple LIN mode.

Note 2. Adjust the bit rate so that the widths at high and low level of the SCL output in simple IIC mode satisfy the I2C standard.

Table 31.23 and Table 31.24 list examples of N settings in BRR and M settings in MDDR in Asynchronous mode.

Table 31.23 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (1) (1 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	8					9.8304					10				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	5	236	0	0.03	0	7	(256) *1	0	0	0	10	173	1	–0.01
57600	0	3	236	0	0.03	0	4	240	0	0	0	4	236	0	0.03
115200	0	1	236	0	0.03	0	1	192	0	0	0	4	236	1	0.03
230400	0	0	236	0	0.03	0	0	192	0	0	0	1	189	1	0.14
460800	0	0	236	1	0.03	0	0	192	1	0	0	0	189	1	0.14

Table 31.23 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (1) (2 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	12					12.288					14				
	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)
38400	0	8	236	0	0.03	0	9	(256) *1	0	0	0	16	191	1	0
57600	0	5	236	0	0.03	0	4	192	0	0	0	13	236	1	0.03
115200	0	2	236	0	0.03	0	4	192	1	0	0	6	236	1	0.03
230400	0	2	236	1	0.03	0	2	230	1	–0.17	0	2	202	1	–0.11
460800	0	0	157	1	–0.18	0	0	154	1	0.26	0	0	135	1	0.14

Table 31.23 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (1) (3 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	16					17.2032					18				
	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)
38400	0	11	236	0	0.03	0	13	(256) *1	0	0	0	18	166	1	–0.01
57600	0	7	236	0	0.03	0	6	192	0	0	0	18	249	1	–0.01
115200	0	3	236	0	0.03	0	6	192	1	0	0	8	236	1	0.03
230400	0	1	236	0	0.03	0	3	219	1	–0.2	0	1	210	0	0.14
460800	0	1	236	1	0.03	0	1	219	1	–0.2	0	0	210	0	0.14

Note: This is an example when the CCR2.ABCS = 0 and CCR2.ABCSE = 0.
Note 1. It means that the bit rate modulation function is disable. (CCR2.BRME = 0, M = 256)

Table 31.24 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (2) (1 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	19.6608					20					25				
	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)
38400	0	15	(256) *1	0	0	0	10	173	0	–0.01	0	11	151	0	0
57600	0	9	240	0	0	0	9	236	0	0.03	0	7	151	0	0
115200	0	4	240	0	0	0	4	236	0	0.03	0	3	151	0	0
230400	0	1	192	0	0	0	4	236	1	0.03	0	1	151	0	0
460800	0	0	192	0	0	0	0	189	0	0.14	0	0	151	0	0

Table 31.24 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (2) (2 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	30					33					40				
	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)
38400	0	36	194	1	0.01	0	14	143	0	0.01	0	21	173	0	–0.01
57600	0	10	173	0	–0.01	0	9	143	0	0.01	0	38	230	1	–0.01
115200	0	10	173	1	–0.01	0	4	143	0	0.01	0	9	236	0	0.03
230400	0	6	220	1	–0.09	0	4	143	1	0.01	0	4	236	0	0.03
460800	0	3	252	1	0.14	0	1	229	0	0.1	0	4	236	1	0.03

Table 31.24 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (2) (3 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)													
	50					60					120				
	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)
38400	0	23	151	0	0	0	36	194	0	0.01	0	73	194	0	0.01
57600	0	15	151	0	0	0	21	173	0	–0.01	0	58	232	0	0
115200	0	7	151	0	0	0	10	173	0	–0.01	0	21	173	0	–0.01
230400	0	3	151	0	0	0	10	173	1	–0.01	0	10	173	0	–0.01
460800	0	1	151	0	0	0	6	220	1	–0.09	0	10	173	1	–0.01

Note: This is an example when the CCR2.ABCS = 0 and CCR2.ABCSE = 0.
Note 1. It means that the bit rate modulation function is disable. (CCR2.BRME = 0, M = 256)

31.2.8 CCR3 : Common Control Register 3

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 4, 9)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	BLK	GM	—	ACS0	CKE[1:0]		—	—	DEN	FM	MP	MOD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RXDESEL	STP	SINV	LSBF	—	—	CHR[1:0]		BPEN	—	—	—	—	—	CPOL	CPHA
Value after reset:	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	CPHA	Clock Phase Select Valid in Clock synchronous mode and Simple SPI mode. Set this bit only when CCR0.TE = 0 and RE = 0. 0: Data is sampled at an odd edge and changes at an even edge (clock is delayed) 1: Data changes at an odd edge and is sampled at an even edge (clock is not delayed)	R/W
1	CPOL	Clock Polarity Select Valid in Clock synchronous mode and Simple SPI mode. Set this bit only when CCR0.TE = 0 and RE = 0. 0: SCKn in idle state is 0 1: SCKn in idle state is 1	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	BPEN	Synchronizer Bypass Enable This bit controls whether to bypass the synchronizer circuit between the bus clock and operation clock. 0: Synchronizer circuit is not bypassed 1: Synchronizer circuit is bypassed	R/W
9:8	CHR[1:0]	Character Length Valid in Asynchronous mode and Manchester mode[*1] Select the data length for transmission and reception. 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length[*2]	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
12	LSBF	LSB First select Set this bit to 0 in Simple IIC mode. Set this bit to 1 in Simple LIN mode. 0: MSB-first 1: LSB-first	R/W
13	SINV	Transmitted/Received Data Invert Set this bit to 0 in Simple IIC mode. The level of communication pins (RXDn/TXDn) are controlled by combination of this bit and CCR1.TINV/RINV. For details, see Figure 31.3. 0: TDR contents are transmitted to TSR as is. RSR contents are stored to RDR as is. 1: TDR contents are inverted before being transmitted to TSR. RSR contents are inverted and stored to RDR.	R/W
14	STP	Stop Bit Length Valid in Asynchronous mode, Manchester mode, and Simple LIN mode 0: 1 stop bit/break delimiter length is 1-bit 1: 2 stop bits/break delimiter length is 2-bit	R/W
15	RXDESEL	Asynchronous Start Bit Edge Detection Select Valid only in Asynchronous mode Set this bit to 1 in Simple LIN mode. 0: The low level on the RXDn pin is detected as the start bit 1: A falling edge on the RXDn pin is detected as the start bit	R/W
18:16	MOD[2:0]	Communication Mode Select Select the SCI communication mode. 0 0 0: Asynchronous mode (multi-processor mode) 0 0 1: Smart card interface mode 0 1 0: Clock synchronous mode 0 1 1: Simple SPI mode 1 0 0: Simple IIC mode 1 0 1: Manchester mode 1 1 0: Simple LIN mode 1 1 1: Setting prohibited	R/W
19	MP	Multi-Processor Mode Valid in Asynchronous mode and Manchester mode 0: Multi-processor communications function is disabled 1: Multi-processor communications function is enabled	R/W
20	FM	FIFO Mode Select Valid in Asynchronous mode (including Multi-processor mode), Clock synchronous mode, and Simple SPI mode 0: TDR register and RDR register are non-FIFO buffer configuration 1: TDR register and RDR register are FIFO buffer configuration	R/W
21	DEN	Driver Enable 0: RS-485 driver control function disabled 1: RS-485 driver control function enabled	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
25:24	CKE[1:0]	Clock Enable See Table 31.26 for details.	R/W
26	ACS0	Asynchronous Mode Clock Source Select Valid only in Asynchronous mode. These bits for SCI channels other than SCIn (n = 1, 2) are reserved. 0: External clock input 1: Logical AND of compare matches output from the internal GPT	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
28	GM	GSM Mode Valid only in Smart card interface mode. 0: Non-GSM mode operation 1: GSM mode operation	R/W

Bit	Symbol	Function	R/W
29	BLK	Block Transfer Mode Valid only in Smart card interface mode. 0: Non-block transfer mode operation 1: Block transfer mode operation	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. In other than Asynchronous mode and Manchester mode, this bit setting is invalid and a fixed data length of 8 bits is used. In the Simple LIN mode, only the data length of 8 bits can be used, so set it to the initial value.
Note 2. LSB first is fixed and the MSB (bit 7) in TDR is not transmitted in transmission.
Note 3. GPT clock is selectable for SCI1 and SCI2.

CPHA bit (Clock Phase Select)

The CPHA bit selects the phase of the clock signal output through the SCKn pin. For details, see Figure 31.99.

Set the bit to 1 in other than Simple SPI mode and Clock synchronous mode.

CPOL bit (Clock Polarity Select)

The CPOL bit selects the polarity of the clock signal output through the SCKn pin. For details, see Figure 31.99.

Set the bit to 1 in other than Simple SPI mode and Clock synchronous mode.

BPEN bit (Synchronizer Bypass Enable)

The BPEN bit selects whether to enable or disable the synchronization bypass function. Set this bit to 1 when the bus clock (PCLK) is also used as operation clock (TCLK). For details, see Figure 31.2 and section 31.18. Synchronizer Bypass Function.

Note: For details of this bit setting, see section 31.19.17. Notes on CCR3.BPEN bit setting.

CHR[1:0] bits (Character Length)

Selects the data length for transmission and reception.

Except for Asynchronous mode and Manchester mode, a fixed data length of 8 bits is used.

LSBF bit (LSB First select)

Select whether to transmit/receive data in MSB-first or LSB-first.

SINV bit (Transmitted/Received Data Invert)

SINV can invert the transmit data bit from TDR to TSR, and can invert the received data from RSR to RDR. The SINV bit does not affect the logic level of the parity bit. To invert the parity bit, invert the CCR1.PM.

STP bit (Stop Bit Length)

Selects the stop bit length in transmission.

In reception, only the first stop bit is checked regardless of the STP bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

In addition, it is used as the break delimiter length setting when sending start frame in Simple LIN mode.

RXDESEL bit (Asynchronous Start Bit Edge Detection Select)

Selects the detection method of the start bit for reception in Asynchronous mode. When a break occurs, data receiving operation depends on the settings of the RXDESEL bit. Set this bit to 1 when reception should be stopped while a break occurs or when reception should be started without retaining the RXDn pin input at high level for the period of one data frame or longer after completion of the break.

Set this bit to 1 in Simple LIN mode. Set this bit to 0 in modes except of Asynchronous mode and Simple LIN mode.

MOD[2:0] bits (Communication Mode Select)

Selects the SCI communication mode.

Table 31.25 Relationship between communication mode selection bits (MOD[2:0]) and other operation mode setting bits

Communication mode	Asynchronous	Smart card interface	Clock synchronous	Simple SPI	Simple IIC	Manchester	Simple LIN
CCR3.MOD[2:0]	000b	001b	010b	011b	100b	101b	110b
CCR3.MP	0 1	—	—	—	—	0 1	—
CCR3.FM	0 1 0 1	—	0 1	0 1	—	—	—
CCR3.DEN	0 1 0 1 0 1 0 1	—	—	—	—	—	—
CCR0.SSE	—	—		0 1 0 1	—	—	—

Note: — is prohibited setting.

MP bit (Multi-Processor Mode)

Disables/enables multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

FM bit (FIFO Mode Select)

When the FM bit is set to 1, the TDR register / RDR register switches to FIFO configuration, and transmit FIFO (TDR register) / receive FIFO (RDR register) can be used for serial transmission/reception.

DEN bit (Driver Enable)

Select RS-485 driver control function disabled or enabled.

CKE[1:0] bits (Clock Enable)

Table 31.26 Association between Clock Enable bits CKE[1:0] and operation mode (1 of 2)

CKE[1:0]	Function
00b	In Asynchronous mode: On-chip baud rate generator The SCKn pin is available for use as an I/O port in accord with the I/O port settings. In Manchester mode and Simple LIN mode: On-chip baud rate generator The SCKn pin functions as I/O port. In Clock synchronous mode and Simple SPI mode: Internal clock (master operation) The SCKn pin functions as the clock output pin. In Smart card interface mode and when CCR3.GM = 0: Output disabled (the SCKn pin is available for use as an I/O port in accord with the I/O port settings). In Smart card interface mode and when CCR3.GM = 1: Output fixed low
01b	In Asynchronous mode: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. In Manchester mode and Simple LIN mode: Prohibited In Clock synchronous mode and Simple SPI mode: Internal clock (master operation) The SCKn pin functions as the clock output pin. In Smart card interface mode and when CCR3.GM = 0: Clock output In Smart card interface mode and when CCR3.GM = 1: Clock output

Table 31.26 Association between Clock Enable bits CKE[1:0] and operation mode (2 of 2)

CKE[1:0]	Function
10b	In Asynchronous mode: External clock or GPT clock • When using the external clock 16 times the bit rate should be input from the SCKn pin when CCR2.ABCS bit is 0. Input a clock signal with a frequency 8 times the bit rate when the CCR2.ABCS bit is 1. • When using the GPT clock*3 The SCKn pin is available for use as an I/O port in accord with the I/O port settings when the GPT clock is used. In Manchester mode and Simple LIN mode: Prohibited In Clock synchronous mode and Simple SPI mode: External clock (slave operation) The SCKn pin functions as the clock input pin. In Smart card interface mode and when CCR3.GM = 0: Prohibited In Smart card interface mode and when CCR3.GM = 1: Output fixed high
11b	In Asynchronous mode: External clock or GPT clock • When using the external clock 16 times the bit rate should be input from the SCKn pin when CCR2.ABCS bit is 0. Input a clock signal with a frequency 8 times the bit rate when the CCR2.ABCS bit is 1. • When using the GPT clock *3 The SCKn pin is available for use as an I/O port in accord with the I/O port settings when the GPT clock is used. In Manchester mode and Simple LIN mode: Prohibited In Clock synchronous mode and Simple SPI mode: External clock (slave operation) The SCKn pin functions as the clock input pin. In Smart card interface mode and when CCR3.GM = 0: Prohibited In Smart card interface mode and when CCR3.GM = 1: Clock output

The CKE[1:0] bits select the clock source and SCKn pin function.

The combination of these bit settings and of the CCR3.ACS0 bit sets the internal GPT clock.

In Smart card interface mode, these bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, see section 31.7.8. Clock Output Control.

GM bit (GSM Mode)

Setting the GM bit to 1 allows GSM mode operation.

In GSM mode, the CSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, see section 31.7.6. Serial Data Transmission (Except in Block Transfer Mode), section 31.7.8. Clock Output Control.

ACS0 bit (Asynchronous Mode Clock Source Select)

Selects the clock source in the Asynchronous mode.

The ACS0 bit is valid in Asynchronous mode (CCR3.MOD[2:0] = 000b) and when an external clock input is selected (CCR3.CKE[1:0] = 10b or 11b). This bit is used to select an external clock input or the logical AND of compare matches output from the internal GPT.

Set the ACS0 bit to 0 in other than Asynchronous mode.

For SCI1 and SCI2, the GTIOCmA output (m = 3, 4, 6, 7) of GPT can be set as the serial transfer base clock. See Table 31.27 for details.

These bits for SCI channels other than SCI1 and SCI2 are reserved. The write values to these bits for other than SCI1 and SCI2 should be 0.

Table 31.27 Association between SCI channels and compare match outputs

SCI	GPT	Compare match output
SCI1	GPT3	GTIOC3A
	GPT4	GTIOC4A
SCI2	GPT6	GTIOC6A
	GPT7	GTIOC7A

Figure 31.4, Figure 31.5 show a setting example of when GTIOC3A and GTIOC4A are selected for output. This figure shows an example when GPT clock is input to SCI1.

When generating 187.5 kbps of GPT average transfer rate for PCLKD = 32 MHz:

1. Generate a frequency of 4 MHz using GTIOC3A as the base clock.
2. Generate 3/4 clock enable of base clock to set an average transfer rate of 3 MHz/16 = 187.5 kbps using GTIOC4A.

Setting examples of GPT and SCI

- GPT3.GTSSR = 0x80000000, GPT4.GTSSR = 0x80000000 (enable software counter start)
- GPT3.GTPR = 0x00000007, GPT4.GTPR = 0x0000001F (maximum count value of GTCNT)
- GPT3.GTCCRA = 0x00000003, GPT4.GTCCRA = 0x00000007 (compare match value)
- GPT3.GTCR = 0x00000000, GPT4.GTCR = 0x00000000 (Saw-wave PWM mode, timer prescaler is PCLKD)
- GPT3.GTIOR = 0x00000306, GPT4.GTIOR = 0x00000306
  (initial output low, output high at GTCCRA compare match, output low at cycle end)
- CCR3.CKE[1:0] = 10b (external clock input or GPT clock input is selected)
- CCR3.ACS0 = 1, CCR2.ABCSE = 1, CCR2.ABCS=0 (GPT clock input is selected with 16 base clock cycles for 1-bit period)
- GPT3.GTSTR = 0x00000018 (software start GTCNT counter)

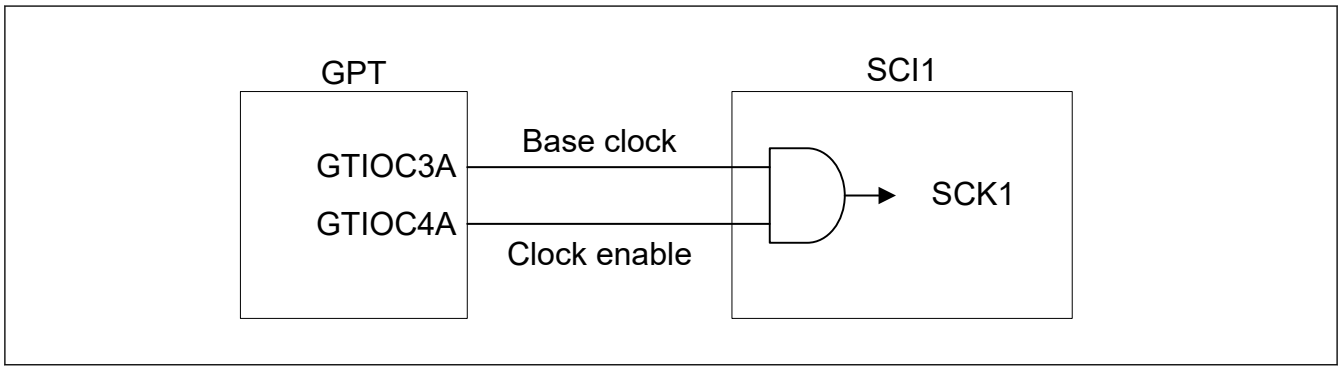


Figure 31.4 Example of GPT and SCI connection

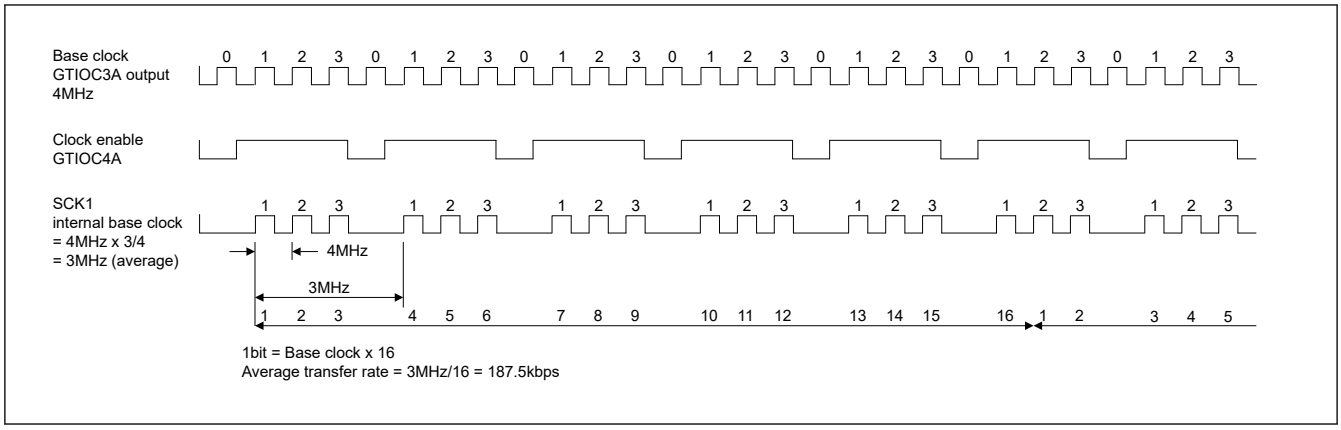


Figure 31.5 Example of average transfer rate setting when GPT clock is input

BLK bit (Block Transfer Mode)

Setting the BLK bit to 1 allows block transfer mode operation.

For details, see section 31.7.3. Block Transfer Mode.

31.2.9 CCR4 : Common Control Register 4

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 4, 9)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	AET	ATT[2:0]			AJD	AST[2:0]			—	—	—	—	SCKSEL	—	ATEN	ASEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CMPD[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	CMPD[8:0]	Compare Match Data Valid only in Asynchronous mode.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	ASEN	Adjust Receive Sampling Timing Enable Valid in Asynchronous mode using internal clock, Simple LIN mode using internal clock, Clock-synchronous mode operating as master, and Simple SPI mode operating as master. 0: Adjust sampling timing disabled 1: Adjust sampling timing enabled	R/W
17	ATEN	Adjust Transmit Timing Enable Valid only in Asynchronous mode using internal clock. 0: Adjust transmit timing disabled 1: Adjust transmit timing enabled	R/W
18	—	This bit is read as 0. The write value should be 0.	R/W
19	SCKSEL	Master receive clock selection Valid only in Clock synchronous and Simple-SPI mode. 0: Master receive clock is disabled 1: Master receive clock is enabled	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
26:24	AST[2:0]	Adjustment Value for Receive Sampling Timing This bit enables only when ASEN is 1. In Asynchronous mode and Simple LIN mode using internal clock: The sampling timing of RXDn pin is adjusted from the middle of bit by the following formula. Adjustment sampling timing = base clock × the setting value of AST[2:0]. In Clock synchronous mode and Simple SPI mode using internal clock: The RXDn sampling timing can be adjusted by delaying 1 to 4 TCLK 0 0 0: 1-TCLK delay 0 0 1: 2-TCLK delay 0 1 0: 3-TCLK delay 0 1 1: 4-TCLK delay Others: Setting prohibited	R/W
27	AJD	Adjustment Direction for Receive Sampling Timing Valid in Asynchronous mode using internal clock and Simple LIN mode using internal clock. This bit enables only when ASEN is 1. Adjustment direction for RXDn receive sampling timing is determined by this bit. For details, see section 31.3.10. The function of adjust receive sampling timing (Asynchronous Mode). 0: The sampling timing is adjusted backward to the middle of bit. 1: The sampling timing is adjusted forward to the middle of bit.	R/W
30:28	ATT[2:0]	Adjustment Value for Transmit Timing Valid in Asynchronous mode using internal clock and Simple LIN mode using internal clock. This bit enables only when ATEN is 1. The selected edge timing of TXDn is adjusted by the following formula. Adjustment edge timing = base clock × the setting value of ATT[2:0] This setting timing is limited by setting the base clock cycles. For details, see section 31.3.11. The function of adjust transmit timing (Asynchronous Mode).	R/W
31	AET	Adjustment Edge for Transmit Timing Valid in Asynchronous mode using internal clock and Simple LIN mode using internal clock. The adjustable edge is set by this bit. This bit enables only when ATEN is 1. For details, see section 31.3.11. The function of adjust transmit timing (Asynchronous Mode). 0: When CCR1.TINV is 0, adjust the rising edge timing. When CCR1.TINV is 1, adjust the falling edge timing. 1: When CCR1.TINV is 0, adjust the falling edge timing. When CCR1.TINV is 1, adjust the rising edge timing.	R/W

Note: S-TYPE-3, P-TYPE-3

CMPD[8:0] bits (Compare Match Data)

Set the comparison data for receive data when address match function is enabled (CCR0.DCME = 1). CCR4.CMPD[8:0] should be written while CCR0.DCME is 0.

For the comparison data, it can select length from three types, they are CMPD[6:0] with 7-bit length, CMPD[7:0] with 8-bit length, and CMPD[8:0] with 9-bit length.

Note: If the description in this document and the timing chart do not specify the ASEN / ATEN setting value, it means that the reception sampling adjustment function/transmission timing adjustment function are OFF (CCR4.ASEN = 0, CCR4.ATEN = 0).

ASEN bit (Adjust Receive Sampling Timing Enable)

When the ASEN bit is 1, the receive sampling timing adjustment function is enabled. Control is different in Asynchronous mode, Simple LIN mode, Clock-synchronous mode, and Simple SPI mode.

For details of using internal clock in Asynchronous mode, see section 31.6.7. Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with internal clock used. The operation when the Simple LIN mode internal clock is selected is the same as when the Asynchronous mode internal clock is selected.

For details of Clock synchronous mode operating as master and Simple SPI mode operating as master, see section 31.6.6. Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode. Only the digital delay of the master mode receive sampling clock can be controlled by this bit.

ATEN bit (Adjust Transmit Timing Enable)

When the ATEN bit is 1, the transmission timing adjustment function is enabled. The transmission timing adjustment function can adjust the edge timing of the waveform output from the TXDn pin. For details, see section 31.3.11. The function of adjust transmit timing (Asynchronous Mode).

SCKSEL bit (Master receive clock selection)

In Clock synchronous or Simple-SPI mode, SCKSEL bit selects whether to enable or disable the internal clock for master reception.

AST[2:0] bits (Adjustment Value for Receive Sampling Timing)

When the ASEN bit is 1, the receive sampling timing can be adjusted according to this bit setting value.

In Asynchronous mode and Simple LIN mode using internal clock.

The sampling timing of RXDn pin is adjusted from the middle of bit by the following formula. This setting value is limited by setting the base clock cycles. For details, see section 31.3.10. The function of adjust receive sampling timing (Asynchronous Mode).

Adjustment sampling timing = base clock × the setting value of AST[2:0].

In Clock-synchronous mode and Simple SPI mode using internal clock

The sampling timing of RXDn pin can be adjusted by delaying 1 to 4 TCLK. For details, see section 31.6.7. Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with internal clock used.

000: 1-TCLK delay

001: 2-TCLK delay

010: 3-TCLK delay

011: 4-TCLK delay

1xx: Setting prohibited

Note: In Clock-synchronous mode and Simple SPI mode, it is recommended to set delay time within half the period of the SCK clock cycle.

AJD bit (Adjustment Direction for Receive Sampling Timing)

Set the RXDn pin sampling timing adjustment direction from the bit center to the rear or front. For details, see section 31.3.10. The function of adjust receive sampling timing (Asynchronous Mode).

ATT[2:0] bits (Adjustment Value for Transmit Timing)

The edge timing of the TXDn pin specified by the AET bit is adjusted by the base clock × ATT[2:0] setting value. The upper limit of the adjustment time that can be set is limited by the number of base clock cycles. For details, see section 31.3.11. The function of adjust transmit timing (Asynchronous Mode).

AET bit (Adjustment Edge for Transmit Timing)

Set the TXDn pin edge for timing adjustment. For details, see section 31.3.11. The function of adjust transmit timing (Asynchronous Mode).

31.2.10 ICR : Simple IIC Control Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 4, 9)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	IICSCLS[1:0]		IICSDAS[1:0]		—	IICSTPREQ	IICRSTAREQ	IICSTAREQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	IICACKT	—	—	—	IICCSC	IICINTM	—	—	—	IICDL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	IICDL[4:0]	SDA Delay Output Select The following cycles are of the clock signal from the on-chip baud rate generator: 0x00: No output delay 0x01: 0 to 1 cycle 0x02: 1 to 2 cycles 0x03: 2 to 3 cycles 0x04: 3 to 4 cycles 0x05: 4 to 5 cycles ⋮ 0x1E: 29 to 30 cycles 0x1F: 30 to 31 cycles	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	IICINTM	IIC Interrupt Mode Select 0: Use ACK/NACK interrupts 1: Use reception and transmission interrupts	R/W
9	IICCSC	Clock Synchronization 0: No synchronization with the clock signal 1: Synchronization with the clock signal	R/W
12:10	—	These bits are read as 0. The write value should be 0.	R/W
13	IICACKT	ACK Transmission Data 0: ACK transmission 1: NACK transmission and reception of ACK/NACK	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	IICSTAREQ	Start Condition Generation 0: A start condition is not generated 1: A start condition is generated *1 *3 *4 *5	R/W
17	IICRSTAREQ	Restart Condition Generation 0: A restart condition is not generated 1: A restart condition is generated *2 *3 *4 *5	R/W
18	IICSTPREQ	Stop Condition Generation 0: A stop condition is not generated. 1: A stop condition is generated *2 *3 *4 *5	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
21:20	IICSDAS[1:0]	SDA Output Select 0 0: Serial data output 0 1: Generate a start, restart, or stop condition 1 0: Output the low level on the SDAn pin 1 1: Place the SDAn pin in the high-impedance state	R/W

Bit	Symbol	Function	R/W
23:22	IICSCLS[1:0]	SCL Output Select 0 0: Serial clock output 0 1: Generate a start, restart, or stop condition 1 0: Output the low level on the SCLn pin 1 1: Place the SCLn pin in the high-impedance state	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. In bus free state, perform the start condition generation.
Note 2. In bus busy state, perform restart or stop condition generation when the SCLn pin after acknowledgment described in Figure 31.86 and Figure 31.87 is low level.
Note 3. Do not set more than one from among the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits to 1 at a given time.
Note 4. Execute the generation of a condition after the value of the IICSTIF flag is 0.
Note 5. Do not write 0 to this bit while it is 1. Generation of a condition is suspended by writing 0 to this bit while it is 1.

IICDL[4:0] bits (SDA Delay Output Select)

The IICDL[4:0] bits are used to set a delay for output on the SDAn pin relative to the falling edge of the output on the SCLn pin. The available delay settings range from no delay to 31 cycles, with the clock signal from the on-chip baud rate generators the base. The signal obtained by frequency-dividing TCLK by the divisor set in CCR2.CKS[1:0] is supplied as the clock signal from the on-chip baud rate generator.

Set these bits to 00000b unless operation is in simple IIC mode. In simple IIC mode, set the bits to a value in the range from 00001b to 11111b.

IICINTM bit (IIC Interrupt Mode Select)

The IICINTM bit selects the sources of interrupt requests in Simple IIC mode.

IICCSC bit (Clock Synchronization)

Set the IICCSC bit to 1 if the internally generated SCLn clock signal is to be synchronized when the SCLn pin is placed at the low level and when a wait is inserted by other device, for example.

The SCLn clock signal is not synchronized if the IICCSC bit is 0. The SCLn clock signal is generated in accord with the rate selected in the BRR regardless of the level being input on the SCLn pin.

Set the IICCSC bit to 1 except during debugging.

IICACKT bit (ACK Transmission Data)

Transmitted data contains ACK bits. Set this bit to 1 when ACK and NACK bits are received.

IICSTAREQ bit (Start Condition Generation)

When a start condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTAREQ bit to 1.

To generate the start condition after generating the stop condition, start the generation of the start condition with a half cycle period of the bit rate from the stop condition generation interrupt (STI) request output.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the start condition

IICRSTAREQ bit (Restart Condition Generation)

When a restart condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICRSTAREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the restart condition

IICSTPREQ bit (Stop Condition Generation)

When a stop condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTPREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the stop condition

IICSDAS[1:0] bits (SDA Output Select)

The IICSDAS[1:0] bits control output from the SDAn pin.

IICSCLS[1:0] bits (SCL Output Select)

The IICSCLS[1:0] bits control output from the SCLn pin.

31.2.11 FCR : FIFO Control Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 4, 9)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	RSTRG[4:0]					RFRST	—	—	RTRG[4:0]				
Value after reset:	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TFRST	—	—	TTRG[4:0]					—	—	—	—	—	—	—	DRES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DRES	Receive Data Ready Error Select Bit Valid in Asynchronous mode. This bit selects the interrupt request for a reception data ready detection. 0: Reception data full interrupt (SCIn_RXI) 1: Receive error interrupt (SCIn_ERI)	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
12:8	TTRG[4:0]	Transmit FIFO Data Trigger Number Valid in Asynchronous mode (including Multiprocessor mode), Clock synchronous mode, and Simple SPI mode. Trigger number must be set to 15 or less. 0x00: Trigger number 0 ⋮ 0x1F: Trigger number 31	R/W
14:13	—	These bits are read as 0. The write value should be 0.	R/W
15	TFRST	Transmit FIFO Data Register Reset This bit enables only when CCR3.FM is 1. The read value is always 0. 0: When set, this bit is invalid and does not affect operation 1: The number of data stored in Transmit-FIFO (TDR register) is 0	W

Bit	Symbol	Function	R/W
20:16	RTRG[4:0]	Receive FIFO Data Trigger Number Valid in Asynchronous mode (including Multi processor mode), Clock synchronous mode, and Simple SPI mode. Trigger number must be set to 15 or less. 0x00: Trigger number 0 ⋮ 0x1F: Trigger number 31	R/W
22:21	—	These bits are read as 0. The write value should be 0.	R/W
23	RFRST	Receive FIFO Data Register Reset This bit enables only when CCR3.FM is 1. The read value is always 0. 0: When set, this bit is invalid and does not affect operation 1: The number of data stored in Receive-FIFO(RDR register) is 0	W
28:24	RSTRG[4:0]	RTS Output Active Trigger Number Select Valid in Asynchronous mode (including Multi processor mode) and Clock synchronous mode. This bit enables only when CCR3.FM = 1 and CCR1.CTSE = 0 and CCR0.SSE = 0. Trigger number must be set to 15 or less. 0x00: Trigger number 0 ⋮ 0x1F: Trigger number 31	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

DRES bit (Receive Data Ready Error Select Bit)

Select whether the detection of receive data ready (FRSR.DR flag = 1) is the cause of SCIn_RXI interrupt request or the cause of SCIn_ERI interrupt request.

TTRG[4:0] bits (Transmit FIFO Data Trigger Number)

The TDRE flag is set to 1 when the quantity of transmit data in the transmit-FIFO(TDR register) is equal to or less than the specified transmit triggering number. If CCR0.TIE = 1, SCIn_TXI interrupt request occurs.

Note: Trigger number must be set to 15. When the trigger number is set to 16 or more, unexpected SCIn_TXI interrupt occurs.

TFRST bit (Transmit FIFO Data Register Reset)

When the TFRST bit is set to 1, the number of the transmission data stored in Transmit-FIFO(TDR register) is 0.

RTRG[4:0] bits (Receive FIFO Data Trigger Number)

The CSR.RDRF flag is set to 1 when the quantity of receive data in the receive-FIFO(RDR register) is equal to or greater than the specified receive triggering number. If CCR0.RIE = 1, SCIn_RXI interrupt request occurs. When FCR.RTRG is set to 0, RDRF bit is set if the quantity of data in receive-FIFO is greater than or equal to 1.

Note: Trigger number must be set to 15. When the trigger number is set to 16 or more, unexpected SCIn_RXI interrupt occurs.

RFRST bit (Receive FIFO Data Register Reset)

When the RFRST bit is set to 1, the number of the reception data stored in Receive-FIFO(RDR register) is 0.

RSTRG[4:0] bits (RTS Output Active Trigger Number Select)

When the quantity of receive data stored in the receive-FIFO (RDR register) is equal to or greater than this number, the RTSn signal is in the High state. When FCR.RSTRG is set to 0, RTSn is in the high state if the quantity of data in receive FIFO is greater than or equal to 1.

Note: Trigger number must be set to 15. When the trigger number is set to 16 or more, RTSn goes to a high state at unexpected timing.

31.2.12 MCR : Manchester Control Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0)

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	SBERE N	SYERE N	PFERE N	—	—	RPPAT[1:0]		RPLEN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TPPAT[1:0]		TPLEN[3:0]				—	SBSEL	SYNSEL	SYNVAL	—	ERTEN	TMPOL	RMPOL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RMPOL	Polarity of Received Manchester Code Sets the polarity of the received Manchester code. 0: Logic 0 is coded as a zero-to-one transition in Manchester code Logic 1 is coded as a one-to-zero transition in Manchester code 1: Logic 0 is coded as a one-to-zero transition in Manchester code Logic 1 is coded as a zero-to-one transition in Manchester code	R/W
1	TMPOL	Polarity of Transmit Manchester Code Sets the polarity of the transmit Manchester code 0: Logic 0 is coded as a zero-to-one transition in Manchester code Logic 1 is coded as a one-to-zero transition in Manchester code 1: Logic 0 is coded as a one-to-zero transition in Manchester code Logic 1 is coded as a zero-to-one transition in Manchester code	R/W
2	ERTEN	Manchester Edge Retiming Enable Sets the receive retiming function. 0: Disables the receive retiming function 1: Enables the receive retiming function	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	SYNVAL	SYNC value Setting Sets the SYNC type of the start bit(s) in the Manchester code. When the start bit area consists of one bit (SBSEL = 0) ● When transmitting: 0: The start bit is added as a zero-to-one transition 1: The start bit is added as a one-to-zero transition ● When receiving: 0: Only when the start bit is a zero-to-one transition, the data is received. Other cases are determined as an error. 1: Only when the start bit is a one-to-zero transition, the data is received. Other cases are determined as an error. When the start bit area consists of three bits (SBSEL = 1) ● When transmitting: 0: The start bits are added as a zero-to-one transition (DATA SYNC) 1: The start bits are coded as a one-to-zero transition (COMMAND SYNC) ● When receiving: When the start bit area consists of three bits, data is received regardless of the value of this bit.	R/W
5	SYNSEL	SYNC Select 0: The start bit pattern is set with the SYNVAL bit 1: The start bit pattern is set with the TSYNC bit	R/W
6	SBSEL	Start Bit Select 0: The start bit area consists of one bit 1: The start bit area consists of three bits (COMMAND SYNC or DATA SYNC)	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
11:8	TPLEN[3:0]	Transmit Preface Length Set the preface length of the transmit data in Manchester mode. 0x0: Disables the transmit preface generation Others: Transmit preface length (bit length)	R/W
13:12	TPPAT[1:0]	Transmit Preface Pattern Set the preface pattern of the transmit data. 0 0: ALL ZERO 0 1: ZERO ONE 1 0: ONE ZERO 1 1: ALL ONE	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
19:16	RPLEN[3:0]	Receive Preface Length Set the preface length in received frames when Manchester mode is enabled. 0x0: Disables the receive preface generation Others: Receive preface length (bit length)	R/W
21:20	RPPAT[1:0]	Receive Preface Pattern Set the preface pattern of received frames 0 0: ALL ZERO 0 1: ZERO ONE 1 0: ONE ZERO 1 1: ALL ONE	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
24	PFEREN	Preface Error Enable Specifies whether to handle a preface error as an interrupt source. 0: Does not handle a preface error as an interrupt source 1: Handles a preface error as an interrupt source	R/W
25	SYEREN	Receive SYNC Error Enable Specifies whether to handle a receive SYNC error as an interrupt source. 0: Does not handle a receive SYNC error as an interrupt source 1: Handles a receive SYNC error as an interrupt source	R/W
26	SBEREN	Start Bit Error Enable Specifies whether to handle a start bit error as an interrupt source. 0: Does not handle a start bit error as an interrupt source 1: Handles a start bit error as an interrupt source	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

RMPOL bit (Polarity of Received Manchester Code)

The RMPOL bit sets the polarity of the received Manchester code. For details, see section 31.5.7. Serial Data Reception in Manchester Mode.

TMPOL bit (Polarity of Transmit Manchester Code)

The TMPOL bit sets the polarity of the transmit Manchester code. For details, see section 31.5.6. Serial data transmission in Manchester mode.

ERTEN bit (Manchester Edge Retiming Enable)

The ERTEN bit sets the receive retiming function in Manchester mode.

For information on the receive retiming function, see section 31.5.9. Receive Retiming.

SYNVAL bit (SYNC value Setting)

The SYNVAL bit is valid when the SYNSEL bit of the MCR is set to 0.

The SYNC type can be set by combining this bit and the SBSEL bit.

For the start bit area determined by the combination of this bit and the SBSEL bit, see Figure 31.54 and Figure 31.55.

SYNSEL bit (SYNC Select)

The SYNSEL bit is valid when the SBSEL bit of the MCR is set to 1. This bit determines the destination to be referred to for setting the SYNC type of the start bit area added to Manchester frames.

When this bit is set to 0, the SYNVAL bit of the MCR is referred to.

When this bit is set to 1, the TSYNC bit in the TDR register is referred to.

SBSEL bit (Start Bit Select)

The SBSEL bit sets the start bit area in Manchester frames.

When this bit is set to 1, the start bit area added to each frame consists of three bits, and the SYNSEL and SYNVAL bits in this register are valid.

When this bit is set to 0, the start bit area added to each frame consists of one bit.

TPLEN[3:0] bits (Transmit Preface Length)

The TPLEN[3:0] bits set the preface bit length of the transmit data in Manchester mode.

The settable range is 0x0 to 0xF (0d to 15d). The setting of 0x0 disables the transmit preface, which is not added.

TPPAT[1:0] bits (Transmit Preface Pattern)

The TPPAT[1:0] bits set one of the four preface patterns in Manchester mode. For the transmit data when the TPPAT[1:0] bits are set, see Figure 31.53.

When these bits are set to 00b, the preface area is set to all 0s.

When these bits are set to 01b, the preface area is set to the zero-one-zero-one pattern.

When these bits are set to 10b, the preface area is set to the one-zero-one-zero pattern.

When these bits are set to 11b, the preface area is set to all 1s.

RPLEN[3:0] bits (Receive Preface Length)

The RPLEN[3:0] bits set the preface bit length of the received frames in Manchester mode.

The settable range is 0x0 to 0xF (0d to 15d). The setting of 0x0 disables the receive preface, which is not added. When 0x1 to 0xF is set, the set value is handled as the receive preface bit length.

RPPAT[1:0] bits (Receive Preface Pattern)

The RPPAT[1:0] bits set one of the four preface patterns in Manchester mode. For the transmit and receive data when the RPPAT[1:0] bits are set, see Figure 31.53.

When these bits are set to 00b, the preface area is handled as all 0s.

When these bits are set to 01b, the preface area is handled as the zero-one-zero-one pattern.

When these bits are set to 10b, the preface area is handled as the one-zero-one-zero pattern.

When these bits are set to 11b, the preface area is handled as all 1s.

PFEREN bit (Preface Error Enable)

The PFEREN bit specifies whether to handle a preface error as an interrupt source.

When it is set to 0, a preface error is not handled as an interrupt source. When it is set to 1, a preface error is handled as an interrupt source.

SYEREN bit (Receive SYNC Error Enable)

The SYEREN bit specifies whether to handle a receive SYNC error as an interrupt source.

When it is set to 0, a receive SYNC error is not handled as an interrupt source. When it is set to 1, a receive SYNC error is handled as an interrupt source.

SBEREN bit (Start Bit Error Enable)

The SBEREN bit specifies whether to handle a start bit error as an interrupt source.

When it is set to 0, a start bit error is not handled as an interrupt source. When it is set to 1, a start bit error is handled as an interrupt source.

31.2.13 DCR : Driver Control Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 4, 9)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	DENGT[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DEAST[4:0]					—	—	—	—	—	—	—	DEPOL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DEPOL	Driver Effective Polarity Select Valid only in Asynchronous mode. 0: The DEn signal is active-high 1: The DEn signal is active-low	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
12:8	DEAST[4:0]	Driver Assertion Time Valid only in Asynchronous mode. Set the driver assertion time. When DEN = 1, the driver assertion time is inserted in addition to the normal transmission waiting time. Setting DEAST[4:0] = 0x00 is prohibited.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
20:16	DENGT[4:0]	Driver Negate Time Valid only in Asynchronous mode. Set the driver negation time. When DEN = 1, the driver negate time is inserted after STOP bit transmission end. Setting DENGT[4:0] = 0x00 is prohibited.	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

DEPOL bit (Driver Effective Polarity Select)

Select the active level of the DEn signal.

DEAST[4:0] bits (Driver Assertion Time)

Set the driver assertion time (= time from the activation of the DEn (Driver Enable) signal to the start of the start bit). It is expressed in base clock period.

Driver assertion time

= DEAST [4:0] set value × base clock period + transmission waiting time

DENGT[4:0] bits (Driver Negate Time)

Set the driver negation time (= time from the end of the last stop bit of the transmitted message until the DEn (Driver Enable) signal is disabled). It is expressed in base clock period.

Driver negate time

= DENGT[4:0] set value × base clock period

If the transmission data is written during the driver negate time, transmit starting operation is different depends on the writing timing. The DEn signal remains valid, and transmission of the start bit may start after the transmission wait time has

elapsed. Also, the DEn signal may become invalid once, and start bit transmission may start after the Driver assertion time has elapsed.

31.2.14 XCR0 : Simple LIN Control Register 0

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0, 1)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x34

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	BCCS[1:0]		—	AEDIE	COFIE	BFDIE	—	—	BCDIE	BFOIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PIBS[2:0]			PIBE	CF1DS[1:0]		CF0RE	BFE	—	—	—	—	—	—	TCSS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TCSS[1:0]	Timer Count Clock Source Selection Valid in Simple LIN mode. Select the clock source of the timer in Simple LIN module. 0 1: TCLK/4 1 0: TCLK/16 1 1: TCLK/64	R/W*1
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	BFE	Break Field Enable Set the presence or absence of Break Field of Start Frame. 0: No Break Field 1: With Break Field	R/W*3
9	CF0RE	Control Field 0 Enable Set the presence or absence of Control Field 0 of Start Frame 0: No Control Field 0 1: With Control Field 0	R/W*3
11:10	CF1DS[1:0]	Control Field1 Compare Data Select Select the compare data for Control Field 1. 0 0: Select XCR1.PCF1D[7:0] as the compare data 0 1: Select XCR1.SCF1D[7:0] as the compare data 1 0: Select both XCR1.PCF1D[7:0] and XCR1.SCF1D[7:0] as the compare data 1 1: Setting prohibited	R/W*3
12	PIBE	Priority Interrupt Bit Enable 0: Priority interrupt bit disabled 1: Priority interrupt bit enabled	R/W*3
15:13	PIBS[2:0]	Priority Interrupt Bit Select Specify one of bits 0 to 7 of Control Field 1 as the priority interrupt bit. 0 0 0: Bit 0 of Control Field 1 0 0 1: Bit 1 of Control Field 1 0 1 0: Bit 2 of Control Field 1 0 1 1: Bit 3 of Control Field 1 1 0 0: Bit 4 of Control Field 1 1 0 1: Bit 5 of Control Field 1 1 1 0: Bit 6 of Control Field 1 1 1 1: Bit 7 of Control Field 1	R/W*3
16	BFOIE	Break Field Output Completion Interrupt Enable Select whether to include Break Field output completion as a SCIn_TXI interrupt factor. 0: Break Field output completion is not included in SCIn_TXI interrupt factor 1: Break Field output completion is included in SCIn_TXI interrupt factor	R/W

Bit	Symbol	Function	R/W
17	BCDIE	Bus Conflict Detection Interrupt Enable Select whether to output an SCIn_ERI interrupt when a bus collision is detected. 0: Bus conflict detection is not included in SCIn_ERI interrupt factor 1: Bus conflict detection is included in SCIn_ERI interrupt factor	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
20	BFDIE	Break Field Detection Interrupt Enable Select whether to output a SCIn_BFD interrupt when a Break Field is detected. 0: Break Field detection interrupt disabled 1: Break Field detection interrupt enabled	R/W
21	COFIE	Counter Overflow Interrupt Enable Select whether to include counter overflow as an SCIn_ERI interrupt factor. 0: Counter overflow is not included in SCIn_ERI interrupt factor 1: Counter overflow is included in SCIn_ERI interrupt factor	R/W
22	AEDIE	Active Edge Detection Interrupt Enable Select whether to output an SCIn_AED interrupt when a valid edge is detected. 0: Active edge detection interrupt disabled 1: Active edge detection interrupt enabled	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
25:24	BCCS[1:0]	Bus Conflict Detection Clock Selection Select the sampling clock for the bus conflict detection circuit. When CCR2.ABCS = 1, setting BCCS[1] = 1 is prohibited. 0 0: Base clock*2 0 1: Base clock/2 1 0: Base clock/4 1 1: Setting prohibited	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. You can rewrite TCSS[1:0] only when the timer is stopped (TCST = 0 and SDST = 0 and BMEN = 0)
Note 2. Base clock: 1/16 period of 1 bit period when CCR2.ABCS = 0, 1/8 period of 1 bit period when CCR2.ABCS = 1.
Note 3. This bit is a setting bit required for the start frame reception operation. Rewrite this bit when the start frame reception or transmission is not in progress (XCR1.SDST = 0 and XCR1.TCST = 0).

TCSS[1:0] bits (Timer Count Clock Source Selection)

Select clock source of timer in Simple LIN module.

BFE bit (Break Field Enable)

Set the presence or absence of Break Field of Start Frame.

CF0RE bit (Control Field 0 Enable)

Set the presence or absence of Control Field 0 of Start Frame.

CF1DS[1:0] bits (Control Field1 Compare Data Select)

Select the compare data for Control Field 1.

PIBE bit (Priority Interrupt Bit Enable)

Select whether to enable priority interrupt bit comparison of Control Field 1. When the PIBE bit is 1, regardless of the XCR1.CF1CE [7:0] setting value, the bit specified in PIBS [2:0] is compared with the primary comparison data for Control Field 1 (XCR1.PCF1D [7:0]).

PIBS[2:0] bits (Priority Interrupt Bit Select)

Specify bit N (N = 0-7) of Control Field 1 as the priority interrupt bit.

BFOIE bit (Break Field Output Completion Interrupt Enable)

Select whether to include Break Field output completion as a SCIn_TXI interrupt factor. Set CCR0.TIE = 1 and CCR3.MOD [1: 0] = 110b, to output SCIn_TXI upon completion of Break Field output.

BCDIE bit (Bus Conflict Detection Interrupt Enable)

Select whether to output an SCIn_ERI interrupt when a bus collision is detected. In Simple LIN mode (CCR3.MOD [1:0] = 110b), SCIn_ERI output control is performed with the BCDIE bit. When CCR3.MOD [1:0] = 110b and BCDIE = 1, an SCIn_ERI interrupt is issued when a bus collision is detected even if CCR0.RIE = 0.

COFIE bit (Counter Overflow Interrupt Enable)

Select whether to include counter overflow as an SCIn_ERI interrupt factor. Set CCR0.RIE = 1 and CCR3.MOD [1:0] = 110b are required to output SCIn_ERI upon counter overflow.

AEDIE bit (Active Edge Detection Interrupt Enable)

Select whether to output an SCIn_AED interrupt when a valid edge is detected. To output SCIn_AED with valid edge detection, XCR1.BMEN = 1 and CCR3.MOD [1:0] = 110b must be set.

BCCS[1:0] bits (Bus Conflict Detection Clock Selection)

Select the sampling clock for the bus conflict detection circuit.

31.2.15 XCR1 : Simple LIN Control Register 1

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0, 1)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CF1CE[7:0]								SCF1D[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PCF1D[7:0]								—	—	BMEN	SDST	—	—	—	TCST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCST	Break Field Output Timer Count Start Trigger 0: Break Field output timer count stopped 1: Break Field output timer count start	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	SDST	Start Frame Detection Enable 0: Start Frame/Break Field detection disabled 1: Start Frame/Break Field detection enabled Do not set this bit and TCST bit to 1 at the same time.	R/W
5	BMEN	Bit Rate Measurement Enable Set this bit to 1 simultaneously with the SDST bit. This bit can be set to 0 at any timing. 0: Bit rate measurement disabled 1: Bit rate measurement enabled	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PCF1D[7:0]	Priority Compare Data for Control Field 1 The priority compare data for Control Field 1.	R/W
23:16	SCF1D[7:0]	Secondary Compare Data for Control Field 1 The secondary compare data for Control Field 1.	R/W
31:24	CF1CE[7:0]	Control Field 1 Compare Bit Enable Select whether to compare bit N of Control Field 1. (N = 0 -7) 0: Control Field 1 bit N compare disabled 1: Control Field 1 bit N compare enabled	R/W

Note: S-TYPE-3, P-TYPE-3

TCST bit (Break Field Output Timer Count Start Trigger)

[Clearing condition]

- When 0 is written to TCST. Break Field output timer count is stopped and TXDn output becomes idle level.
- When Break Field output for the period set in XCR2.BFLW [15: 0] is completed.

[Setting condition]

- When 1 is written to TCST. Start Break Field output from TXDn. Holds 1 during Break Field output.

SDST bit (Start Frame Detection Enable)

When 1 is written to the SDST bit, Start Frame detection starts. When XCR0.BFE = 1 is set, Break Field can be detected during Start Frame is detected and after Start Frame is detected. When XCR0.BFE = 0 is set, Break Field is not detected.

When 0 is written to the SDST bit, Start Frame detection and Break Field detection are stopped. However, if XSR0.RXDSF = 0 at the time of stop, it is not possible to stop data reception of the SCI core with this bit. Write 0 to CCR0.RE to stop the reception operation or perform reception completion processing (CSR.RDRF clear or RDR read) after reception is completed.

BMEN bit (Bit Rate Measurement Enable)

Set this bit to 1 simultaneously with the SDST bit. When the BMEN bit is set to 1, the valid edge interval of Control Field 0 and Control Field 1 data is measured.

PCF1D[7:0] bits (Priority Compare Data for Control Field 1)

Set the priority compare data for Control Field 1.

SCF1D[7:0] bits (Secondary Compare Data for Control Field 1)

Set the secondary compare data for Control Field 1.

CF1CE[7:0] bits (Control Field 1 Compare Bit Enable)

Select whether to compare bit N of Control Field 1. (N = 0 - 7)

When all of the CF1CE[7:0] bits are set to 0 (CF1CE[7:0] = 0x00), it is always determined that Control Field 1 matches when reception is completed, and XSR0.CF1MF is set. This bit is a comparison enable with PCF1D[7:0] or SCF1D[7:0], and is not a priority interrupt bit comparison enable.

31.2.16 XCR2 : Simple LIN Control Register 2

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0, 1)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x3C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BFLW[15:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CF0CE[7:0]								CF0D[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CF0D[7:0]	Control Field 0 Compare Data The compare data for Control Field 0.	R/W
15:8	CF0CE[7:0]	Control Field 0 Compare Bit Enable Select whether to compare bit N of Control Field 0. (N = 0 - 7) 0: Control Field 0 bit N compare disabled 1: Control Field 0 bit N compare enabled	R/W

Bit	Symbol	Function	R/W
31:16	BFLW[15:0]	Break Field Length Setting This register sets the Break Field length. The Break Field length is (BFLW [15:0] setting value + 1) × clock of the timer. The upper limit for setting this register is 0xFFFE. Setting is prohibited for 0xFFFF.	R/W

Note: S-TYPE-3, P-TYPE-3

CF0D[7:0] bits (Control Field 0 Compare Data)

The compare data for Control Field 0.

CF0CE[7:0] bits (Control Field 0 Compare Bit Enable)

Select whether to compare bit N of Control Field 0. (N = 0 - 7)

When all of the CF0CE[7:0] bits are set to 0 (CF0CE[7:0] = 0x00), it is always determined that Control Field 0 matches when reception is completed, and XSR0.CF0MF is set.

BFLW[15:0] bits (Break Field Length Setting)

BFLW[15:0] are 16-bit Break Field length setting bits and the initial value is 0xFFFE.

Set the Break Field length to 1 frame or more. The LIN standard stipulates that the Break Field length is 13 bits or more.

When the Break Field is sending, writing 1 to TCST causes SCI to output the Break Field on TXDn. Up-counting is performed with the clock of the timer selected by XCR0.TCSS[1:0]. When the count value matches the value set in this register, up-counting is stopped and Break Field output from TXDn is also stopped.

When the Break Field is receiving, writing 1 to SDST enables Start Frame detection. SCI starts counting from the negative edge of RXDn. The clock of the timer is selected by XCR0.TCSS[1:0].

When the count value matches the value set in the XCR2, a Break Field has been detected. Up-counting continues until the next valid edge or counter overflow.

31.2.17 CSR : Common Status Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 4, 9)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x48

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RDRF	TEND	TDRE	FER	PER	MFF	—	ORER	—	—	—	—	—	DFER	DPER	DCMF
Value after reset:	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RXDMON	—	—	—	—	—	—	—	—	—	—	ERS	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0.	R
4	ERS	Error Signal Status Flag Valid only in Smart card interface mode. 0: Error signal low not responded 1: Error signal low responded	R
14:5	—	These bits are read as 0.	R
15	RXDMON	Serial Input Data Monitor Bit The state of the RXDn pin without synchronizing by bus clock is shown. 0: When RINV is 0, RXDn pin is the low level. When RINV is 1, RXDn pin is the high level. 1: When RINV is 0, RXDn pin is the high level. When RINV is 1, RXDn pin is the low level.	R

Bit	Symbol	Function	R/W
16	DCMF	Data Compare Match Flag Valid only in Asynchronous mode. 0: No matched 1: Matched	R
17	DPER	Data Compare Match Parity Error Flag Valid only in Asynchronous mode. 0: No parity error occurred at address match detection 1: A parity error has occurred at address match detection	R
18	DFER	Data Compare Match Framing Error Flag Valid only in Asynchronous mode. 0: No framing error occurred at address match detection 1: A framing error has occurred at address match detection	R
23:19	—	These bits are read as 0.	R
24	ORER	Overrun Error Flag 0: No overrun error occurred 1: An overrun error has occurred	R
25	—	This bit is read as 0.	R
26	MFF	Mode Fault Error Flag Valid only in Simple SPI mode. 0: No mode fault error 1: Mode fault error	R
27	PER	Parity Error Flag 0: Non-FIFO selected (CCR3.FM = 0): No parity error occurred FIFO selected (CCR3.FM = 1): No parity error in all received data in receive-FIFO 1: Non-FIFO selected (CCR3.FM = 0): A parity error has occurred FIFO selected (CCR3.FM = 1): One or more parity errors occurred in received data in receive-FIFO	R
28	FER	Framing Error Flag 0: Non-FIFO selected (CCR3.FM = 0): No framing error occurred FIFO selected (CCR3.FM = 1): No framing error in all received data in receive-FIFO 1: Non-FIFO selected (CCR3.FM = 0): A framing error has occurred FIFO selected (CCR3.FM = 1): One or more framing errors occurred in received data in receive-FIFO	R
29	TDRE	Transmit Data Empty Flag 0: Non-FIFO selected (CCR3.FM = 0): Transmit data is in TDR register FIFO selected (CCR3.FM = 1): The quantity of transmit data written in transmit-FIFO exceeds the specified transmit triggering number. 1: Non-FIFO selected (CCR3.FM = 0): No transmit data is in TDR register FIFO selected (CCR3.FM = 1): The quantity of transmit data written in transmit-FIFO is equal to or less than the specified transmit triggering number.	R
30	TEND	Transmit End Flag 0: A character is being transmitted or standing by for transmission. 1: Character transfer has been completed, or sending Break Field.	R
31	RDRF	Receive Data Full Flag 0: Non-FIFO selected (CCR3.FM = 0): No received data is in RDR register FIFO selected (CCR3.FM = 1): The quantity of receive data written in receive-FIFO falls below the specified receive triggering number. 1: Non-FIFO selected (CCR3.FM = 0): Received data is in RDR register FIFO selected (CCR3.FM = 1): The quantity of receive data written in receive-FIFO is equal to or greater than the specified receive triggering number.	R

Note: S-TYPE-3, P-TYPE-3

ERS bit (Error Signal Status Flag)

[Setting condition]

- When an error signal low is sampled.

[Clearing condition]

- When write 1 to CFCLR.ERSC.

DCMF bit (Data Compare Match Flag)

The DCMF bit indicates that SCI detects the match to the comparison data (CCR4.CMPD) with receive data.

Clearing the CCR0.RE bit to 0 does not affect the DCMF flag, which retains its previous state.

[Setting condition]

- Matched to the comparison data (CCR4.CMPD) with receive data, while CCR0.DCME = 1.

[Clearing condition]

- When write 1 to CFCLR.DCMFC.

DPER bit (Data Compare Match Parity Error Flag)

The DPER bit indicates that a parity error occurred at address match detection (reception data match detection).

Clearing the CCR0.RE bit to 0 does not affect the DPER flag, which retains its previous state.

[Setting condition]

- When a parity error is detected by the frame in which an address match was detected.

[Clearing condition]

- When writing 1 to CFCLR.DPERC.

DFER bit (Data Compare Match Framing Error Flag)

The DFER bit indicates that a framing error occurred at address match detection (reception data match detection).

Clearing the CCR0.RE bit to 0 does not affect the DFER flag, which retains its previous state.

[Setting condition]

- When a stop bit of the frame in which an address match was detected is 0.
  When it is a 2-stop mode, the 1st bit of the stop bit only determines whether it is 1 and does not check the 2nd bit of the stop bit.

[Clearing condition]

- When writing 1 to CFCLR.DFERC.

ORER bit (Overrun Error Flag)

The ORER bit indicates that an overrun error has occurred during reception and the reception ends abnormally.

Clearing the CCR0.RE bit to 0 does not affect the ORER flag, which retains its previous state. In Simple IIC mode, the ORER bit is not used.

[Setting condition with non-FIFO mode (CCR3.FM = 0)]

- When the next data is received before reading out RDR with no-error reception data stored in RDR.
  In RDR, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, reception data is not forwarded to RDR register.

  Note: In Clock synchronous mode and Simple SPI mode, serial reception is stopped.

[Setting condition with FIFO mode (CCR3.FM = 1)]

- When the next serial reception is completed while the receive FIFO is full of 16 receive data.

[Clearing condition]

- When write 1 to CFCLR.ORERC.

MFF bit (Mode Fault Error Flag)

The MFF bit indicates mode fault errors. In a multi-master configuration, determine the mode fault error occurrence by reading the MFF flag.

[Setting condition]

- Input on the SSn pin being at the low level during master operation (CCR3.CKE[1:0] = 00 or 01) in Simple SPI mode.

[Clearing condition]

- When writing 1 to CFCLR.MFFC.

PER bit (Parity Error Flag)

The PER bit indicates that a parity error has occurred during reception and the reception ends abnormally.

Clearing the CCR0.RE bit to 0 does not affect the PER flag, which retains its previous state.

In Clock synchronous mode, Simple SPI mode, and Simple IIC mode, the PER bit is not used.

[Setting condition]

- When a parity error is detected during reception. In FIFO select mode, when one or more parity error is detected in receive-FIFO data.
  In non-FIFO mode, although receive data when the parity error occurs is transferred to RDR, no SCIn_RXI interrupt request occurs. Note that when the PER flag is set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When writing 1 to CFCLR.PERC.

FER bit (Framing Error Flag)

The FER bit indicates that a framing error has occurred during reception and the reception ends abnormally.

Clearing the CCR0.RE bit to 0 does not affect the FER flag, which retains its previous state.

In Clock synchronous mode, Simple SPI mode, and Simple IIC mode, the FER bit is not used.

[Setting condition]

- When 0 is sampled as the stop bit during reception. In FIFO select mode, when one or more framing error is detected in receive-FIFO data. In Manchester mode, when both sampling results (1⁄4 and 3⁄4 sampling points) are not 1 for 1 stop bit. In Simple LIN mode, even if a condition that changes to 1 occurs when XCR1.SDST = 1, the FER set timing is delayed up to the Break Field evaluation timing at the longest, since it may be a Break Field. If an edge is detected on the RXDn signal before the Break Field evaluation timing, FER is detected. If no edge is detected in the RXDn signal before the Break Field evaluation timing, Break Field is detected.
  In 2-stop-bit mode, only the first stop bit is checked whether it is 1 but the second stop bit is not checked. In non-FIFO mode, although receive data when the framing error occurs is transferred to RDR, no SCIn_RXI interrupt request occurs. In addition, when the FER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When writing 1 to CFCLR.FERC.

TDRE bit (Transmit Data Empty Flag)

[Non-FIFO selected (CCR3.FM = 0)]

The TDRE bit indicates the presence of transmit data in the TDR register.

The condition of CCR0.TE = 0 has priority over the condition of 0.

If other conditions that become 1 and conditions that become 0 are satisfied at the same time, the TDRE flag is set to 0.

[Setting condition]

- When CCR0.TE is 0.
- When data is transmitted from the TDR register to the TSR register.

[Clearing condition]

- When writing 1 to CFCLR.TDREC.
- When the transmission data is written to the TDR register during CCR0.TE = 1.

[FIFO selected (CCR3.FM = 1)]

The TDRE bit indicates that data has been transferred from the transmit-FIFO (TDR) into the transmit shift register (TSR), the quantity of data in transmit-FIFO has fallen below the specified transmit triggering number.

When the condition which becomes 1 and the condition which becomes 0 are formed at the same time, TDRE flag is 0. After that, when the number of data stored in transmit-FIFO is evaluated, and if that is equal to or less than TTRG value, TDRE is set to 1 after 1 PCLK.

[Setting condition]

- When the quantity of transmit data written in transmit-FIFO is equal to or less than the specified transmit triggering number*1.

Note 1. The transmit-FIFO is FIFO register of 16 stages, the maximum number of data that can be written when the TDRE flag is 1 is indicated in 0x10 - FTSR.T[5:0]. Even if more data is written, data is discarded.

[Clearing condition]

- When writing 1 to CFCLR.TDREC.
- When the transmission data is written to transmit-FIFO by the DTC or DMAC (the last block transfer of block transfer).

TEND bit (Transmit End Flag)

[Non-FIFO selected (CCR3.FM = 0), and not Smart card interface mode (CCR3.MOD[2:0] ≠ 001)]

The TEND bit indicates completion of transmission.

[Setting condition]

- When CCR0.TE is 0.
- When the CCR0.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated at the time of transmission of the tail-end bit of a character being transmitted.
- When the TDR register is not updated at the end of DE negate time with DE control function enable (CCR3.DEN = 1).
- When Break Field is sending.

[Clearing condition]

- After the synchronization delay time has elapsed since the transmission data was written to the TDR register during CCR0.TE = 1.
- When writing 1 to CFCLR.TDREC during CCR0.TE = 1.

[Non-FIFO selected (CCR3.FM = 0), and Smart card interface mode (CCR3.MOD [2:0] = 001)]

With no error signal from the receiving side, this bit is set to 1 when additional data for transfer is ready to be transferred to the TDR register.

[Setting condition]

- When CCR0.TE is 0.
- When the CCR0.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period has elapsed after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated. The set timing is determined by register settings as listed below.1
  When GM = 0 and BLK = 0, 12.5etu after the start of transmission
  When GM = 0 and BLK = 1, 11.5etu after the start of transmission
  When GM = 1 and BLK = 0, 11.0etu after the start of transmission
  When GM = 1 and BLK = 1, 11.0etu after the start of transmission

[Clearing condition]

- After the synchronization delay time has elapsed since the transmission data was written to the TDR register during CCR0.TE = 1.
- When writing 1 to CFCLR.TDREC during CCR0.TE = 1.

[FIFO selected (CCR3.FM = 1)]

The TEND bit indicates that the transmit-FIFO does not contain valid data when transmitting the last bit of a serial character, so the transmission is halted.

[Setting condition]

- TEND is set to 1 when transmit-FIFO does not contain transmit data when the last bit of a one-byte serial character is transmitted.
- When the TDR register is not updated at the end of DE negate time with DE control function enable (CCR3.DEN = 1).

[Clearing condition]

- After the synchronization delay time has elapsed since the transmission data was written to the TDR register during CCR0.TE = 1.

RDRF bit (Receive Data Full Flag)

[Non-FIFO selected (CCR3.FM = 0)]

The RDRF bit indicates the presence of receive data in the RDR register.

[Setting condition]

- When the reception ends normally, and receive data is forwarded from the RSR register to the RDR register.

[Clearing condition]

- When writing 1 to CFCLR.RDRFC.
- When the read data is read from the RDR register.

[FIFO selected (CCR3.FM = 1)]

The RDRF bit indicates that receive data has been transferred to the receive FIFO data register (RDR), and the quantity of data in receive-FIFO is equal to or greater than the specified receive triggering number. When FCR.RTRG is set to 0, RDRF is set if the quantity of data in receive-FIFO is greater than or equal to 1.

[Setting condition]

- RDRF is set to 1 when the quantity of receive data in receive-FIFO is equal to or greater than the specified receive triggering number*1.

Note 1. Since the receive-FIFO is 16 stage FIFO register, the maximum quantity of data that can be read when RDRF is 1 is equivalent to the specified receive data count number (FDR.R[5:0]). If an attempt is made to read after all the data in receive-FIFO has been read, the data is undefined.

[Clearing condition]

- When writing 1 to CFCLR.RDRFC.
- When the reception data is read form receive-FIFO by the DTC or DMAC (the last block transfer of a block transfer).

When the condition which becomes 1 and the condition which becomes 0 are formed at the same time, RDRF flag is 0. After that, when the number of stored data in receive-FIFO is evaluated, and if that is equal to or greater than RTRG value, RDRF is set to 1 after 1 PCLK.

Note: When Non-FIFO is selected, except when interruption communication, RDRF and TDRE should not be cleared by CFCLR register.

31.2.18 ISR : Simple IIC Status Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 4, 9)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x4C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	IICSTIF	—	—	IICACKR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IICACKR	ACK Reception Data Flag 0: ACK received 1: NACK received	R
1	—	This bit is read as 0.	R
2	—	The read value is undefined.	R
3	IICSTIF	Issuing of Start, Restart, or Stop Condition Completed Flag 0: There are no requests for generating conditions or a condition is being generated. 1: A start, restart, or stop condition is completely generated.	R
5:4	—	The read value is undefined.	R
31:6	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

IICACKR bit (ACK Reception Data Flag)

Received ACK and NACK bits can be read from this bit.

The IICACKR flag is updated at the rising of SCLn clock for the ACK/NACK receiving bit.

IICSTIF bit (Issuing of Start, Restart, or Stop Condition Completed Flag)

After generating a condition, this bit indicates that the generation is completed. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after setting the IICSTIF flag to 0.

When the IICSTIF flag is 1 while an interrupt request is enabled by setting the CCR0.TEIE bit, an STIn request is output.

[Setting condition]

- Completion of the generation of a start, restart, or stop condition (however, in cases where this conflicts with any of the conditions for the flag becoming 0 listed below, the clearing condition takes precedence)

[Clearing condition]

- Writing 1 to ICFCLR.IICSTIFC bit
- When operation is not in Simple IIC mode
- Writing 0 to CCR0.TE bit

31.2.19 FRSR : FIFO Receive Status Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 4, 9)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	FNUM[5:0]						—	—	PNUM[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	R[5:0]						—	—	—	—	—	—	—	DR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0

Bit	Symbol	Function	R/W
0	DR	Receive Data Ready Flag 0: Receiving is in progress, or no received data remains in the receive-FIFO after a successful reception (receive-FIFO is empty) 1: The following receive data is not received for a fixed period after storing data under the threshold in the receive-FIFO	R
1	—	The read value is undefined.	R
7:2	—	These bits are read as 0.	R
13:8	R[5:0]	Receive-FIFO Data Count Valid in Asynchronous mode (including Multiprocessor mode), Clock synchronous mode, and Simple SPI mode, when CCR3.FM is 1. Indicate the quantity of receive data stored in receive-FIFO	R
15:14	—	These bits are read as 0.	R
21:16	PNUM[5:0]	Parity Error Count Valid only in Asynchronous mode Indicates the quantity of data with a parity error for the receive data stored in the Receive FIFO Data register.	R
23:22	—	These bits are read as 0.	R
29:24	FNUM[5:0]	Framing Error Count Valid only in Asynchronous mode. Indicates the quantity of data with a framing error for the receive data stored in the Receive FIFO Data register.	R
31:30	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

DR bit (Receive Data Ready Flag)

The DR bit indicates that the quantity of data stored in the Receive FIFO Data register (RDR) falls below the specified receive triggering number, and that no next data has been received yet after the elapse of 15 etu from the last stop bit in Asynchronous mode. This bit is valid only Asynchronous mode (including Multiprocessor mode) and FIFO selected. In other mode, this bit does not set to 1.

[Setting conditions]

- DR is set to 1 when the following conditions are met:
  - After receive FIFO Data register (RDR) receives less data than the specified receive triggering number, no next data has been received yet after the elapse of 15 etu*1 from the last stop bit
  - CSR.FER, PER flags are 0.

Note 1. This is equivalent to one and a half (1.5) frames in the 8-bit format with one stop bit (etu: elementary time unit).

[Clearing conditions]

- When all receive data in the receive FIFO Data (RDR register) is read and 1 is written to FFCLR.DRC.

- When CCR3.FM bit is 0.

R[5:0] bits (Receive-FIFO Data Count)

The R[5:0] bits indicate the quantity of receive data stored in the Receive FIFO Data register.

0x00 means no receive data. 0x10 means receive-FIFO is full.

PNUM[5:0] bits (Parity Error Count)

The value indicates the quantity of data stored in the Receive FIFO Data register with a parity error.

FNUM[5:0] bits (Framing Error Count)

The value indicates the quantity of data stored in the Receive FIFO Data registers with a framing error.

31.2.20 FTSR : FIFO Transmit Status Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 4, 9)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x54

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	T[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	T[5:0]	Transmit-FIFO Data Count Valid in Asynchronous mode (including Multiprocessor mode), Clock synchronous mode, and Simple SPI mode, when CCR3.FM is 1. Indicate the quantity of non-transmit data stored in transmit-FIFO	R
31:6	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

T[5:0] bits (Transmit-FIFO Data Count)

The T[5:0] bits indicate the quantity of non-transmitted data stored in transmit-FIFO.

0x00 means no non-transmit data. 0x10 means transmit-FIFO is full.

31.2.21 MSR : Manchester Status Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0)

Offset address: 0x58

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	RSYNC	—	MER	—	SBER	SYER	PFER
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PFER	Preface Error Flag This bit is set when a preface error (pattern mismatch) is detected 0: No preface error detected 1: Preface error detected	R
1	SYER	SYNC Error Flag This bit is set when no edge is detected in the adjustable range during receive retiming 0: No receive SYNC error detected 1: Receive SYNC error detected	R
2	SBER	Start Bit Error Flag This bit is set when a pattern mismatch in the start bit area is detected 0: No start bit error detected 1: Start bit error detected	R
3	—	This bit is read as 0.	R
4	MER	Manchester Error Flag Valid for Manchester mode only 0: No Manchester error occurred 1: Manchester error has occurred	R
5	—	This bit is read as 0.	R
6	RSYNC	Receive SYNC Data Bit It is valid when MCR.SBSEL = 1 in Manchester mode, 0 is read otherwise. 0: The received start bit is DATA SYNC 1: The received start bit is COMMAND SYNC	R
31:7	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

PFER bit (Preface Error Flag)

The PFER bit indicates that a preface error was detected when receiving frames in Manchester mode.

Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the PFER flag is not affected and retains its previous value.

[Setting condition]

- When detecting a preface error when receiving frames in Manchester mode
  The following operations are performed when a preface error occurs.
  - When MCR.PFEREN = 1
    The received data is not transferred to the RDR register and no SCIn_RXI interrupt request occurs. Instead, an SCIn_ERI interrupt request occurs. Note that when the PFER flag is set to 1, the subsequently received data is not transferred to the RDR register.
  - When MCR.PFEREN = 0
    The received data is transferred to the RDR register and an SCIn_RXI interrupt request occurs. An SCIn_ERI interrupt request is not generated. The subsequent receive operations are not affected even with the PFER flag is set to 1.

[Clearing condition]

- Writing 1 to MFCLR.PFERC.

SYER bit (SYNC Error Flag)

The SYER bit indicates that a receive SYNC error was detected when receiving frames in Manchester mode with MCR.ERTEN = 1 (Manchester edge retiming enabled).

Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the SYER flag is not affected and retains its previous value.

[Setting condition]

- When detecting a receive SYNC error when receiving frames in Manchester mode
  The following operations are performed when a receive SYNC error occurs.

– When MCR.SYEREN = 1
Although the received data is transferred to the RDR register, no SCIn_RXI interrupt request occurs. Instead, an SCIn_ERI interrupt request occurs. Note that when the SYER flag is set to 1, the subsequently received data is not transferred to the RDR register.

– When MCR.SYEREN = 0
The received data is transferred to the RDR register and an SCIn_RXI interrupt request occurs. An SCIn_ERI interrupt request is not generated. The subsequent receive operations are not affected even with the SYER flag is set to 1.

[Clearing condition]

- Writing 1 to MFCLR.SYERC.

SBER bit (Start Bit Error Flag)

The SBER bit indicates that a start bit error was detected when receiving frames in Manchester mode.

Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the SBER flag is not affected and retains its previous value.

[Setting condition]

- When detecting a start bit error when receiving frames in Manchester mode
The following operations are performed when a start bit error occurs.
  – When MCR.SBEREN = 1
  The received data is not transferred to the RDR register and no SCIn_RXI interrupt request occurs. Instead, an SCIn_ERI interrupt request occurs. Note that when the SBER flag is set to 1, the subsequently received data is not transferred to the RDR register.
  – When MCR.SBEREN = 0
  The received data is transferred to the RDR register and an SCIn_RXI interrupt request occurs. An SCIn_ERI interrupt request is not generated. The subsequent receive operations are not affected even with the SBER flag is set to 1.

[Clearing condition]

- Writing 1 to MFCLR.SBERC.

MER bit (Manchester Error Flag)

When data is received in Manchester mode, Manchester error is detected and displayed. Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the MER flag is not affected and retains its previous value.

[Setting conditions]

- When receiving in Manchester mode and detecting Manchester code error in data area of received frame.
Received data when an error occurs is transferred to the RDR register, but the SCIn_RXI interrupt request is not generated and the SCIn_ERI interrupt request is generated.
When the Manchester error flag is set to 1, subsequent receive data is not transferred to the RDR register.
For details on Manchester error, section 31.5.11. Errors in Manchester Mode.

[Clearing condition]

- Writing 1 to MFCLR.MERC.

RSYNC bit (Receive SYNC Data Bit)

When Manchester mode (CCR3.MOD[2:0] = 101b) and MCR.SBSEL = 1, this bit indicates the type of SYNC of the received start bit. For other settings, it is fixed to 0.

31.2.22 XSR0 : Simple LIN Status Register 0

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0, 1)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x5C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CF1RD[7:0]								CF0RD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AEDF	COF	PIBDF	CF1MF	CF0MF	BFDF	BCDF	BFOF	—	—	—	—	—	—	RXDSF	SFSF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SFSF	Start Frame Status Flag 0: Start Frame detection disabled or Start Frame detection complete 1: Before Start Frame detection or during detection	R*1
1	RXDSF	RXDn Input Status Flag 0: RXDn input to SCI is enabled 1: RXDn input to SCI is disabled	R*1
7:2	—	These bits are read as 0.	R
8	BFOF	Break Field Output Completion Flag 0: Break Field is being output or is not being output 1: Completed BF output	R
9	BCDF	Bus Conflict Detection Flag 0: No bus collision detected 1: Detected a bus collision	R
10	BFDF	Break Field Detection Flag 0: Break Field is not detected 1: Break Field is detected	R
11	CF0MF	Control Field 0 Compare Match Flag 0: Control Field 0 received data does not match the setting data 1: Control Field 0 received data matches the setting data	R
12	CF1MF	Control Field 1 Compare Match Flag 0: Control Field 1 received data does not match the setting data 1: Control Field 1 received data matches the setting data	R
13	PIBDF	Priority Interrupt Bit Detection Flag 0: Priority interrupt bit is not detected 1: Priority interrupt bit is detected	R
14	COF	Counter Overflow Flag 0: Break Field detection counter has not overflowed 1: Break Field detection counter overflowed	R
15	AEDF	Active Edge Detection Flag 0: No valid edge detected 1: Detected a valid edge	R
23:16	CF0RD[7:0]	Control Field 0 received data Control Field 0 received data.	R
31:24	CF1RD[7:0]	Control Field 1 received data Control Field 1 received data.	R

Note: S-TYPE-3, P-TYPE-3

Note 1. When PCLK is faster than TCLK, the flag set timing is delayed from the receive data full interrupt (SCIn_RXI) output. To see this flag under these conditions, wait at least 1 TCLK cycle after the receive data full interrupt (SCIn_RXI) before reading this register.

SFSF bit (Start Frame Status Flag)

Indicates whether detect Start Frame is being detected.

[Setting condition]

- When 1 is written to XCR1.SDST.
- When a Break Field is detected in the Control Field 0/Control Field 1/Information Field phase and the transition to the Control Field 0 or Control Field 1 reception state occurs.

[Clearing condition]

- When XCR1.SDST is 0.
- When Start Frame detection is completed.

RXDSF bit (RXDn Input Status Flag)

Indicates the RXDn input status to the SCI core. When the RXDSF bit is 1, RXDn input is received only by the simple LIN module and the Break Field is detected and is not input to the SCI core.

BFOF bit (Break Field Output Completion Flag)

Indicates the completion of Break Field output.

The BFOF bit can be cleared to 0 by writing 1 to XFCLR.BFOC.

BCDF bit (Bus Conflict Detection Flag)

Indicates the detection of bus conflict in Simple LIN transmit operation.

The BCDF bit can be cleared to 0 by writing 1 to XFCLR.BCDC.

BFDF bit (Break Field Detection Flag)

Indicates Break Field detection.

The BFDF bit can be cleared to 0 by writing 1 to XFCLR.BFDC.

CF0MF bit (Control Field 0 Compare Match Flag)

Indicates compare match of Control Field 0 and compare data.

The CF0MF bit can be cleared to 0 by writing 1 to XFCLR.CF0MC.

CF1MF bit (Control Field 1 Compare Match Flag)

Indicates compare match detection of Control Field 1 and compare data.

The CF1MF bit can be cleared to 0 by writing 1 to XFCLR.CF1MC.

PIBDF bit (Priority Interrupt Bit Detection Flag)

Indicates compare match detection of Control Field 1 and priority interrupt bit.

The PIBDF bit can be cleared to 0 by writing 1 to XFCLR.PIBDC.

COF bit (Counter Overflow Flag)

Indicates that the 16-bit counter in the simple LIN module has overflowed.

The COF bit can be cleared to 0 by writing 1 to XFCLR.COFC.

AEDF bit (Active Edge Detection Flag)

Indicates active edge detection.

The AEDF bit can be cleared to 0 by writing 1 to XFCLR.AEDC and when read out XSR1.TCNT[15:0].

CF0RD[7:0] bits (Control Field 0 received data)

Stores the received data with a Control Field 0 match detected.

CF1RD[7:0] bits (Control Field 1 received data)

Stores the received data with a Control Field 1 match detected.

31.2.23 XSR1 : Simple LIN Status Register 1

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0, 1)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x60

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TCNT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TCNT[15:0]	Timer Count Capture Value Stores the 16-bit counter capture value. The initial value is 0000.	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

TCNT[15:0] bits (Timer Count Capture Value)

Stores the capture value of the 16-bit counter of the Simple LIN module.

- When sending Start Frame
  The XSR1 holds the previous value.
- When receiving Start Frame with bit rate measurement disabled
  If a Break Field is detected in the Break Field detection state (see Figure 31.81), the Break Field length is captured and held (counter value is captured at the rising edge of RXDn). If a Break Field is detected in a state other than the Break Field detection state, the previous value is retained.
  If the counter overflows, it is not captured.
- When receiving Start Frame with bit rate measurement enabled
  The count value is captured and held at the valid edge (both RXDn edges). However, in the Break Field detection state, the count value is not captured even if a valid edge occurs. Counter capture value retention is canceled by the XSR1 read. Even if a valid edge occurs before reading, the counter value is not captured.

31.2.24 CFCLR : Common Flag Clear Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 4, 9)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x68

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RDRFC	—	TDREC	FERC	PERC	MFFC	—	ORERC	—	—	—	—	—	DFERC	DPERC	DCMFC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	ERSC	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	The write value should be 0.	W
4	ERSC	ERS Clear Bit Setting this bit to 1 clears the CSR.ERS bit. The read value is always 0.	W
15:5	—	The write value should be 0.	W
16	DCMFC	DCMF Clear Bit Setting this bit to 1 clears the CSR.DCMF bit. The read value is always 0.	W
17	DPERC	DPER Clear Bit Setting this bit to 1 clears the CSR.DPER bit. The read value is always 0.	W
18	DFERC	DFER Clear Bit Setting this bit to 1 clears the CSR.DFER bit. The read value is always 0.	W
23:19	—	The write value should be 0.	W
24	ORERC	ORER Clear Bit Setting this bit to 1 clears the CSR.ORER bit. The read value is always 0.	W
25	—	The write value should be 0.	W
26	MFFC	MFF Clear Bit Setting this bit to 1 clears the CSR.MFF bit. The read value is always 0.	W
27	PERC	PER Clear Bit Setting this bit to 1 clears the CSR.PER bit. The read value is always 0.	W
28	FERC	FER Clear Bit Setting this bit to 1 clears the CSR.FER bit. The read value is always	W
29	TDREC	TDRE Clear Bit Setting this bit to 1 clears the CSR.TDRE bit. The read value is always 0.	W
30	—	The write value should be 0.	W
31	RDRFC	RDRF Clear Bit Setting this bit to 1 clears the CSR.RDRF bit. The read value is always 0.	W

Note: S-TYPE-3, P-TYPE-3

31.2.25 ICFCLR : Simple IIC Flag Clear Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 4, 9)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x6C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	IICSTIFC	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	The write value should be 0.	W
3	IICSTIFC	IICSTIF Clear Bit Setting this bit to 1 clears the ISR.IICSTIF bit. The read value is always 0.	W
31:4	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

31.2.26 FFCLR : FIFO Flag Clear Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 4, 9)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x70

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DRC	DR Clear Bit Setting this bit to 1 clears the FRSR.DR bit. The read value is always 0.	W
31:1	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

31.2.27 MFCLR : Manchester Flag Clear Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0)

Offset address: 0x74

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	MERC	—	SBERC	SYERC	PFERC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PFERC	PFER Clear Bit Setting this bit to 1 clears the MSR.PFER bit. The read value is always 0.	W
1	SYERC	SYER Clear Bit Setting this bit to 1 clears the MSR.SYER bit. The read value is always 0.	W
2	SBERC	SBER Clear Bit Setting this bit to 1 clears the MSR.SBER bit. The read value is always 0.	W
3	—	The write value should be 0.	W
4	MERC	MER Clear Bit Setting this bit to 1 clears the MSR.MER bit. The read value is always 0.	W
31:5	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

31.2.28 XFCLR : Simple LIN Flag Clear Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0, 1)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x78

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AEDC	COFC	PIBDC	CF1MC	CF0MC	BFDC	BCDC	BFOC	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	The write value should be 0.	W
8	BFOC	BFOF Clear Bit Setting this bit to 1 clears the XSR0.BFOF bit. The read value is always 0.	W
9	BCDC	BCDF Clear Bit Setting this bit to 1 clears the XSR0.BCDF bit. The read value is always 0.	W
10	BFDC	BFDF Clear Bit Setting this bit to 1 clears the XSR0.BFDF bit. The read value is always 0.	W
11	CF0MC	CF0MF Clear Bit Setting this bit to 1 clears the XSR0.CF0MF bit. The read value is always 0.	W
12	CF1MC	CF1MF Clear Bit Setting this bit to 1 clears the XSR0.CF1MF bit. The read value is always 0.	W
13	PIBDC	PIBDF Clear Bit Setting this bit to 1 clears the XSR0.PIBDF bit. The read value is always 0.	W
14	COFC	COFF Clear Bit Setting this bit to 1 clears the XSR0.COF bit. The read value is always 0.	W
15	AEDC	AEDF Clear Bit Setting this bit to 1 clears the XSR0.AEDF bit and cancels holding the XSR1 register. The read value is always 0.	W
31:16	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

31.2.29 CESR : Communication Enable Status Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 4, 9)
SCIn_B_NS = 0x5035_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x1C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TIST	—	—	—	RIST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RIST	RE Internal Status 0: RE signal internal state value 0 1: RE signal internal state value 1	R
3:1	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
4	TIST	TE Internal Status 0: TE signal internal state value 0 1: TE signal internal state value 1	R
7:5	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The operation clocks of the communication module and control register can be used asynchronously. Since some control register values are transmitted internally through the synchronization circuit so that they operate correctly even if they are asynchronous, it takes some time for the state to be reflected internally after rewriting the control register.

Communication enable CCR0.TE and CCR0.RE correspond to this register, and when these control bits change from 1 to 0 to rewrite the control bits for the next communication, the TE and RE bits signal it is necessary to rewrite the next control bit after the internal state becomes 0. If a very slow clock is used for the communication module clock, the TE and RE bit states are not reflected internally. At this time, you can check the internal status using this register.

31.3 Operation in Asynchronous Mode

Figure 31.6 shows the general format for asynchronous serial communications. One frame consists of a start bit (low level), transmit or receive data, a parity bit, and stop bits (high level). In asynchronous serial communications, the communications line is usually held in the mark state (high level).

The SCI monitors the communications line. When the SCI detects a start bit, it starts serial communication. The detection condition of the start bit changes according to the CCR3.RXDESEL setting. SCI regards space (Low level) as a start bit when CCR3.RXDESEL is 0. SCI regards a fall edge as a start bit when RXDESEL is 1.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and receiver have a double-buffered structure in addition to FIFO mode, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

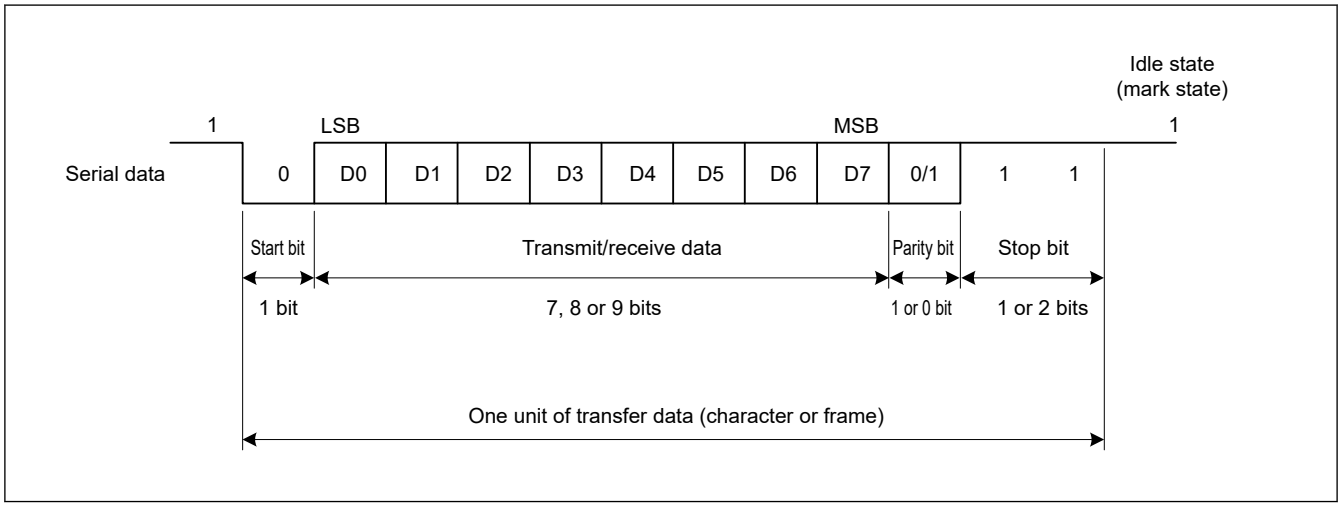


Figure 31.6 Data format in asynchronous serial communications with 8-bit data, parity bit, and 2 stop bits

31.3.1 Serial Data Transfer Format

Table 31.28 lists the serial data transfer formats that can be used in Asynchronous mode. Any of 18 transfer formats can be selected with the CCR1 and CCR3 settings. For details on the multi-processor function, see section 31.4. Multi-Processor Communication Function.

Table 31.28 Serial transfer formats in Asynchronous mode

CCR3		CCR1		CCR3	Serial transfer format and frame length
CHR[1:0]		PE	MP	STP	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	0	0	0	ST 9-bit data SP
0	0	0	0	1	ST 9-bit data SP SP
0	0	1	0	0	ST 9-bit data P SP
0	0	1	0	1	ST 9-bit data P SP SP
1	0	0	0	0	ST 8-bit data SP
1	0	0	0	1	ST 8-bit data SP SP
1	0	1	0	0	ST 8-bit data P SP
1	0	1	0	1	ST 8-bit data P SP SP
1	1	0	0	0	ST 7-bit data SP
1	1	0	0	1	ST 7-bit data SP SP
1	1	1	0	0	ST 7-bit data P SP
1	1	1	0	1	ST 7-bit data P SP SP
0	0	—	1	0	ST 9-bit data MPB SP
0	0	—	1	1	ST 9-bit data MPB SP SP
1	0	—	1	0	ST 8-bit data MPB SP
1	0	—	1	1	ST 8-bit data MPB SP SP
1	1	—	1	0	ST 7-bit data MPB SP
1	1	—	1	1	ST 7-bit data MPB SP SP

ST: Start bit
SP: Stop bit
P: Parity bit
MPB: Multi-processor bit

31.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In Asynchronous mode, the SCI operates on a base clock with a frequency of 16 times[*1] the bit rate.

In reception, the SCI samples the falling edge of the start bit using the base clock, and performs internal synchronization.[*2]

Because receive data is sampled on the rising edge of the 8th pulse[*1] of the base clock, data is latched at the middle of each bit (when sampling timing does not adjust (CCR4.ASEN = 0 or CCR4.ASEN = 1 and CCR4.AST[2:0] = 000b)), as shown in Figure 31.7 The reception margin in Asynchronous mode is determined by the following formula (1):

$$M = \left|\left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F)\right| \times 100\ [\%] \quad \ldots \quad \text{Formula (1)}$$

Note: M: Reception margin

N: Ratio of bit rate to clock

See Table 31.6 for bit rate to basic clock ratios.

D: Duty cycle of clock (D = 0.5 to 1.0)

L: Frame length (L = 9 to 13)

F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined using the following formula:
$M = \{0.5 - 1/(2 \times 16)\} \times 100\ (\%) = 46.875\ \%$

This represents the computed value. Renesas recommends a margin of 20% to 30% in system design.

Note 1. This is an example when CCR2.ABCS is 0, CCR2.ABCSE is 0 and CCR2.ABCSE2 is 0. When CCR2.ABCS is 1, CCR2.ABCSE is 0 and CCR2.ABCSE2 is 0, a frequency of 8 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 4th pulse of the base clock. When CCR2.ABCSE is 1 and CCR2.ABCSE2 is 0, a frequency of 6 times the bit rate is used as a bas clock and receive data is sampled at the rising edge of the 3rd pulse of the base clock. When CCR2.ABCSE is 0 and CCR2.ABCSE2 is 1, a frequency of 4 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 2nd pulse of the base clock.

Note 2. The determination condition of the start bit is as follows.

The function of adjust sampling timing is OFF (ASEN = 0):

The determination condition of a start bit is that Low beyond half bit length continues. It is same as the sampling timing. In Figure 31.7, Low period should be kept over 8-cycles to detect a start bit. If Low period does not keep over 8-cycles, the SCI judges this as a noise. So, the SCI does not start reception and wait start bit.

The function of adjust sampling timing is ON (ASEN = 1):

The determination condition of a start bit is that Low keeps up until the sampling timing. Adjusting the sampling timing forward (AJD = 1) increases the possibility of erroneously determining a noise as the start bit.

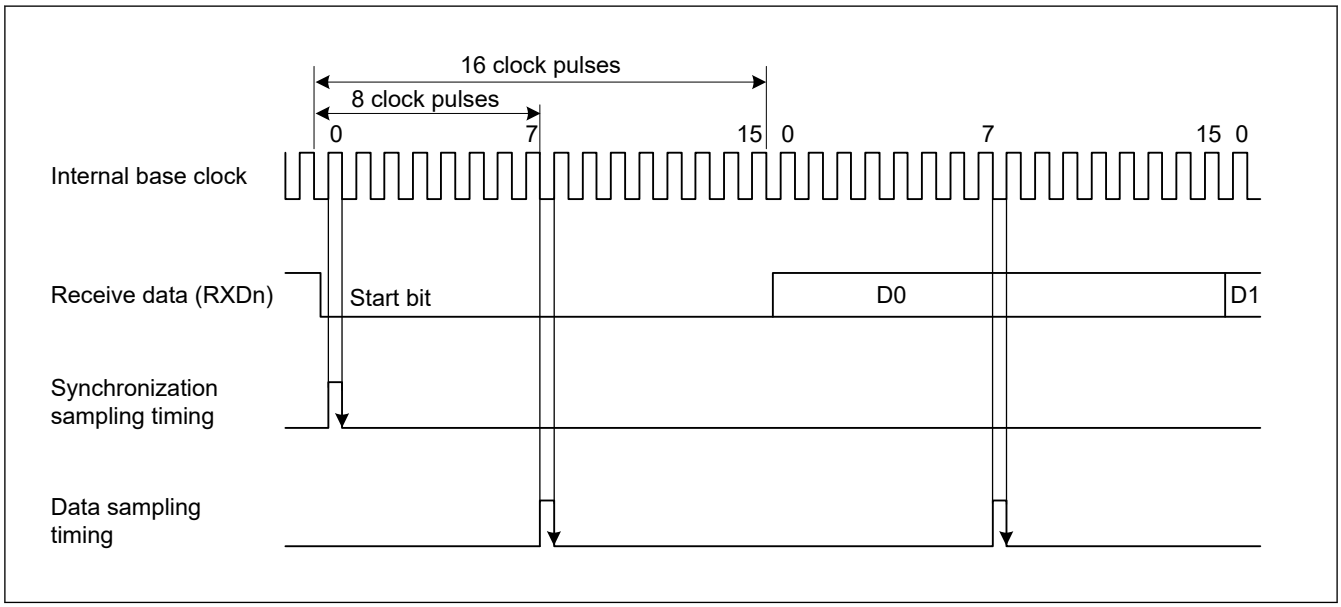


Figure 31.7 Receive data sampling timing in Asynchronous mode

31.3.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input to the SCKn pin can be selected as the transfer clock of the SCI, based on the CCR3.CKE[1:0] settings.

When an external clock is input to the SCKn pin, the clock frequency must be 16 times the bit rate (when CCR2.ABCS = 0) or 8 times the bit rate (when CCR2.ABCS = 1).

In addition, when an external clock is specified, the base clock of GPTn (n = 3, 4, 6, 7) can be selected by the CCR2.ACS0.

When the SCI uses its internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is configured so that the rising edge of the clock is in the middle of the transmit data, as shown in Figure 31.8.

When the internal clock is selected, the SCKn pin is outputted after setting the CCR0.TE or CCR0.RE bit to 1.

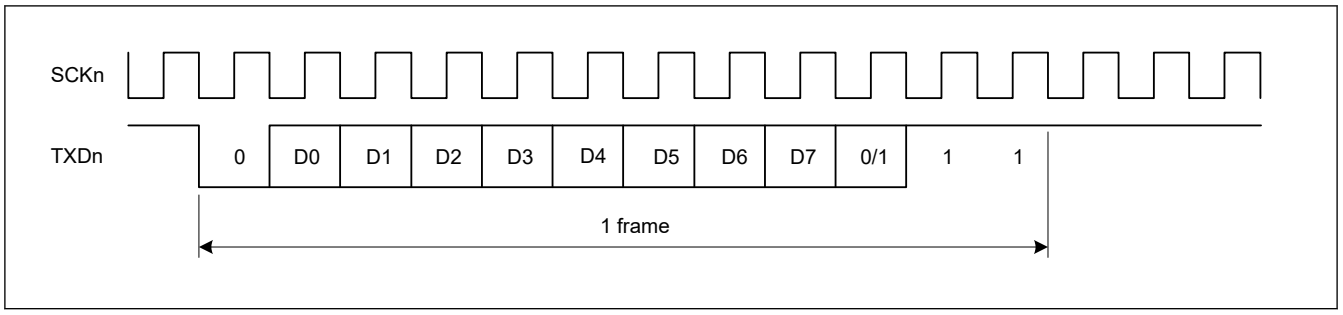


Figure 31.8 Phase relationship between output clock and transmit data in Asynchronous mode when CCR1.PE = 1, CCR3.CHR[1:0] = 10b, MP = 0, and STP = 1

31.3.4 Double-Speed Operation and Frequency of 6 or 4 Times the Bit Rate

When the CCR2.ABCS bit is set to 1, the SCI operates on the bit rate twice that of when ABCS is set to 0. When the CCR2.BGDM bit is set to 1, the cycle of the base clock is half and the bit rate is double that of when BGDM is set to 0. When the CCR3.CKE[1] bit is set to 0 and the on-chip baud rate generator is selected, setting the ABCS and BGDM bits to 1 allows the SCI to operate at a bit rate four times that when the ABCS and BGDM bits are set to 0.

When CCR2.ABCSE is set to 1 and CCR2.ABCSE2 is set to 0, the number of base clock pulses are 6 during a period of 1 bit, and the base clock frequency is half. And RSCI works 16/3 times of bit rate compared with a case of CCR2.ABCS = 0, CCR2.BGDM = 0, CCR2.ABCSE = 0 and CCR2.ABCSE2 = 0.

When CCR2.ABCSE is set to 1 and CCR2.ABCSE2 is set to 1, the number of base clock pulses are 4 during a period of 1 bit, and the base clock frequency is half. And RSCI works 8 times of bit rate compared with a case of CCR2.ABCS = 0, CCR2.BGDM = 0, CCR2.ABCSE = 0 and CCR2.ABCSE2 = 0.

As shown by Formula (1) in section 31.3.2. Receive Data Sampling Timing and Reception Margin in Asynchronous Mode, the reception margin decreases when CCR2.ABCS is set to 1, CCR2.ABCSE is set to 1 or CCR2.ABCSE2 is set to 1. Therefore, if the desired bit rate can be obtained, it is recommended to use the RSCI with CCR2.ABCS set to 1, CCR2.ABCSE set to 1 and CCR2.ABCSE2 set to 1.

31.3.5 CTS and RTS Functions

The CTS function controls transmission using the CTSn pin input. Setting the CCR1.CTSE bit to 1 enables the CTS function. For the functions of CTS and RTS, you can select the alternate setting to set CTSn_RTSn pin as a multiplexed pin that uses either function with one pin or the dedicated setting that uses each function independently with two pins at CTSn pin for CTSn signal and CTSn_RTSn pin for RTSn signal. This setting is done with the CCR1.CTSPEN bit.

When the CTS function is enabled, placing a low level on the CTSn_RTSn pin causes transmission to start.

If FIFO is used and CTSn_RTSn signal is held high before transmission, transmission will not start, so the number of TDR registers written, and the number of data stored are the same (unlike using clock synchronous FIFO).

Driving the CTSn_RTSn pin high while transmission is in progress does not affect transmission of the current frame.

In the RTS function, which uses output on the CTSn_RTSn pin, a low level is output when reception becomes possible. Conditions for output of the low and high levels are shown in this section.

[Conditions for low level output]

Satisfaction of all conditions are listed in this section.

Non-FIFO selected

- The value of the CCR0.RE bit is 1
- When the next reception is possible
  - There are no received data yet to be read and not receiving.
  - CSR.ORER, FER, PER flags are all 0

FIFO selected

- The value of the CCR0.RE bit is 1
- When the next reception is possible
  - When the quantity of receive data written in receive-FIFO(RDR) are less than the setting value of FCR.RSTRG[4:0]
  - CSR.ORER(RDR.ORER) is 0

[Condition for high level output]

- The conditions for low-level output are not satisfied

31.3.6 Address Match (Receive Data Match Detection) Function

The address match function can be used only in Asynchronous mode.

If the CCR0.DCME bit is set to 1[*2], when one frame of data is received, the SCI compares that received data with the data set in CCR4.CMPD. If the SCI detects a match to the comparison data (CCR4.CMPD[*1]) with the received data, the SCI can issue the SCIn_RXI interrupt request.

If the CCR3.MP bit is set to 0, comparison occurs only for valid data in receive format. In multi-processor mode (CCR3.MP bit = 1), if the CCR0.IDSEL bit is set to 1, receive data where the MPB bit is 1 is subject to comparison for address match and receive data where the MPB bit is 0 is always treated as a non-match.

If the CCR0.IDSEL bit is set to 0, SCI performs address match detection regardless of the MPB bit value of the received data.

Until SCI detects a match to the comparison data (CCR4.CMPD[*1]) with receive data, received data is skipped (discarded), and the SCI cannot detect a parity error or framing error.

When SCI detects a match, the CCR0.DCME bit is automatically cleared, and the CSR.DCMF flag is set to 1. If the CCR0.IDSEL bit is set to 1, the CCR0.MPIE bit is automatically cleared. If CCR0.IDSEL is set to 0, the value of the CCR0.MPIE bit is retained. If the CCR0.RIE bit is set to 1, the SCI issues an SCIn_RXI interrupt request.

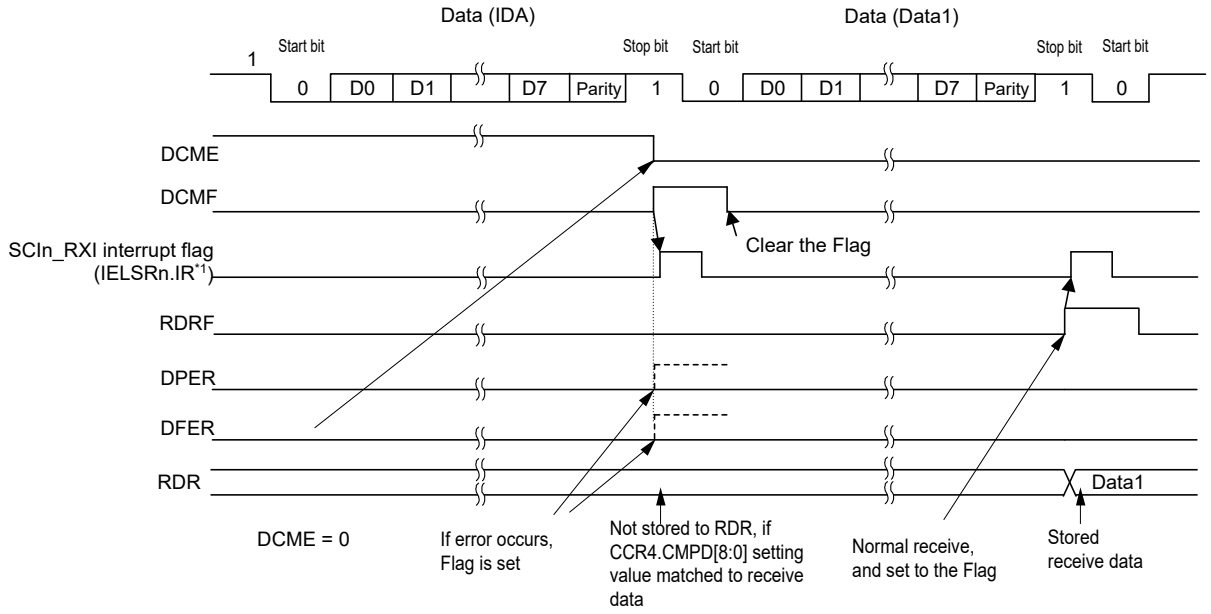
If the SCI detects a framing error in the receive data for which a match is detected, the CSR.DFER flag is set to 1, and if the SCI detects a parity error in that frame, the CSR.DPER flag is set to 1. The compared receive data and MPB bit is not stored in the RDR register, and CSR.RDRF remains 0.

After the SCI detects a match, and CCR0.DCME is automatically cleared, the SCI receives the next data continuously based on the current register setting.

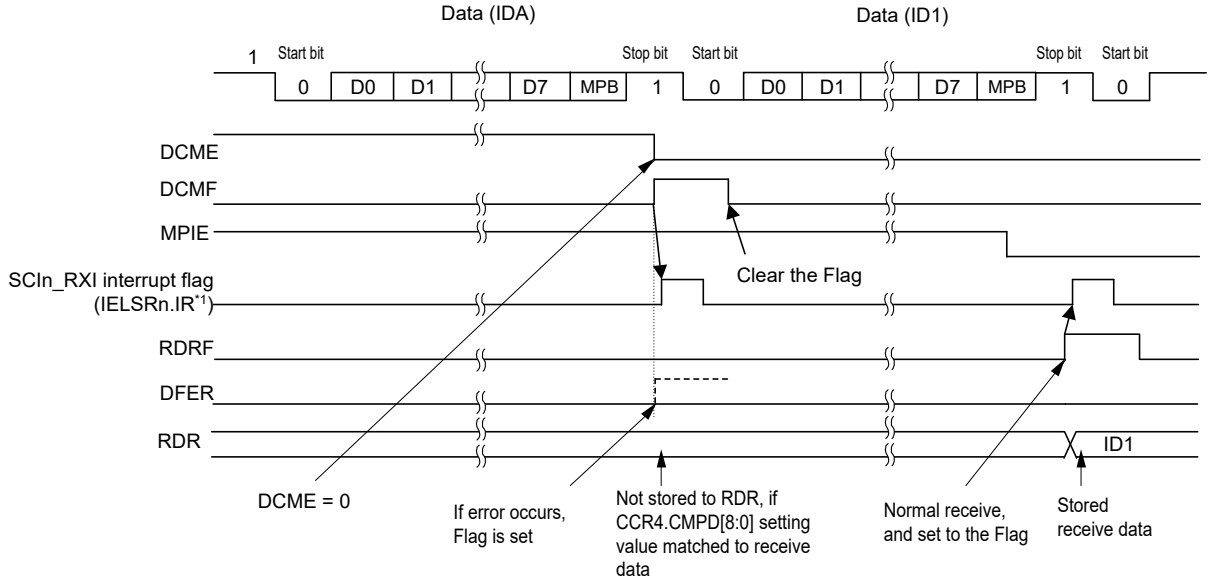
When the CSR.DFER or CSR.DPER flag is set, the address match is not performed. Before enabling the address match function, set the CSR.DFER and CSR.DPER flags to 0.

Examples of the address match function are shown in Figure 31.9 and Figure 31.10.

Note 1. This comparative target can select one length of 3 types: CMPD[6:0] with 7-bit length, CMPD[7:0] with 8-bit length, and CMPD[8:0] with 9-bit length.
Note 2. Set the CCR0.DCME bit to 1 before receiving the start bit of the received frame that performs address matching.



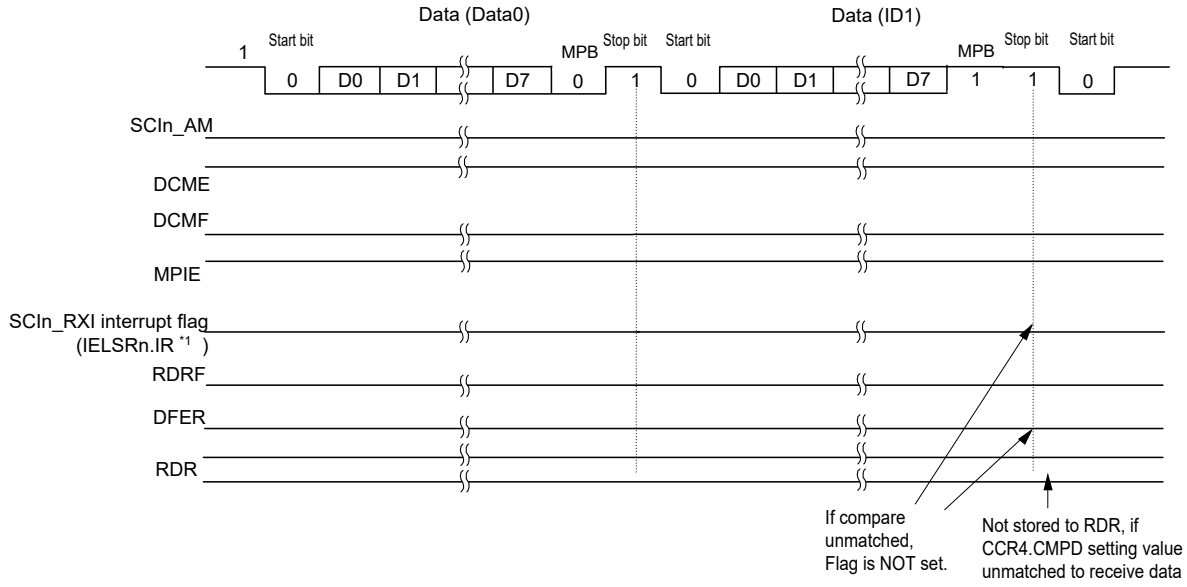
(a) Example of Compare matched between receive data and CCR4.CMPD[8:0] (8bit length/parity/Normal mode)



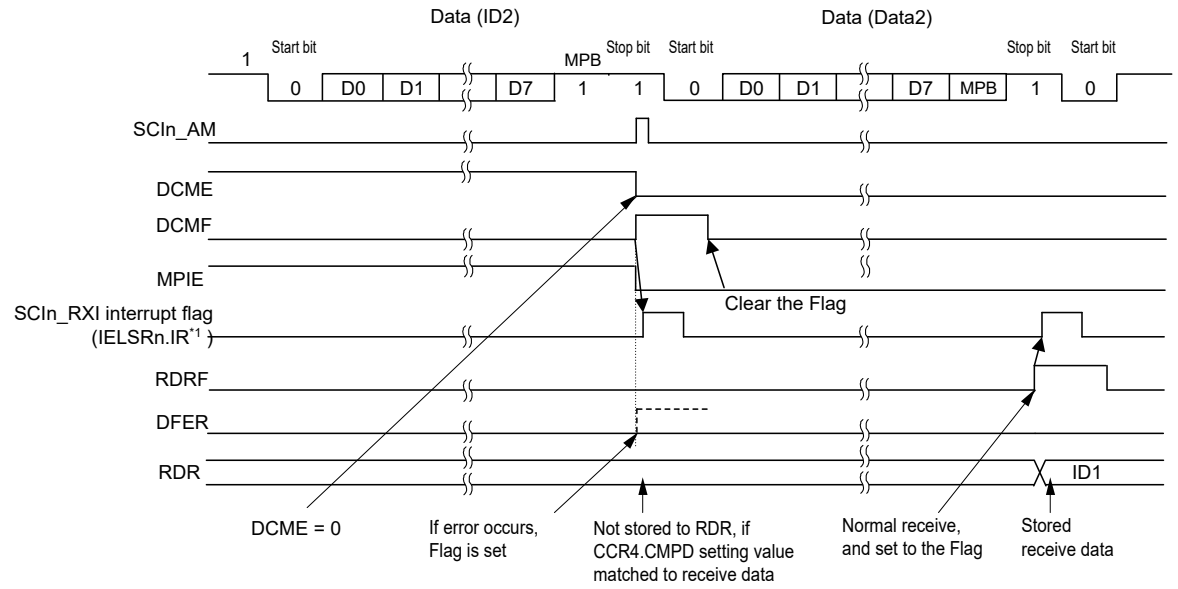
(b) Example of Compare matched between receive data and CCR4.CMPD[8:0] (8bit length/Multi-processor mode)

Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.9 Example of address match (1) normal mode and multi-processor mode



(a) Example of Compare unmatched between receive data and CCR4.CMPD (8bit length/IDSEL = 1/Multi-processor mode)



(b) Example of Compare matched between receive data and CCR4.CMPD (8bit length/IDSEL = 1/Multi-processor mode)

Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.10 Example of address match (2) multi-processor mode

31.3.7 SCI Initialization in Asynchronous Mode

Before transmitting and receiving data, start by writing the initial value 0 to the CCR0.TE and CCR0.RE (or writing the initial value to CCR0), then continue through the SCI initialization procedure (select non-FIFO or FIFO) shown in Table 31.29 and Table 31.30. Whenever the operating mode or transfer format is to be changed, the CCR0.TE and CCR0.RE bits must be initialized before the change is made.

When the external clock is used in Asynchronous mode, ensure that the clock signal is supplied during initialization.

Note: Setting the CCR0.RE bit to 0 initializes neither the ORER, FER, RDRF, RDAT, PER, and DR flags in CSR nor RDR. When FIFO selected, even if the TE bit is set as 0, the TEND flag for the selected FIFO buffer is not initialized.

Please be also careful at the time of change in the operation mode.

Note: Switching the value of the CCR0.TE bit from 0 to 1 while the CCR0.TIE bit is 1 leads to the generation of an SCIn_TXI interrupt request.

Table 31.29 Example flow of SCI initialization in Asynchronous mode with non-FIFO selected

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set CCR0.TEIE, TIE, RIE, TE, RE to 0. If you have not changed from the initial settings, you can skip this step.
3	Set CCR3	Set up following function and communication mode. Driver control function for RS-485, FIFO no-use, Multi-Processor mode, Communication mode (MOD[2:0] = 000b) Transmission / reception format Clock enable (Leave the initial value when outputting the clock) Leave unused bits at their initial values.
4	Set CCR2	Set up the bit-rate-modulation function*1 *2, select clock, set bit rate*2
5	Set CCR1	Set up the Noise-filter function, the loop-back function, communication pin status, the parity check function, and the CTSn_RTSn function.
6	Set CCR4	Set up the adjust sampling timing function and the adjust transmit timing function.
7	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
8	Set CCR3	Set clock enable bit (CKE[1:0]) at this step when outputting the clock. After this register setting, the clock pin will be in the output state immediately.
9	Set CFCLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC If it's initialization flow after a reset, you can skip this step.
10	Set CCR0	Set the TE or RE bit to 1. To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Setting the TE and RE bits allows TXDn and RXDn to be used.
11	Initialization completed	—

Note 1. If you do not use the bit rate modulation function, you do not need to set it.
Note 2. If you use an external clock, you do not need to set it.

Table 31.30 Example flow of SCI initialization in Asynchronous mode with FIFO selected (1 of 2)

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set CCR0.TEIE, TIE, RIE, TE, RE to 0. If you have not changed from the initial settings, you can skip this step.
3	Set CCR3	Set up following function and communication mode. Driver control function for RS-485, FIFO use, Multi-Processor mode, Communication mode (MOD[2:0] = 000b) Transmission / reception format Clock enable (Leave the initial value when outputting the clock) Leave unused bits at their initial values.
4	Set CCR2	Set up the bit-rate-modulation function*1 *2, select clock, set bit rate*2
5	Set CCR1	Set up the Noise-filter function, the loop-back function, communication pin status, the parity check function, and the CTSn_RTSn function.
6	Set CCR4	Set up the adjust sampling timing function and the adjust transmit timing function.
7	Set FCR	Set the TFRST and RFRST to 1 to empty FIFO. Set the DRES, TTRG[4:0], RTRG[4:0], and RSTRG[4:0] bits

Table 31.30 Example flow of SCI initialization in Asynchronous mode with FIFO selected (2 of 2)

No.	Step Name	Description
8	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
9	Set CCR3	If you select a clock output in Asynchronous mode, set the CKE[1:0] bit here. After setting the register, the clock pin will be in the output state immediately. But the clock operates after setting TE or RE to 1.
10	Set CFCLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC If it's initialization flow after a reset, you can skip this step.
11	Set FFCLR	Write 1 to The FFCLR.BRKC, DRC and clear the corresponding flag. If it's initialization flow after a reset, you can skip this step.
12	Set CCR0	Set the TE or RE bit to 1. To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Setting the TE and RE bits allows TXDn and RXDn to be used.
13	Initialization completed	—

Note 1. If you do not use the bit rate modulation function, you do not need to set it.
Note 2. If you use an external clock, you do not need to set it.

Figure 31.11 shows an example of the timing when data is transmitted after reset is released, and SCI is set to Asynchronous mode according to Table 31.29 or Table 31.30. As shown in the figure, when the pin function is set to the TXDn pin, the CCR0.TE bit is 0, so the pin is high impedance. When transmit data is written after setting the CCR0.TE bit to 1, data transmission starts. There is a transmission wait time from writing TDR to data transmission starts. In Asynchronous mode, TXDn is high during this period.

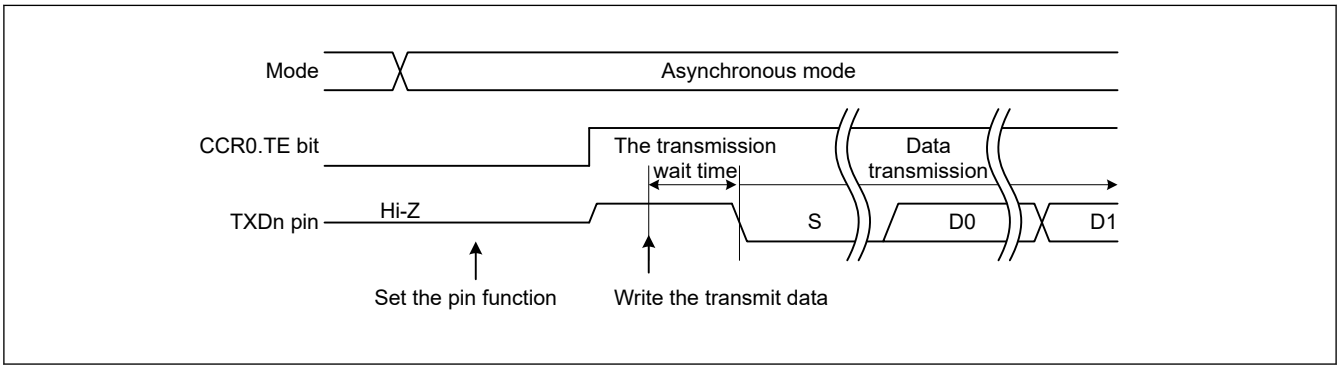


Figure 31.11 Data transmission timing example in Asynchronous mode

31.3.8 Serial Data Transmission in Asynchronous Mode

(1) Non-FIFO selected

Figure 31.12, Figure 31.13, Figure 31.14 and Figure 31.15 show examples of serial transmission in Asynchronous mode.

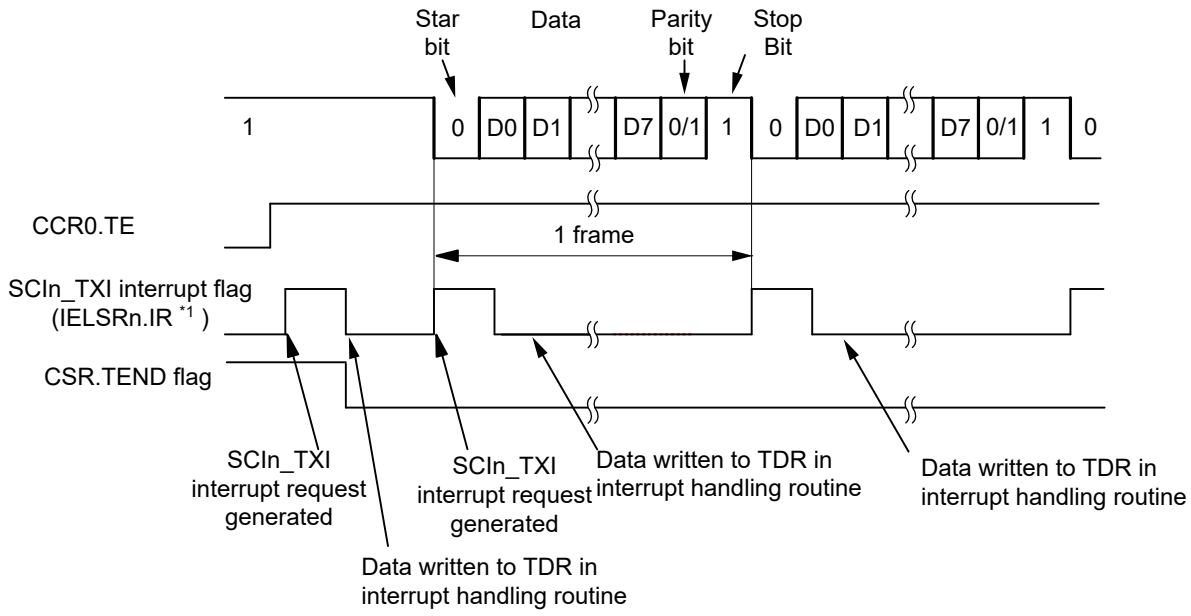
In serial transmission, the SCI operates as described in this section.

1. The SCI transfers data from the TDR register to the TSR register when data is written to TDR in the SCIn_TXI interrupt handling routine.
   The SCIn_TXI interrupt request at the beginning of transmission is generated when the CCR0.TE and CCR0.TIE bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the CCR1.CTSE bit is set to 0 (CTS function is disabled) or a low level on the CTSn_RTSn pin causes data transfer from the TDR register to the TSR register. If the CCR0.TIE bit is 1, an SCIn_TXI interrupt request is generated. Continuous transmission is possible by writing the next transmit data to the TDR register in the SCIn_TXI interrupt handling routine before transmission of the current transmit data is complete. When SCIn_TEI interrupt requests are in use, set the CCR0.TIE bit to 0 (SCIn_TXI interrupt requests are disabled) and the CCR0.TEIE bit to 1 (an SCIn_TEI interrupt request is enabled) after the last of the data to be transmitted is written to the TDR register from the handling routine for SCIn_TXI requests.
3. Data is sent from the TXDn pin in the following order:
   - Start bit

- Transmit data
- Parity bit or multi-processor bit (can be omitted depending on the format)
- Stop bit

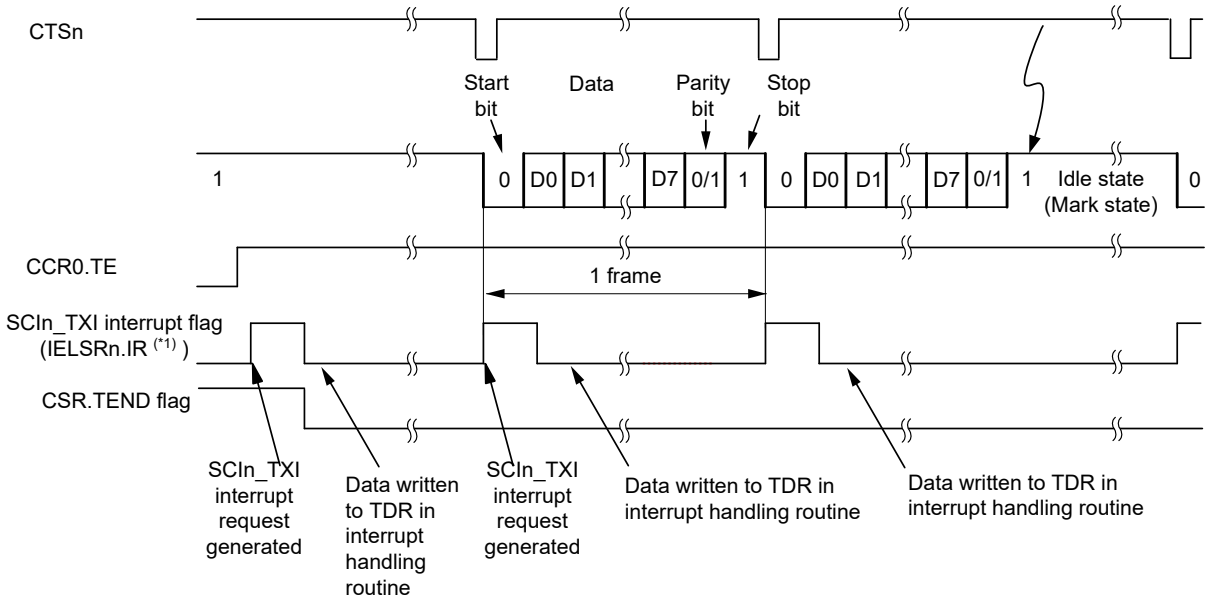
4. The SCI checks for update of the TDR register on output of the stop bit.
5. When the TDR register is updated, setting the CCR1.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn_RTSn pin causes transfer of the next transmit data from the TDR register to the TSR register and transmission of the stop bit, after which serial transmission of the next frame starts.
6. If the TDR register is not updated, the CSR.TEND flag is set to 1, the stop bit is sent, and the mark state is entered, in which 1 is output. If the CCR0.TEIE bit is 1, the CSR.TEND flag is set to 1 and an SCIn_TEI interrupt request is generated.

Figure 31.12, Figure 31.13, Figure 31.14 and Figure 31.15 show examples of serial transmission in Asynchronous mode.



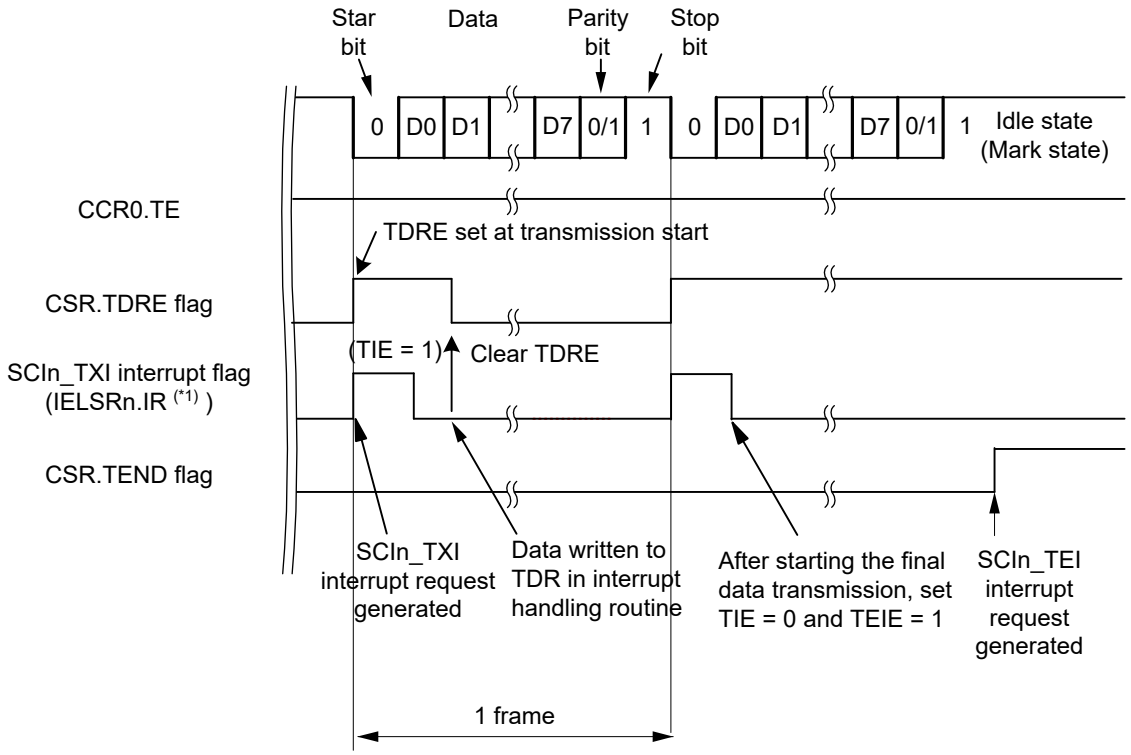
Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.12 Example operation for serial transmission in Asynchronous mode (1) with 8-bit data, parity bit, 1 stop bit, CTS function not used, and at the beginning of transmission



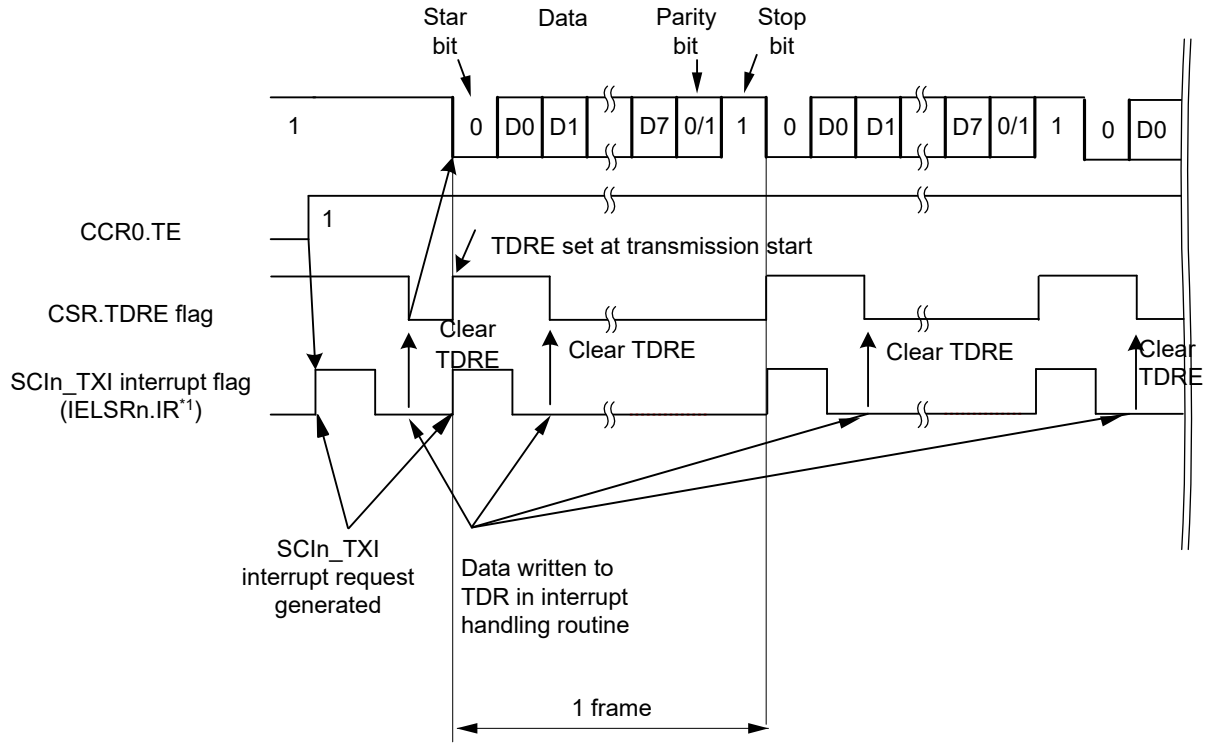
Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.13 Example operation for serial transmission in Asynchronous mode (2) with 8-bit data, parity bit, one stop bit, CTS function used, and at the beginning of transmission



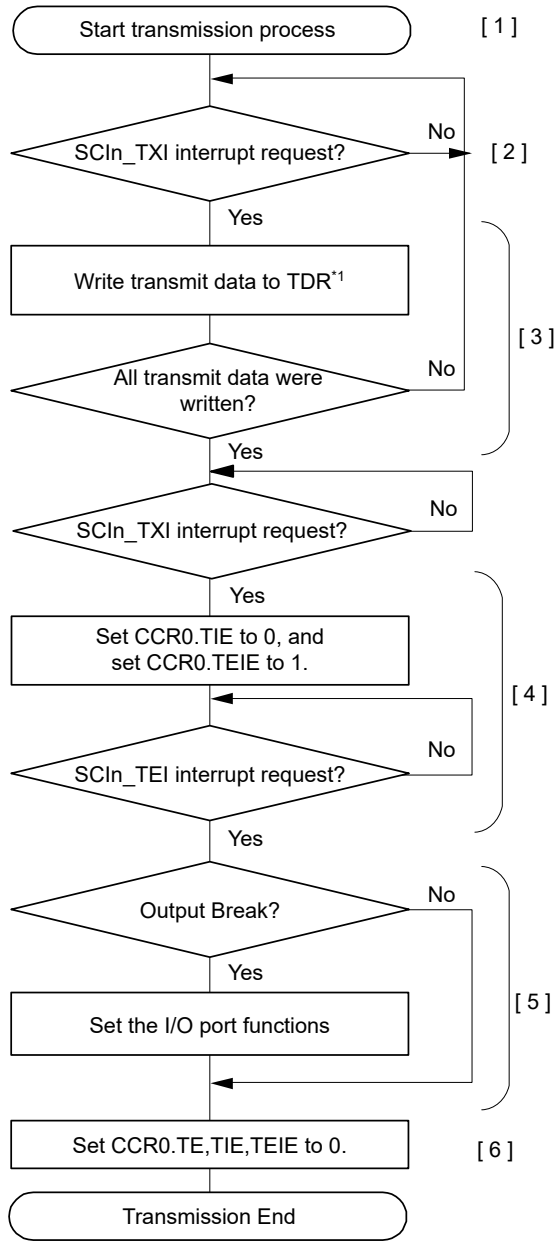
Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.14 Example operation for serial transmission in Asynchronous mode (3) with 8-bit data, parity bit, one stop bit, CTS function not used, and from the middle of transmission until transmission completion



Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.15 Example of Operation for Serial Transmission in Asynchronous Mode (4)(with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, from the Middle of Transmission until Transmission Completion)



[1] Start transmission process

[2] SCIn_TXI interrupt request:
The transmit-data-empty-interrupt(SCIn_TXI) is generated at following conditions.
When set TE and TIE to 0 at same time.
When transmission-data forwards to TSR from TDR.

[3] Serial transmission continuation procedure:
To continue serial transmission write transmit data to TDR*1 once by using a SCIn_TXI.
Transmit data can also be written by activating the DMAC or DTC.

[4] If you use TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR*1 and transmission is started.

[5] Break output at the end of serial transmission:
To output a break, it is necessary to set SPB2IO and SPB2DT, for output status setting when TE is 0.
(If you set these bits at initialization, you can skip this step.)

[6] Set CCR0.TE,TIE,TEIE to 0.

Note 1. In case of byte access with data length 9 bits, write transmit data in order of TDR[15:8] and TDR[7:0].

Figure 31.16 Example flow of serial transmission in Asynchronous mode with non-FIFO selected

(2) FIFO selected

Figure 31.17 shows an example of a data format that is written to TDR register in asynchronous mode with FIFO selected.

Data corresponding to the data length is set to TDR[8:0]. Write 0 for unused bits. Write in order from TDR[15:8] to TDR[7:0].

Data Length	Register setting		Transmit data in TDR[15:0]															
	CCR3.CHR[1:0]		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bit	1	1	–	–	–	–	–	–	MPBT	–	–	TDAT[6:0]						
8 bit	1	0	–	–	–	–	–	–	MPBT	–	TDAT[7:0]							
9 bit	0	Don't Care	–	–	–	–	–	–	MPBT	TDAT[8:0]								

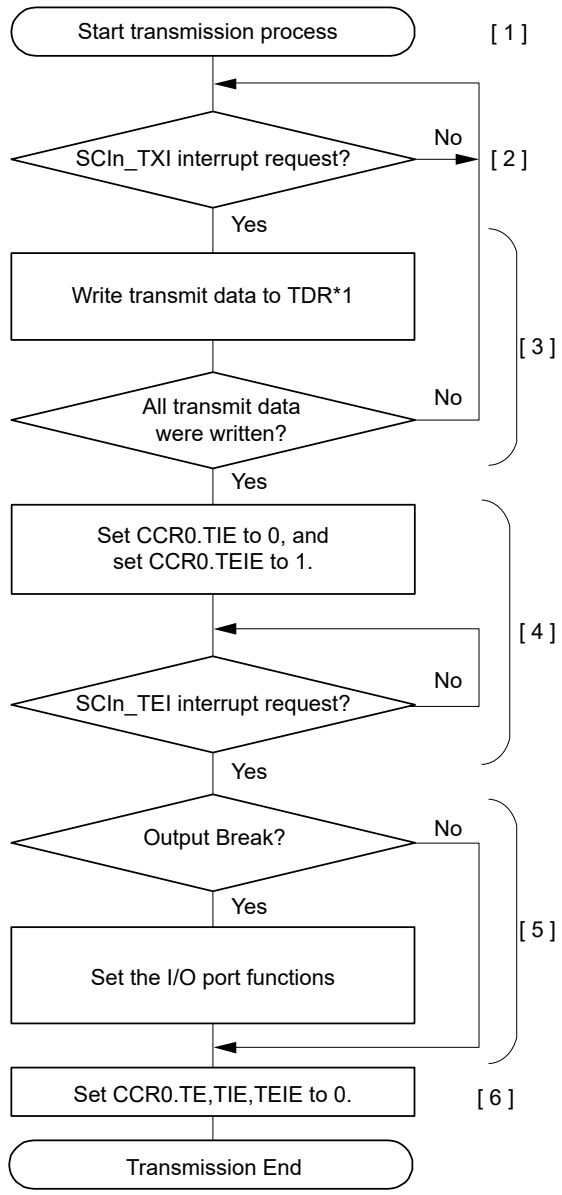
Note: —: Invalid. The write value should be 0.

Figure 31.17 Data format written to transmit-FIFO(TDR) with FIFO selected

In serial transmission, the SCI operates as described in this section.

1. The SCI transfers data from the TDR register to the TSR register when data is written to TDR in the SCIn_TXI interrupt handling routine. The amount of data that can be written to TDR is 16 minus FTSR.T[5:0] bytes. The SCIn_TXI interrupt request at the beginning of transmission is generated when the CCR0.TE and CCR0.TIE bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the CCR1.CTSE bit is set to 0 (CTS function is disabled) or a low level on the CTSn_RTSn pin causes data transfer from the TDR register to the TSR register. When the amount of transmit data written in TDR is equal to or less than the specified transmit triggering number, CSR.TDRE is set to 1. If the CCR0.TIE bit is 1, an SCIn_TXI interrupt request is generated. Continuous transmission is possible by writing the next transmit data to TDR in the SCIn_TXI interrupt handling routine before transmission of the current transmit data is complete. When SCIn_TEI interrupt requests are in use, set the CCR0.TIE bit to 0 (SCIn_TXI interrupt requests are disabled) and the CCR0.TEIE bit to 1 (an SCIn_TEI interrupt request is enabled) after the last of the data to be transmitted is written to the TDR register from the handling routine for SCIn_TXI requests.
3. Data is sent from the TXDn pin in the following order:
   - Start bit
   - Transmit data
   - Parity bit or multi-processor bit (can be omitted depending on the format)
   - Stop bit
4. On output of the stop bit, the SCI checks whether non-transmitted data remains in the TDR register.
5. When data is set to transmit-FIFO (TDR), setting the CCR1.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn_RTSn pin causes transfer of the next transmit data from TDR to TSR and transmission of the stop bit, after which serial transmission of the next frame starts.
6. If data is not set in transmit-FIFO (TDR), the TEND flag in CSR is set to 1, the stop bit is sent, and the mark state is entered in which 1 is output. If the CCR0.TEIE bit is 1, an SCIn_TEI interrupt request is generated.

Figure 31.18 shows an example flow of serial transmission in Asynchronous mode with FIFO selected.



[1] Start transmission process

[2] SCIn_TXI interrupt request:
The transmit-data-empty-interrupt(SCIn_TXI) is generated when transmission-data forwards to TSR from the transmit-FIFO(TDR) and the number of transmit-data written in the transmit-FIFO is equal to or less than the transmit-FIFO threshold.

[3] Serial transmission continuation procedure:
To continue serial transmission, write all transmit data to the transmit-FIFO(TDR) by using a SCIn_TXI interrupt and clear TDRE flag to 0. (Transmit data can also be written to the transmit-FIFO(TDR) by activating the DMAC or DTC. In this case, the TDRE flag is cleared automatically, so please don't operate the TDRE flag.)
The number of transmission data it's possible to write in is 16 - (the number of stored transmit FIFO data).

[4] If you use SCIn_TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR*1.

[5] Break output at the end of serial transmission:
To output a break, it is necessary to set SPB2IO and SPB2DT, for output status setting when TE is 0.
(If you set these bits at initialization, you can skip this step.)

[6] Set CCR0.TE,TIE,TEIE to 0.

Note 1. In case of byte access with data length 9 bits, write transmit data in order of TDR[15:8] and TDR[7:0].

Figure 31.18 Example flow of serial transmission in Asynchronous mode with FIFO selected

31.3.9 Serial Data Reception in Asynchronous Mode

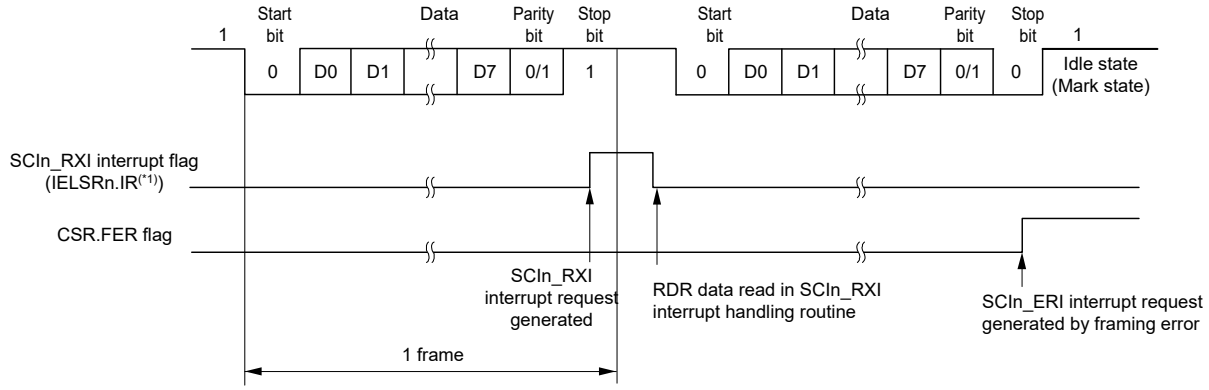
(1) Non-FIFO selected

Figure 31.19 and Figure 31.20 show an example of the operation for serial data reception in Asynchronous mode.

In serial data reception, the SCI operates as follows:

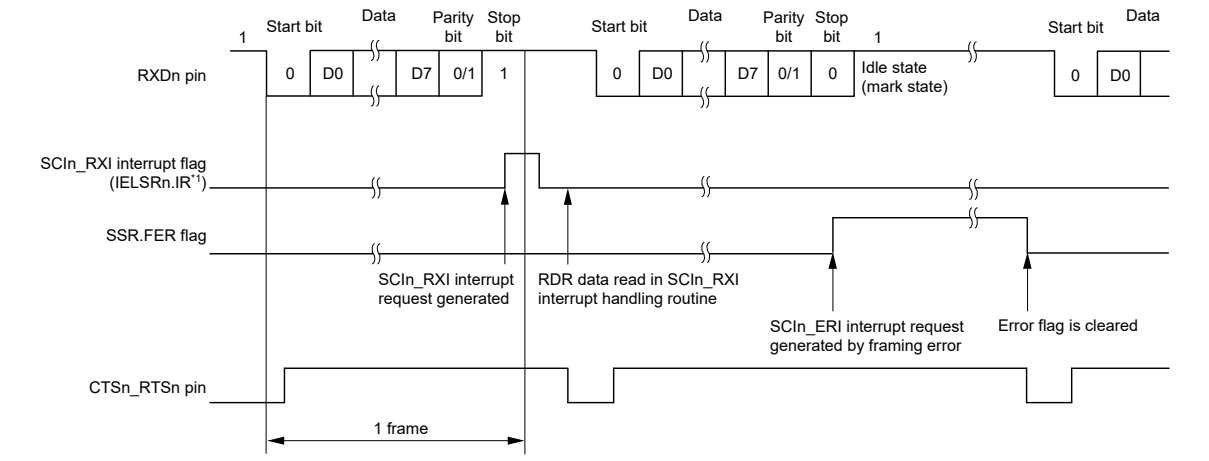
1. When the value of the CCR0.RE bit becomes 1, the output signal on the CTSn_RTSn pin goes low.
2. The SCI monitors the communications line and when it detects a start bit, the SCI performs internal synchronization, stores receive data in RSR, and checks the parity bit and stop bit.
3. If an overrun error occurs, the CSR.ORER flag is set to 1. If the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to the RDR register.
4. If a parity error is detected, the CSR.PER flag is set to 1 and receive data is transferred to the RDR register. If the SCR.RIE bit is 1, an SCIn_ERI interrupt request is generated.

5. If a framing error is detected, the CSR.FER flag is set to 1 and receive data is transferred to the RDR register. If the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to the RDR register. If the CCR0.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register in the SCIn_RXI interrupt handling routine before reception of the next receive data is complete. Reading the received data that was transferred to the RDR register causes the CTSn_RTSn pin to output low.



Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.19 Example of SCI operation for serial reception in Asynchronous mode (1) when the RTS function is not used, and with 8-bit data, parity bit, and 1 stop bit



Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.20 Example of SCI operation for serial reception in Asynchronous mode (2) when RTS function is used, and with 8-bit data, parity bit, and 1 stop bit

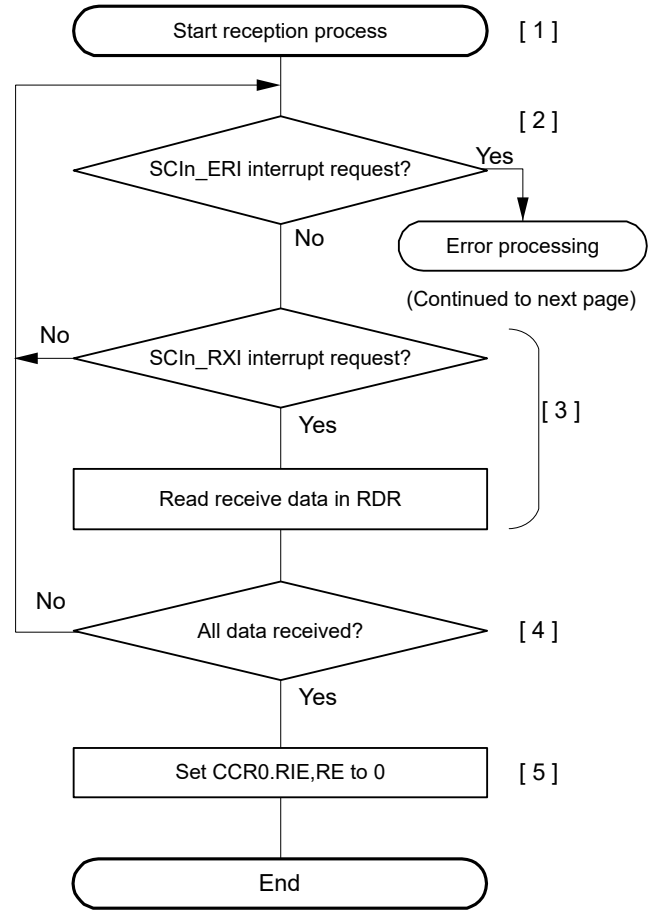
Table 31.31 lists the states of the flags in the CSR status register and receive data handling when a receive error is detected.

If a receive error is detected, an SCIn_ERI interrupt request is generated but an SCIn_RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, and PER bits to 0 before resuming reception. In addition, be sure to read the RDR register during overrun error processing. When a reception is forced to terminate by setting the CCR0.RE bit to 0 during operation, read the RDR register because received data that is not yet read might be left in the RDR.

Figure 31.21 and Figure 31.22 show example flows of serial data reception.

Table 31.31 Flags in CSR Status Register and receive data handling

Flags in the CSR Status Register			Receive data	Receive error type
ORER	FER	PER		
1	0	0	Lost	Overrun error
0	1	0	Transferred to RDR	Framing error
0	0	1	Transferred to RDR	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error



[1] Start reception process

[2] Receive error processing and break detection: If a receive error occurs, an SCIn_ERI interrupt is generated. After performing the appropriate error processing, be sure to set the ORER, PER, and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RXDn pin.

[3] Read the receive data in RDR*1 once in the SCIn_RXI interrupt handling routine.

[4] Serial reception continuation procedure: To continue serial reception, before the stop bit of the current frame is received, read data from RDR in the SCIn_RXI interrupt handling routine. The RDR data can also be read by activating the DMAC or DTC.

[5] Set CCR0.RIE,RE to 0.

Note 1. Do not set RE to 0 before reading RDR.

Figure 31.21 Example flow of serial reception in Asynchronous mode with non-FIFO selected and Address Matching Disabled (1)

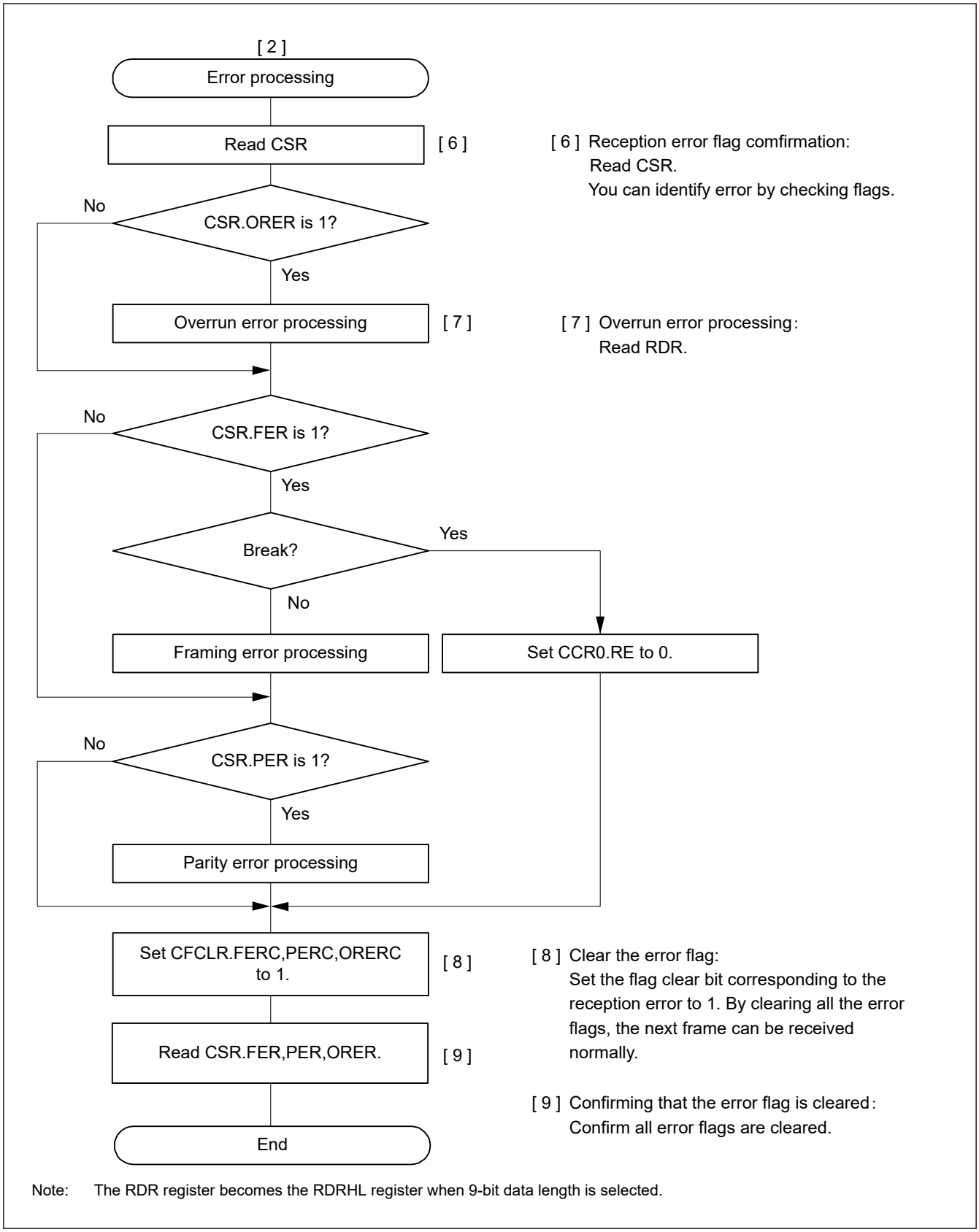


Figure 31.22 Example flow of serial reception in Asynchronous mode with non-FIFO selected and Address Matching Disabled (2)

(2) FIFO selected

Figure 31.23 shows an example of a data format that is written to Receive-FIFO(RDR) register in Asynchronous mode.

In Asynchronous mode, 0 is written to the MPB bit in the RDR register. Data that corresponds to the data length is written to RDR. Unused bits are written as 0. If software reads RDR, the SCI updates FER, PER, and receive data (RDAT[8:0]) in the RDR register with the next data. The flags ORER, and DR in the RDR register always reflect the associated flags in the CSR register.

Data Length	Register Setting		Receive flag in RDR[31:0], MPB, RDAT[8:0]															
	CCR3.CHR[1:0]		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7bit	1	1	-	-	-	FFER	FPER	DR	MPB	0	0	RDAT[6:0]						
8bit	1	0	-	-	-	FFER	FPER	DR	MPB	0	RDAT[7:0]							
9bit	0	Don't care	-	-	-	FFER	FPER	DR	MPB	RDAT[8:0]								
	CCR3.CHR[1:0]		b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
7bit	1	1	-	-	-	FER	PER	-	-	ORER	-	-	-	-	-	-	-	-
8bit	1	0	-	-	-	FER	PER	-	-	ORER	-	-	-	-	-	-	-	-
9bit	0	Don't care	-	-	-	FER	PER	-	-	ORER	-	-	-	-	-	-	-	-

Note: 0 is always read from the MPB flag (RDR [9] bit).
When a 7-bit data length is selected, 0 is read from the RDAT [8:7] bits.
When 8-bit data length is selected, 0 is read from the RDAT [8] bit.

Figure 31.23 Data format stored in receive-FIFO(RDR) with FIFO selected

Table 31.32 lists the states of the flags in CSR status register and receive data handling when a receive error is detected with FIFO selected. Figure 31.24 and Figure 31.25 show samples of flowcharts for serial data reception with FIFO selected.
In serial data reception, the SCI operates as follows:

1. When the value of the CCR0.RE bit becomes 1, the output signal on the CTSn_RTSn pin goes low.
2. The SCI monitors the communications line and, when it detects a start bit, the SCI performs internal synchronization, stores receive data in the RSR register.
3. If an overrun error occurs during normal communications, the CSR.ORER flag is set to 1. If the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to the RDR register.
4. If a parity error is detected, the PER flag and receive data are transferred to the RDR register. If the CCR0.RIE bit is set to 1, an SCIn_ERI interrupt request is generated.
5. If a framing error is detected, the FER flag and receive data are transferred to the RDR register. If the CCR0.RIE bit is set to 1, an SCIn_ERI interrupt request is generated.
6. After a framing error is detected and when SCI detects that the continuous receive data is zero for one frame, reception stops.
7. When the amount of data stored in the RDR register falls below the specified receive triggering number, and the next data is not received after 15 etus from the last stop bit in Asynchronous mode, the FRSR.DR flag is set to 1. When the CCR0.RIE bit is 1 and the FCR.DRES bit is 0, the SCI generates an SCIn_RXI interrupt request. When the FCR.DRES bit is 1, SCI generates an SCIn_ERI interrupt request.
8. When reception finishes successfully, receive data is transferred to the RDR register. The RDRF bit is set to 1 when the amount of receive data written to RDR is equal to or greater than the specified receive triggering number. If the CCR0.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register in the SCIn_RXI interrupt handling routine, before an overrun error occurs. If the received data that is transferred to RDR is less than the RTS trigger number, the CTSn_RTSn pin outputs low.

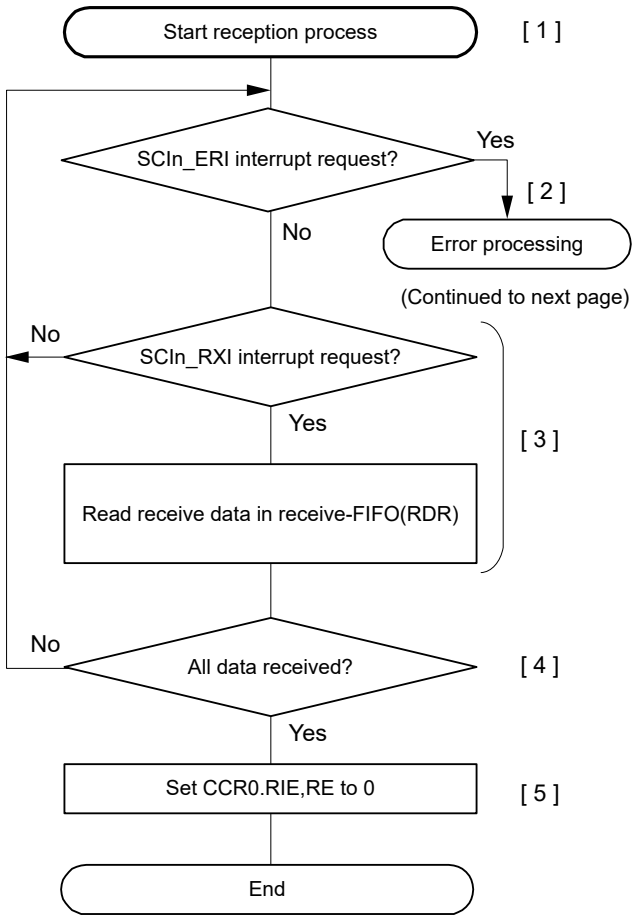
Table 31.32 Flags in the CSR Status Register and Receive Data Handling (FIFO selected) (1 of 2)

CSR value			Receive-FIFO (RDR)	Receive Error Type
ORER	FER*1	PER*1	RDAT[8:0]	
1	0	0	Lost	Overrun error
0	1	0	Transferred RDR	Framing error
0	0	1	Transferred RDR	Parity error

Table 31.32 Flags in the CSR Status Register and Receive Data Handling (FIFO selected) (2 of 2)

CSR value			Receive-FIFO (RDR)	Receive Error Type
ORER	FER*1	PER*1	RDAT[8:0]	
1	1	0	Lost	Overrun error + Framing error
1	0	1	Lost	Overrun error + Parity error
0	1	1	Transferred RDR	Framing error + Parity error
1	1	1	Lost	Overrun error + Framing error + Parity error

Note 1. This flag indicates whether there is an error in received data when reception is completed.



[1] Start reception process

[2] Receive error processing and break detection: If a receive error occurs, an SCIn_ERI interrupt is generated. After performing the appropriate error processing, be sure to set the ORER flag*1 to 0. Reception cannot be resumed if ORER is set to 1. Reception can continue even if PER,FER,DR flags*1 are 1.

[3] Read the receive data in the receive-FIFO(RDR) in the SCIn_RXI interrupt handling routine. The reception data stored in the receive-FIFO(RDR) is read until the number of stored data will be below the FCR.RTRG value at least. You can confirm the number of the reception data in the FIFO by FRSR.R[5:0].

[4] Serial reception continuation procedure: To continue serial reception, before an overrun error occurs, please read data from the receive-FIFO(RDR) in the SCIn_RXI interrupt handling routine and clear CSR.RDRF flag and FRSR.DR flag to 0. The receive-FIFO data can also be read by activating the DMA or DTC. The RDRF flag is cleared automatically in this case, so please don't operate the RDRF flag.

[5] Set CCR0.RIE,RE to 0.

Note 1. Can be read in the RDR.ORER, PER, FER and DR flags.

Figure 31.24 Example flow of serial reception in Asynchronous mode with FIFO selected and Address Matching Enabled (1)

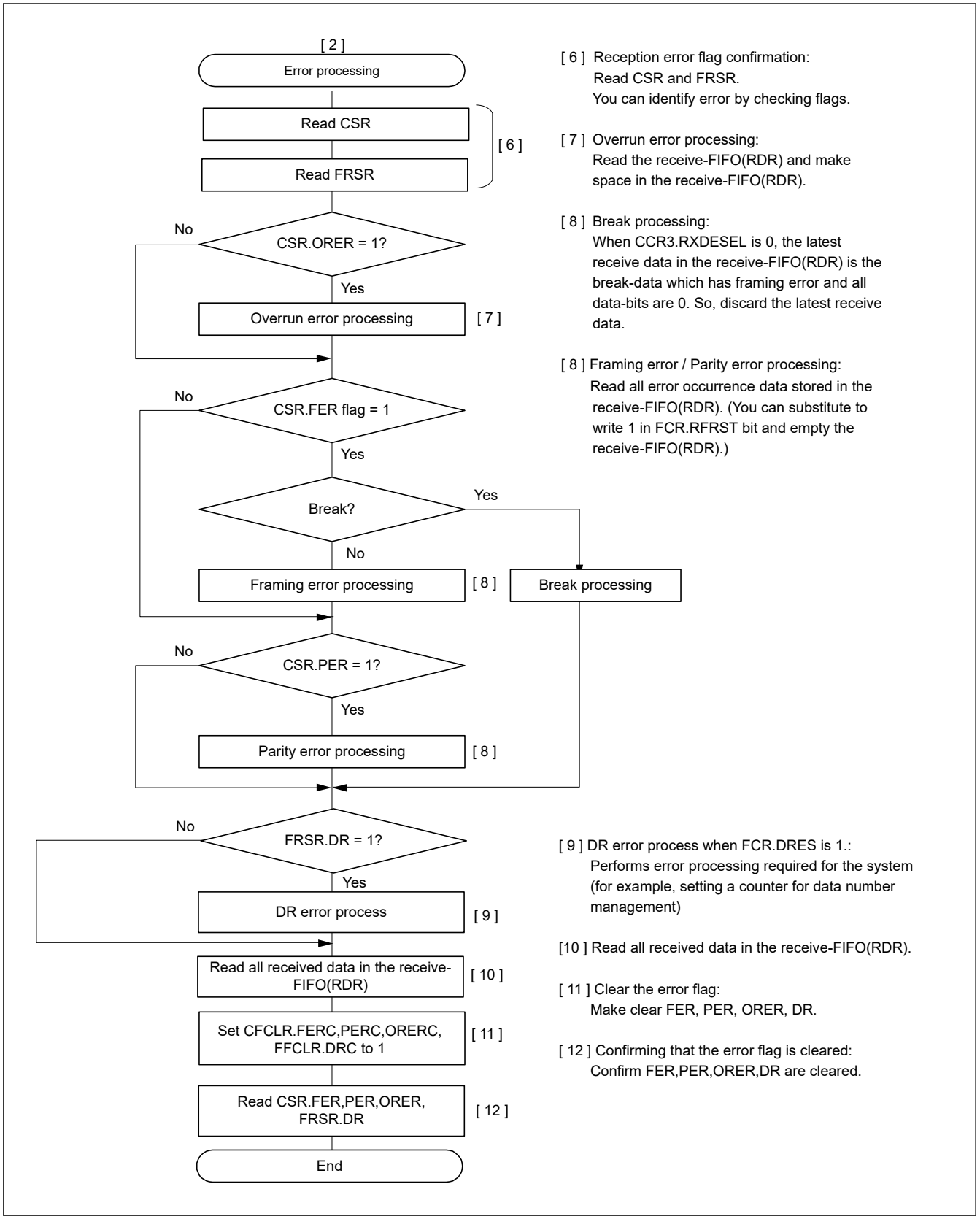


Figure 31.25 Example flow of serial reception in Asynchronous mode with FIFO selected Address Matching Disabled (2)

31.3.10 The function of adjust receive sampling timing (Asynchronous Mode)

When there is the difference between the rising transfer time and the falling transfer time through a photo coupler, the receive sampling timing at middle of bit affects the reception margin. In this case, the receive sampling timing is able to adjust from the middle of bit to the optimum timing by using this function.

The receive sampling timing is adjusted from the middle of bit by following formula. The adjustable direction is set by CCR4.AJD. When adjusting backward (CCR4.AJD = 0), substitute AJD = +1 and substitute AJD = −1 when adjusting forward (CCR4.AJD = 1).

Adjusted sampling timing = the middle of bit + AJD × (base clock × the setting value of CCR4.AST[2:0])

The setting timing is limited by base clock cycles per 1 bit. For details, see Table 31.33.

An overview of reception operation of the communication through a photo coupler with this function is shown in Figure 31.26, Figure 31.27 and Figure 31.28, the explanation of operation with this function is shown in Figure 31.29.

Do not use this function when there is no difference between the rising transfer time and the falling transfer time, because there is a possibility of deteriorating the reception margin.

Table 31.33 The acceptable value of setting register (Asynchronous mode using internal clock)

CCR2.ABCSE2	CCR2.ABCSE	CCR2.ABCS	The number of base cycles/1bit	The acceptable value	
				CCR4.AJD	CCR4.AST*1
1	0	x	4	0	000 to 011
				1	(Note1)
0	1	x	6	0	000 to 010
				1	(Note1)
0	0	1	8	0	000 to 011
				1	(Note1)
0	0	0	16	0	000 to 111
				1	

Note: x: Don't care
Note 1. When the value of CCR4.AST exceeds the acceptable value, sampling is done at default timing. (Adjustment of sampling is not done.)

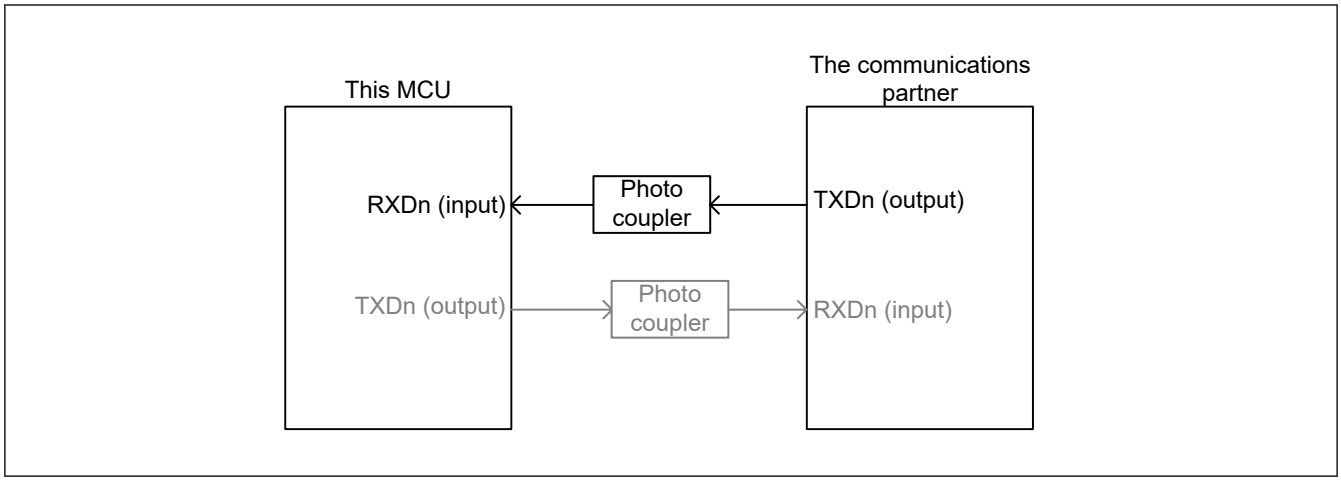
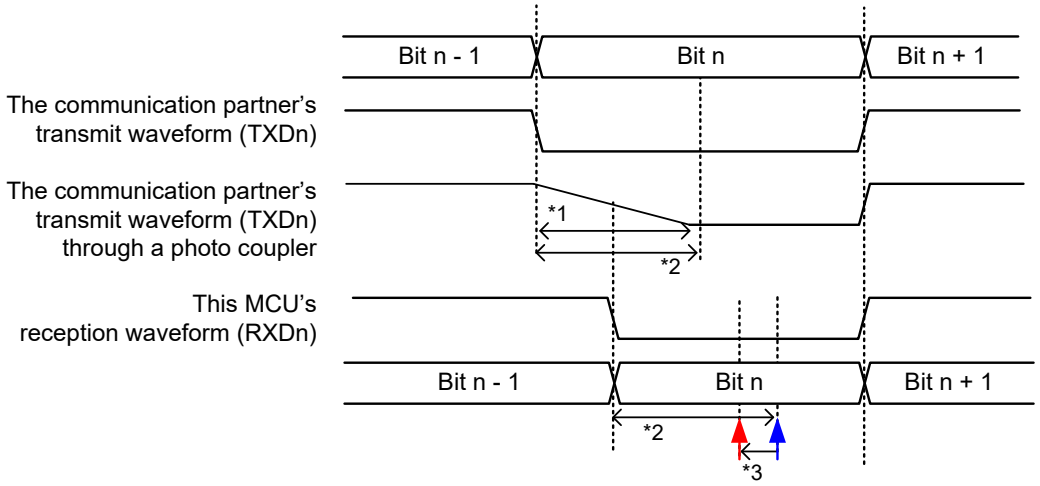


Figure 31.26 block diagram image of the reception through a photo couple

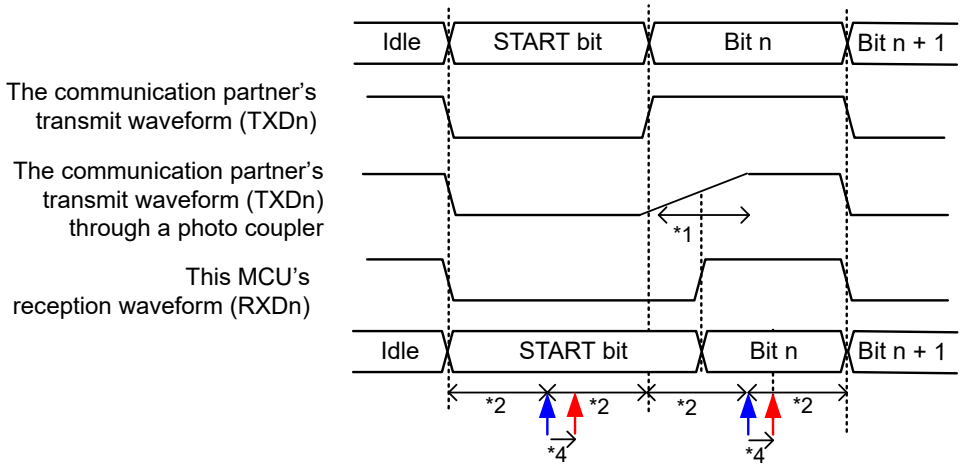
(a) In the case of the falling transfer time >> rising transfer time

The falling edge of reception waveform is made dull like following chart.
In this case, you can sampling at the middle of bit if you adjust the receive sampling timing to forward (AJD = 1).



(b) In the case of the falling transfer time << rising transfer time

The rising edge of reception waveform is made dull like following chart.
So, the reception margin of communications partner will be bad. In this case, you can improve the reception margin if you adjust the receive sampling timing to back.



The receive sampling timing when unadjusted (middle of bit)
The adjusted receive sampling timing

Note: This waveform shows the operation image of adjustment in reception sampling timing.
Note 1. The dull period by a photo coupler
Note 2. Bit center timing at set communication rate
Note 3. When CCR4.AJD is 1, the receive sampling timing is shifted to forward by the setting value of CCR4.AST[2:0].
Note 4. When CCR4.AJD is 0, the Receive sampling timing is shifted to backward by the setting value of CCR4.AST[2:0].

Figure 31.27 Overview of reception operation of the communication through a photo coupler

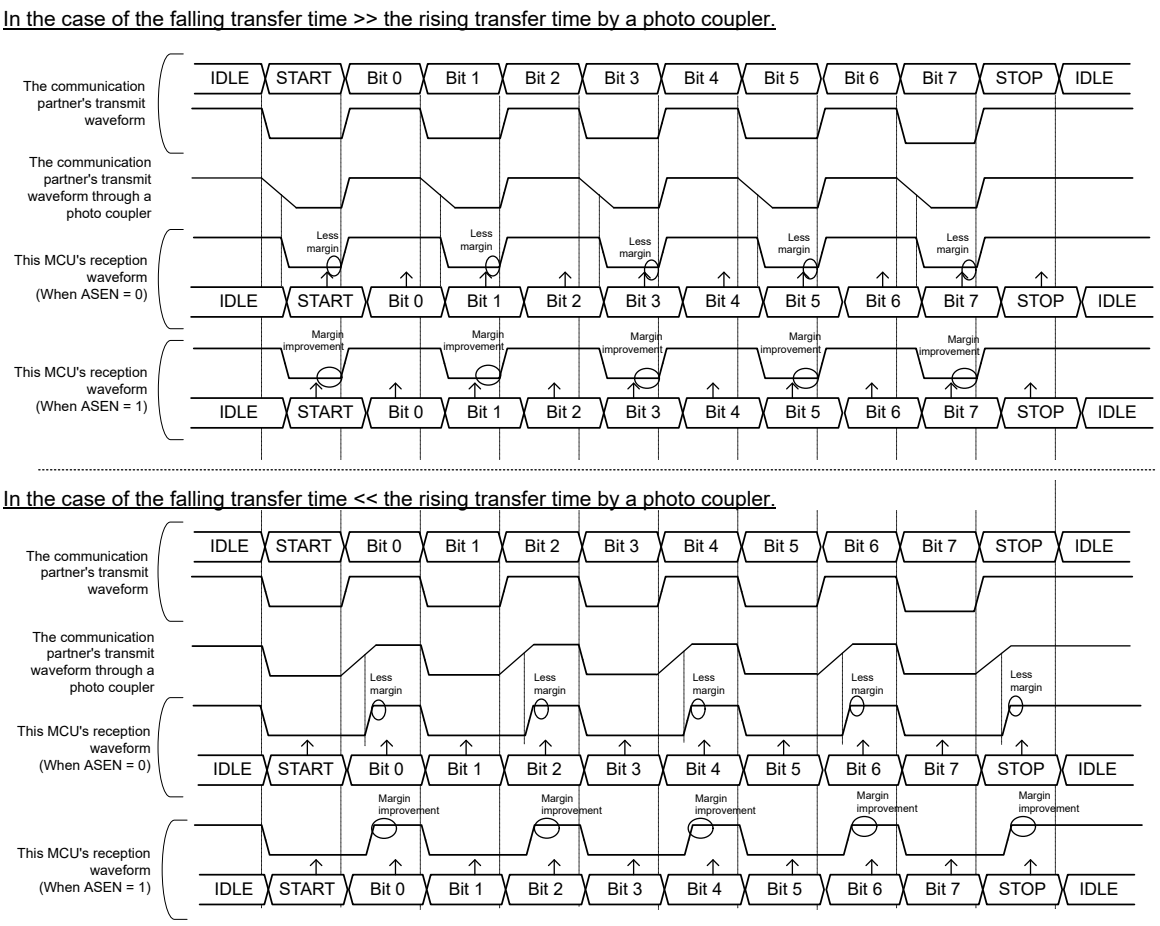


Figure 31.28 Example of improvement in reception margin by the reception sampling timing adjustment function

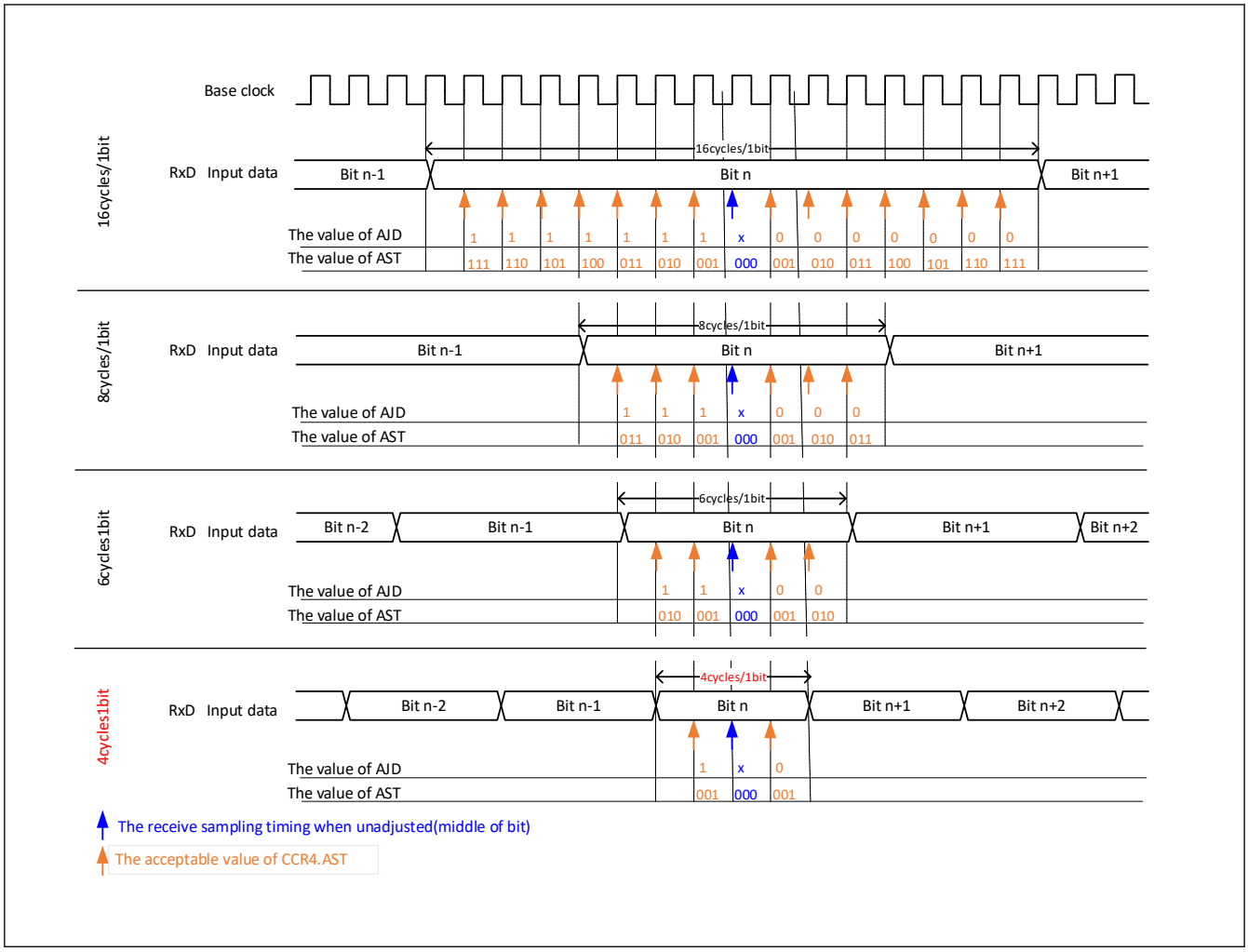


Figure 31.29 Overview of the adjustment operation for the reception sampling timing (Asynchronous mode using internal clock)

31.3.11 The function of adjust transmit timing (Asynchronous Mode)

In communication via a photo coupler or the like, when either the rising or falling transition time of the TXDn output signal is long, then a communication partner receive dulled waveform. In this case, the reception margin may be affected.

In these cases, make a communication partner to be sampling at middle of bit using the function of adjust transmit timing.

When CCR4.ATEN is 1, this function can adjust the edge timing at the timing calculated by the following formula for the edge set with CCR4.AET.

The adjustment edge timing = the base clock × CCR4.ATT[2:0]

In addition, the upper limit of the adjustment edge timing is limited by setting the base clock cycles. For details, see Table 31.34.

A transmission movement image figure of the communication through a photo coupler with this function is shown in Figure 31.30, Figure 31.31 and Figure 31.32, the overview of operation with this function is shown in Figure 31.33 and Figure 31.34.

Do not use this function when there is not the difference between the rising transfer time and the falling transfer time, there is a possibility of deteriorating the reception margin of a communication partner.

Table 31.34 The acceptable value of CCR4.AET and CCR4.ATT (Asynchronous mode using internal clock)

ABCSE2	ABCSE	ABCS	The number of base clock cycles/1bit	The acceptable value	
				AET	ATT [2:0]
1	0	x	4	0	000 to 011
				1	
0	1	x	6	0	000 to 101
				1	
0	0	1	8	0	000 to 111
				1	
0	0	0	16	0	000 to 111
				1	

Note: x: Don’t care
Note: When the value of ACTR.AET/ATT is out of the acceptable value, this SCI module doesn’t adjust transmit timing.

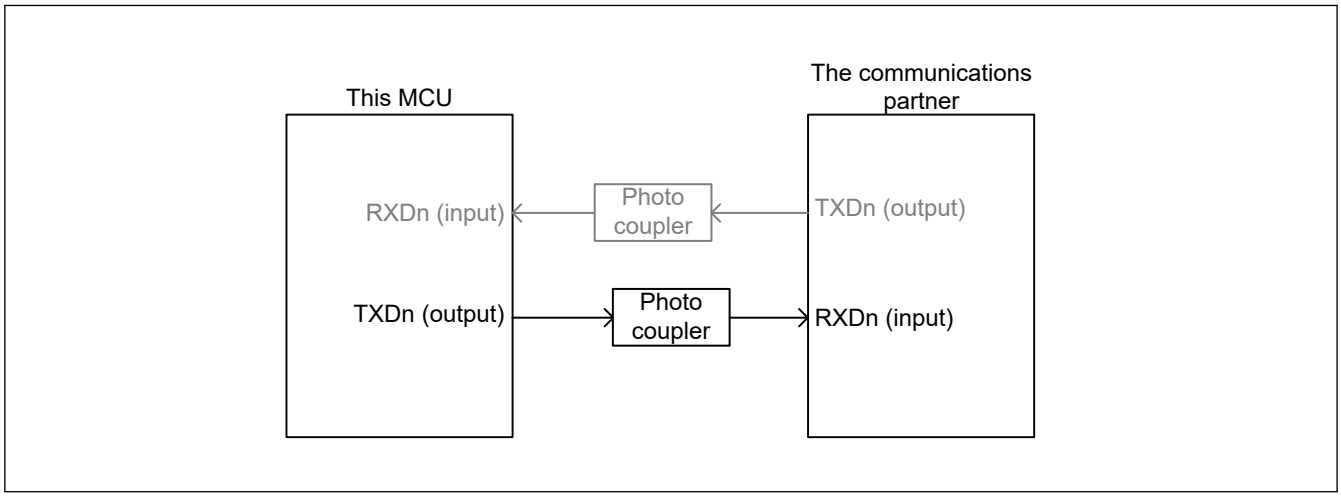
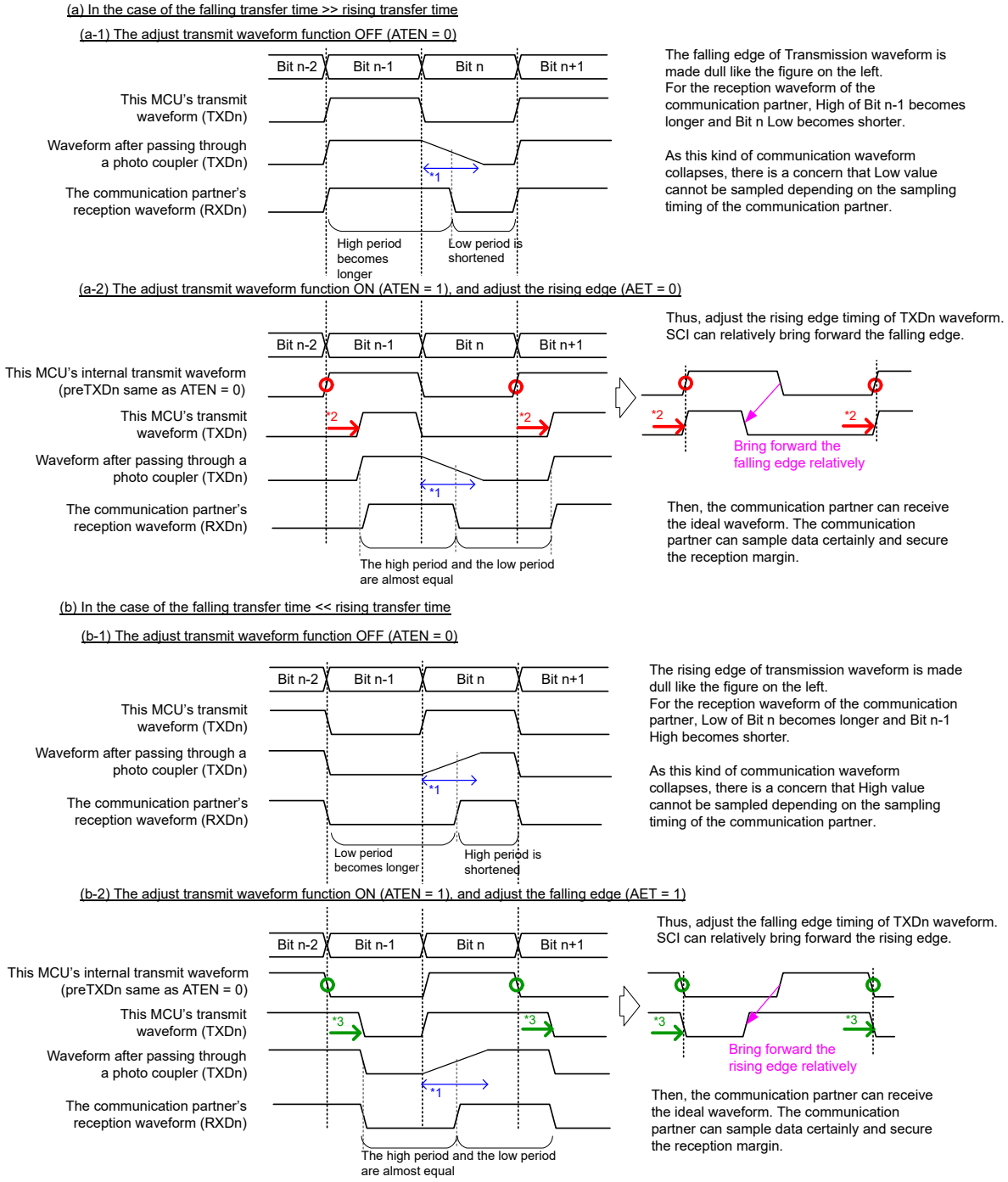


Figure 31.30 block diagram image of the transmission through a photo coupler



Note: This waveform shows the operation image of adjustment transmit waveform.
Note 1. The dull period by a photo coupler
Note 2. When CCR4.AET is 0, the rising edge of transmit waveform timing is shifted to back by the setting value of CCR4.ATT[2:0]. This MCU transmit waveform bringing forward the falling edge relatively.
Note 3. When CCR4.AET is 1, the falling edge of transmit waveform timing is shifted to back by the setting value of CCR4.ATT[2:0]. This MCU transmit waveform bringing forward the rising edge relatively.

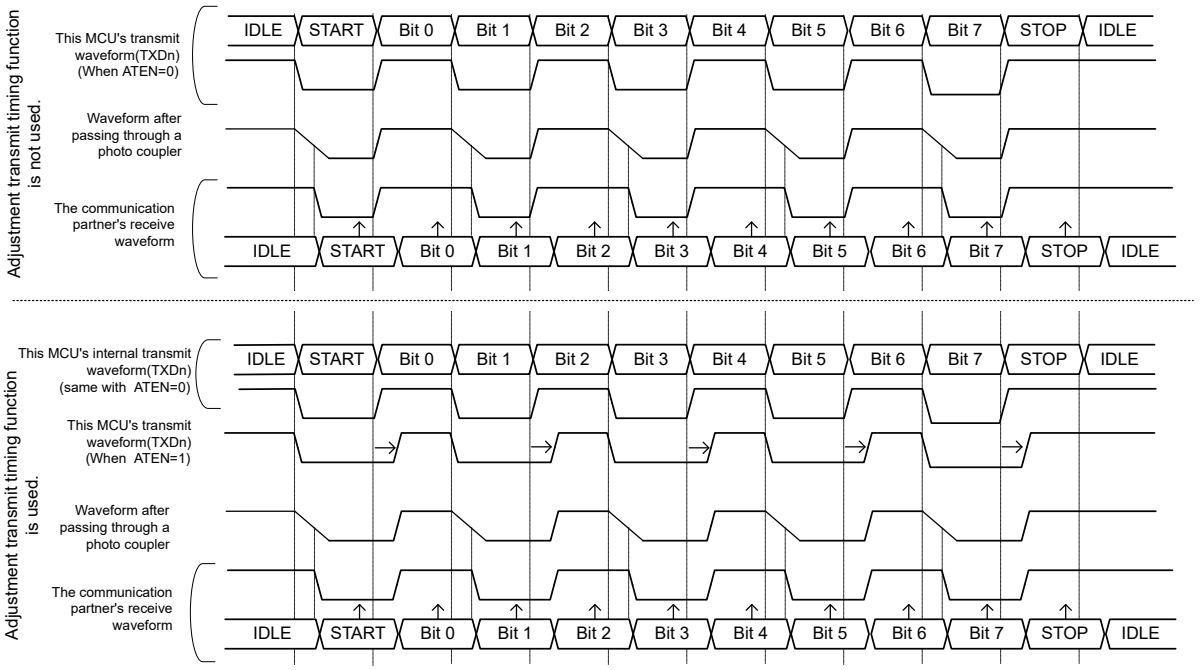
Figure 31.31 The overview of transmission operation in the communication through a photo coupler

The explanation of transmit waveforms of the communication through a photo coupler using adjust transmit timing function

When using the transmission timing adjustment function, adjust the edge timing of the transmission waveform and correct the reception waveform of the communication partner

The following example is 8 bit long data.

(a) In the case of the falling edge transfer time >> the rising transfer time



(b) In the case of the falling edge transfer time << the rising transfer time

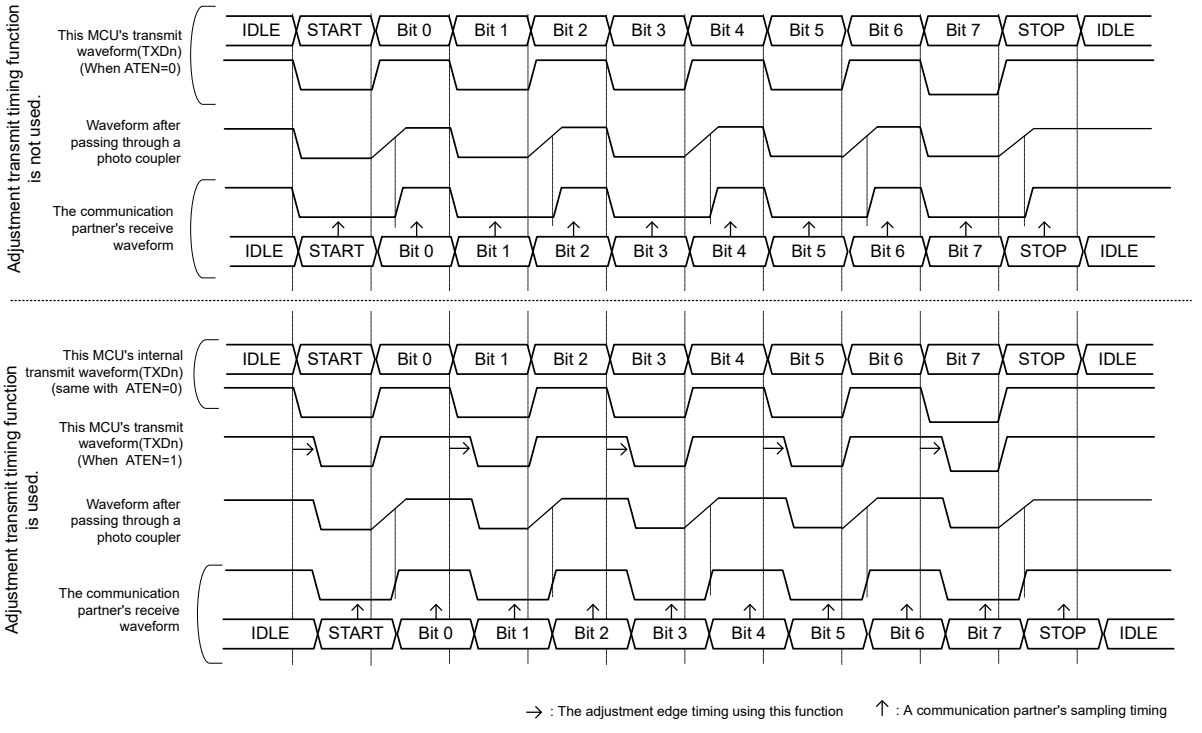
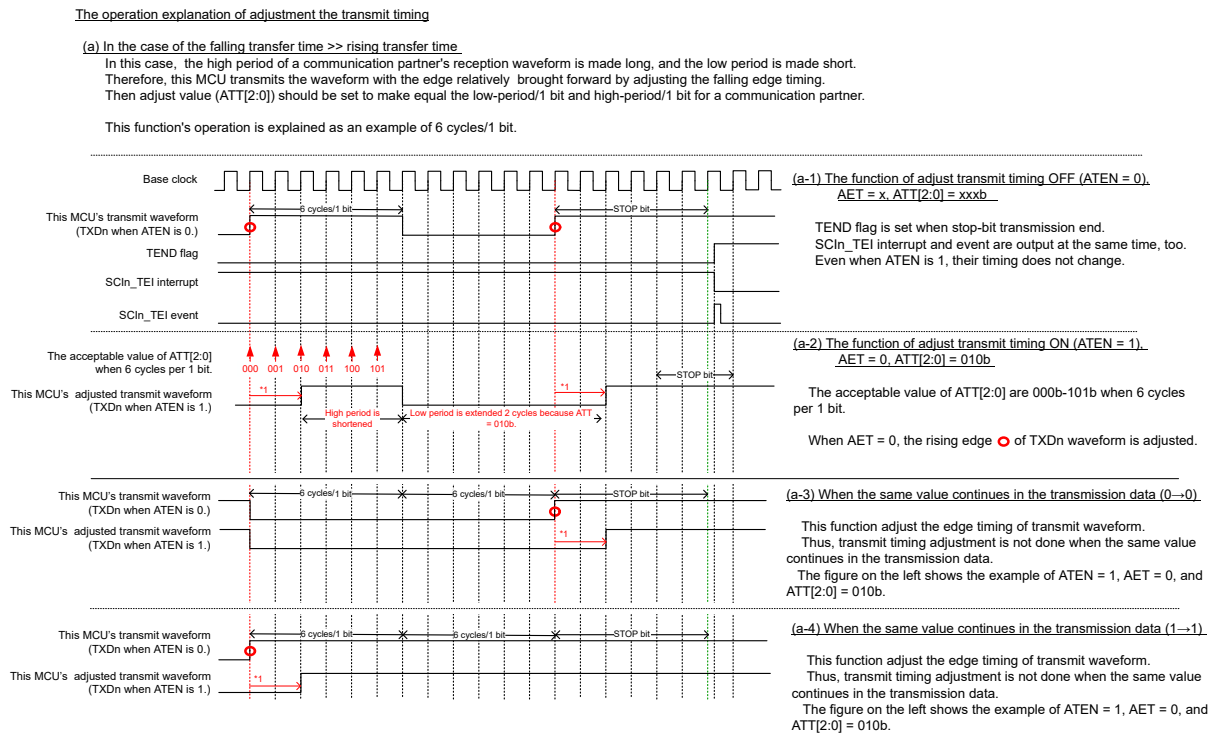
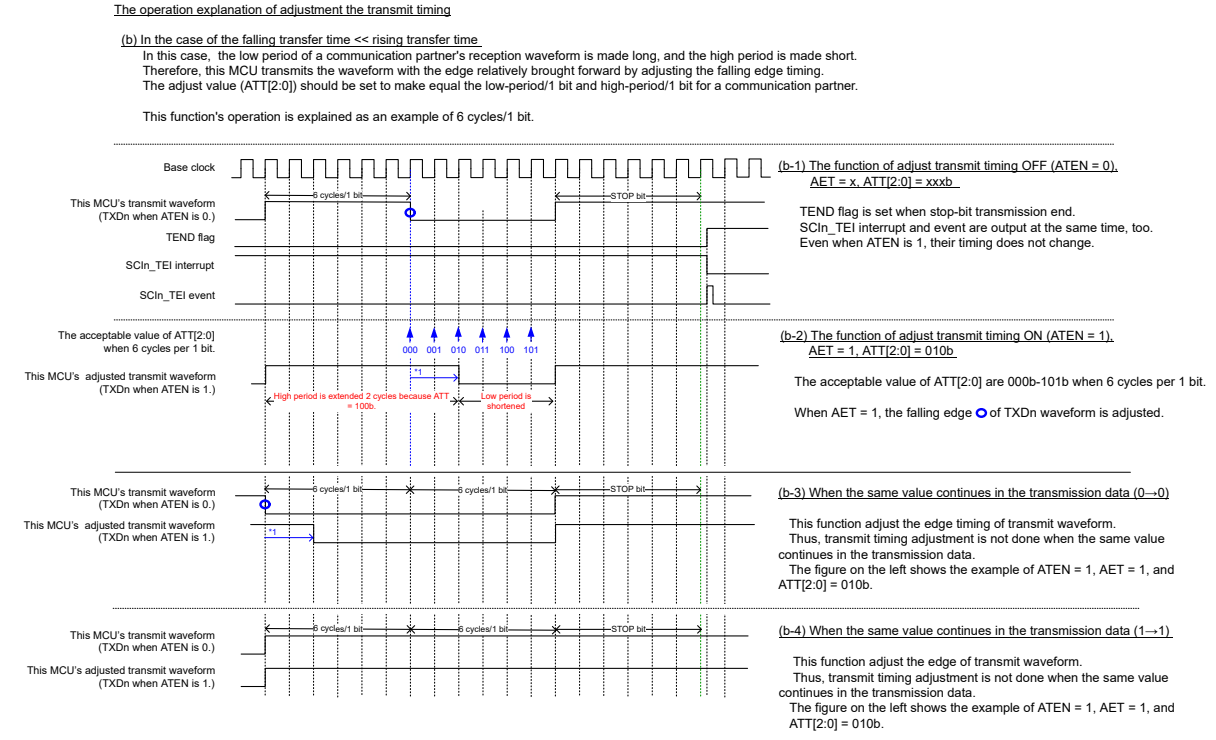


Figure 31.32 The explanation for the transmit waveform through a photo coupler



Note 1. The rising edge of transmit timing is shifted to back by the setting value of CCR4.ATT[2:0].

Figure 31.33 The adjustment operation explanation for the transmit timing when AET is 0



Note 1. The falling edge of transmit timing is shifted to back by the setting value of CCR4.ATT[2:0].

Figure 31.34 The adjustment operation explanation for the transmit timing when AET is 1

31.4 Multi-Processor Communication Function

The multi-processor communication function enables the SCI to transmit and receive data between multiple processors by sharing an asynchronous serial communication line that has an added multi-processor bit. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station.

The multi-processor bit is used to distinguish between the ID transmission cycle and the data transmission cycle:

- When the multi-processor bit is set to 1, the transmission cycle is the ID transmission cycle
- When the multi-processor bit is set to 0, the transmission cycle is the data transmission cycle

Figure 31.35 shows an example of communication between processors using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits communication data in which the multi-processor bit set to 0 is added to the transmit data. After receiving communication data with the multi-processor bit set to 1, the receiving station compares the received ID with the ID of the receiving station itself. If the two match, the receiving station receives communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until it receives data in which the multi-processor bit is set to 1.

RTS control cannot be used at the time of multi-processor communication function use because this is a function corresponding to one-to-many communications.

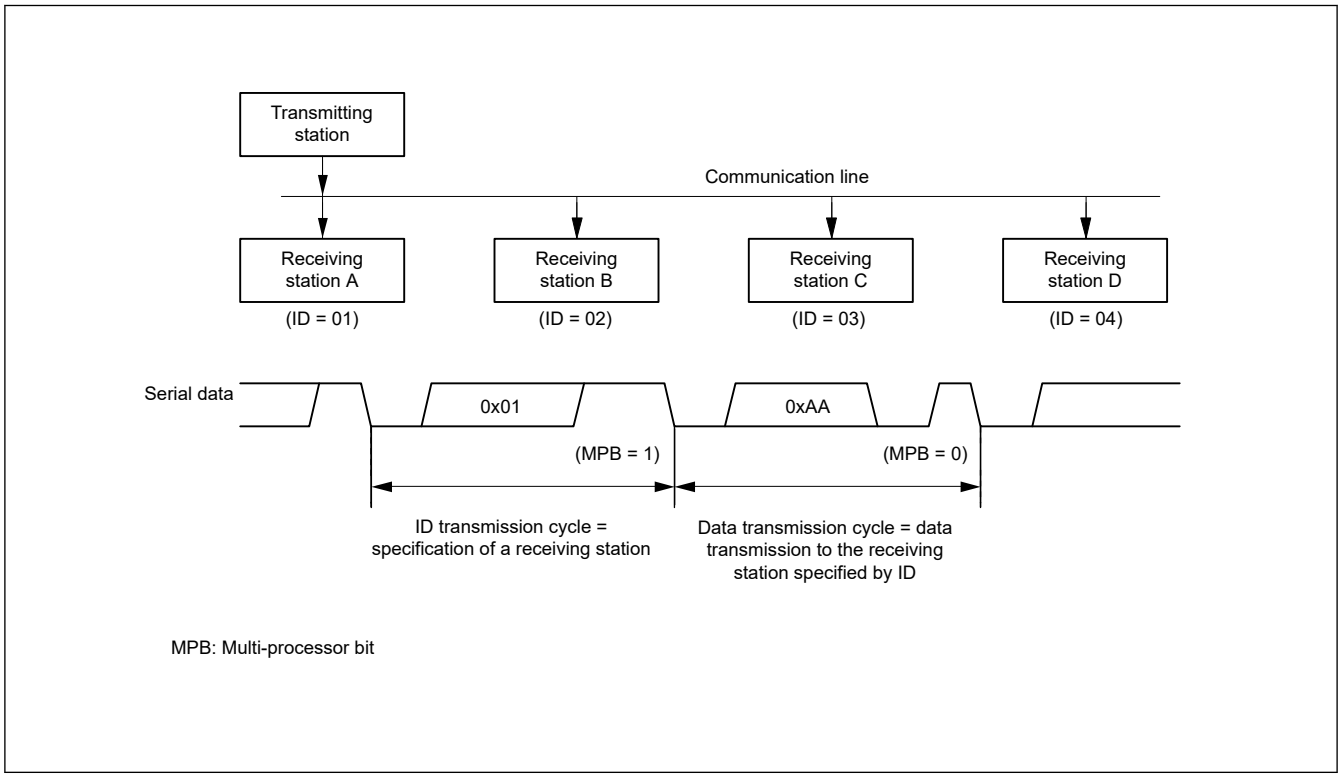


Figure 31.35 Example of communication using multi-processor format with transmission of data 0xAA to receiving station A

(1) Non-FIFO selected

To support this function, the SCI provides the CCR0.MPIE bit. When the MPIE bit is set to 1, the following operations are disabled until the reception of data in which the multi-processor bit is set to 1:

- Transfer of receive data from the RSR register to the RDR register
- Detection of a receive error
- Setting of the respective RDRF, ORER, and FER status flags in the CSR register

When the SCI receives a character in which the multi-processor bit is set to 1, the RDR.MPB bit is set to 1 and the CCR0.MPIE bit is automatically cleared, returning the SCI to non-multi-processor reception operation. If the CCR0.RIE bit is set to 1, an SCIn_RXI interrupt is generated.

When the multi-processor format is specified, the parity bit function is disabled. Apart from this, there is no difference from operation in non-multi-processor Asynchronous mode. The clock used for the multi-processor communication is the same as the clock used in non-multi-processor Asynchronous mode.

(2) FIFO selected

For data transmission, software must write data to TDR.MPBT (Multi-Processor Bit Transfer) that corresponds to transmit data in TDR.TDAT. For data reception, the multi-processor bit that is part of the receive data is written to RDR.MPB and receive data is written to RDR.RDAT.

When the MPIE bit is set to 1, the following operations are disabled until reception of data in which the multi-processor bit is set to 1:

- Transfer of receive data from the RSR register to the RDR.RDAT register
- Detection of a receive error
- Detection of DR
- Setting of the respective RDRF, ORER, and FER status flags in the CSR register

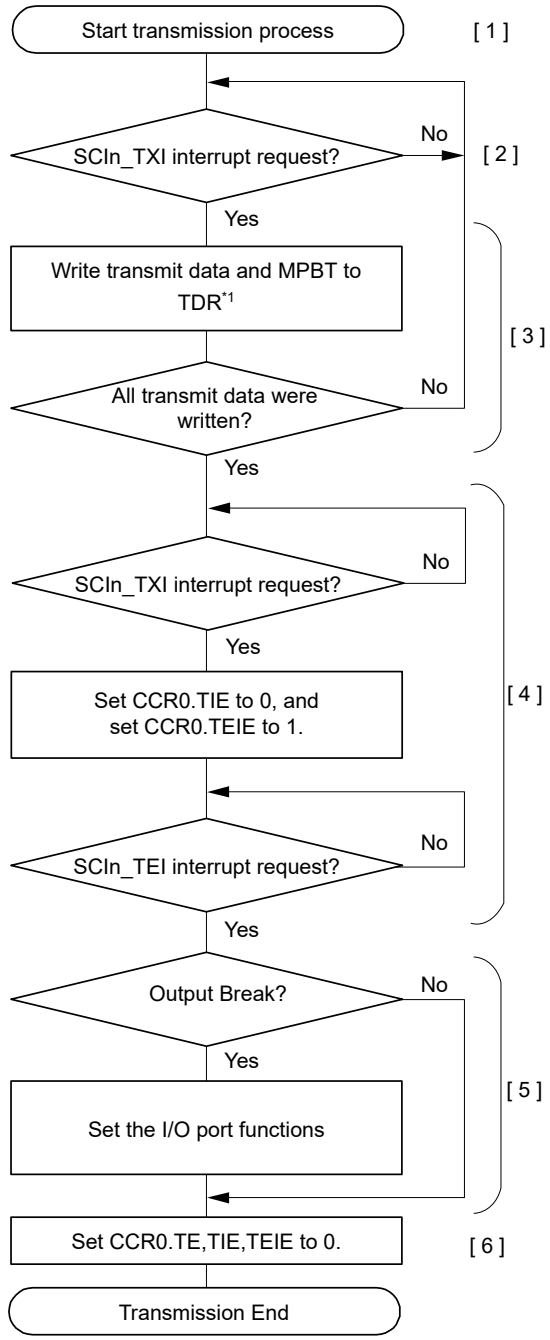
When the SCI receives a character in which the multi-processor bit is set to 1, the RDR.MPB bit is set to 1 and receive data is written to receive-FIFO(RDR.RDAT). The CCR0.MPIE bit is automatically cleared, returning the SCI to normal reception operation. If the CCR0.RIE bit is set to 1, an SCIn_RXI interrupt is generated.

When the multi-processor format is specified, the parity bit function is disabled. Apart from this, there is no difference from operation in normal Asynchronous mode with FIFO selected.

31.4.1 Multi-Processor Serial Data Transmission

(1) Non-FIFO selected

Figure 31.36 shows an example flow of multi-processor data transmission. In the ID transmission cycle, the ID must be transmitted with the TDR.MPBT bit set to 1. In the data transmission cycle, the data must be transmitted with the MPBT bit set to 0. The rest of the operations are the same as operations in Asynchronous mode.



[1] Start transmission process

[2] SCIn_TXI interrupt request:
The transmit-data-empty-interrupt(SCIn_TXI) is generated at following conditions.
When set TE and TIE to 0 at same time.
When transmission-data forwards to TSR from TDR.

[3] Serial transmission continuation procedure:
To continue serial transmission write transmit data and MPBT setting value to TDR*1 once by using a SCIn_TXI. Transmit data and MPBT setting value can also be written by activating the DMAC or DTC.

[4] If you use SCIn_TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR*1 and transmission is started.

[5] Break output at the end of serial transmission:
To output a break, it is necessary to set SPB2IO and SPB2DT, for output status setting when TE is 0.
(If you set these bits at initialization, you can skip this step.)

[6] Set CCR0.TE,TIE,TEIE to 0.

Note 1. In case of byte access with data length 9 bits or to write MPBT bit, write transmit data in order of TDR[15:8] and TDR[7:0].

Figure 31.36 Example flow of multi-processor serial transmission with non-FIFO selected

(2) FIFO selected

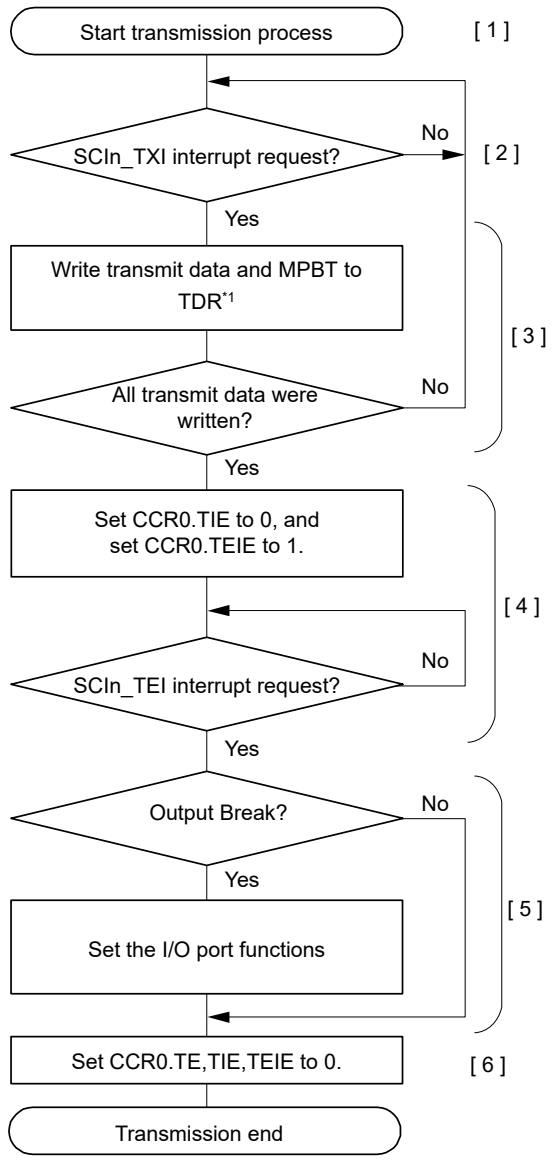
Figure 31.37 shows an example of data format that is written to transmit-FIFO (TDR) in multi-processor mode. The TDR.MPBT bit is set to 1. Data is set to transmit-FIFO (TDR) with the correct data length. Write 0 for unused bits.

Data Length	Register setting		Transmit data in TDR[15:0]															
	CCR3. CHR[1:0]		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bit	1	1	–	–	–	–	–	–	MPBT	–	–	TDAT[6:0]						
8 bit	1	0	–	–	–	–	–	–	MPBT	–	TDAT[7:0]							
9 bit	0	Don't Care	–	–	–	–	–	–	MPBT	TDAT[8:0]								

Note: —: Invalid. The write value should be 0.

Figure 31.37 Data format written to transmit-FIFO (TDR) in multi-processor mode with FIFO selected

Figure 31.38 shows an example flow of multi-processor serial transmission with FIFO selected. In the ID transmission cycle, the ID must be transmitted with the TDR.MPBT bit set to 1. In the data transmission cycle, the data must be transmitted with the MPBT bit set to 0. The rest of the operations are the same as operations in Asynchronous mode with FIFO selected.



[1] Start transmission process
To enable SCIn_TXI interrupt at he start of transmission, set TE and TIE to 1 with one instructions at the same time. And set CCR3.FM,MP to 1.

[2] SCIn_TXI interrupt request:
The transmit-data-empty-interrupt(SCIn_TXI) is generated when transmission-data forwards to TSR from the transmit-FIFO(TDR) and the number of transmit-data written in the transmit-FIFO is equal to or less than the transmit-FIFO threshold.

[3] Serial transmission continuation procedure:
To continue serial transmission, write all transmit data and MPBT setting value to the transmit-FIFO(TDR) by using a SCIn_TXI interrupt. Transmit data and MPBT setting value can also be written to the transmit- FIFO(TDR) by activating the DMAC or DTC.

[4] If you use SCIn_TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR*1.

[5] Break output at the end of serial transmission:
To output a break, it is necessary to set SPB2IO and SPB2DT, for output status setting when TE is 0.
(If you set these bits at initialization, you can skip this step.)

[6] Set CCR0.TE,TIE,TEIE to 0.

Note 1. In case of byte access with data length 9 bits or to write MPBT bit, write transmit data in order of TDR[15:8] and TDR[7:0].

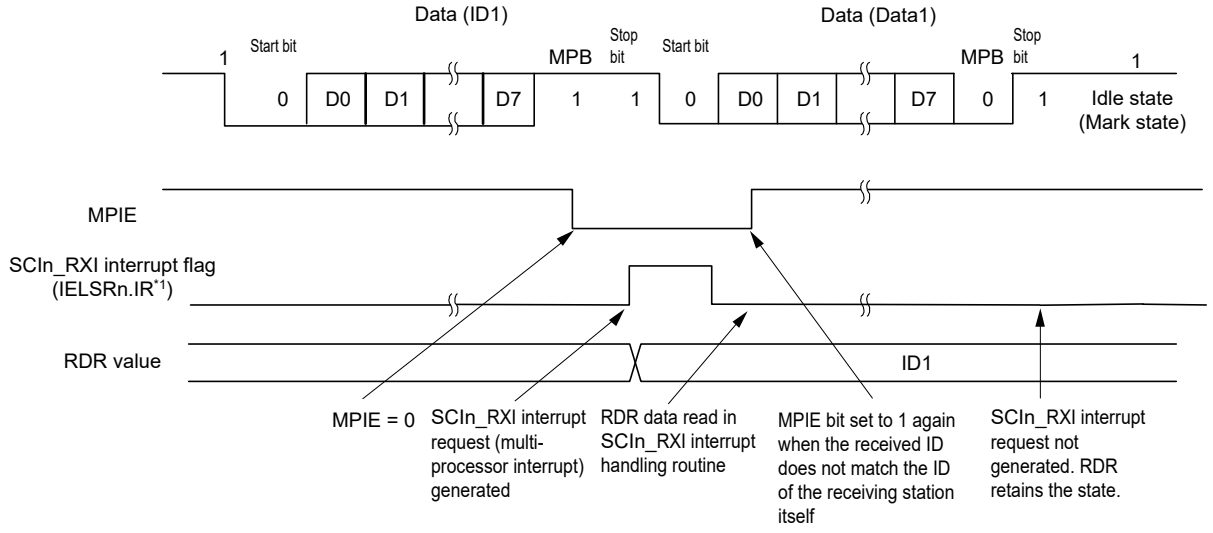
Figure 31.38 Example flow of serial transmission in multi-processor mode with FIFO selected

31.4.2 Multi-Processor Serial Data Reception

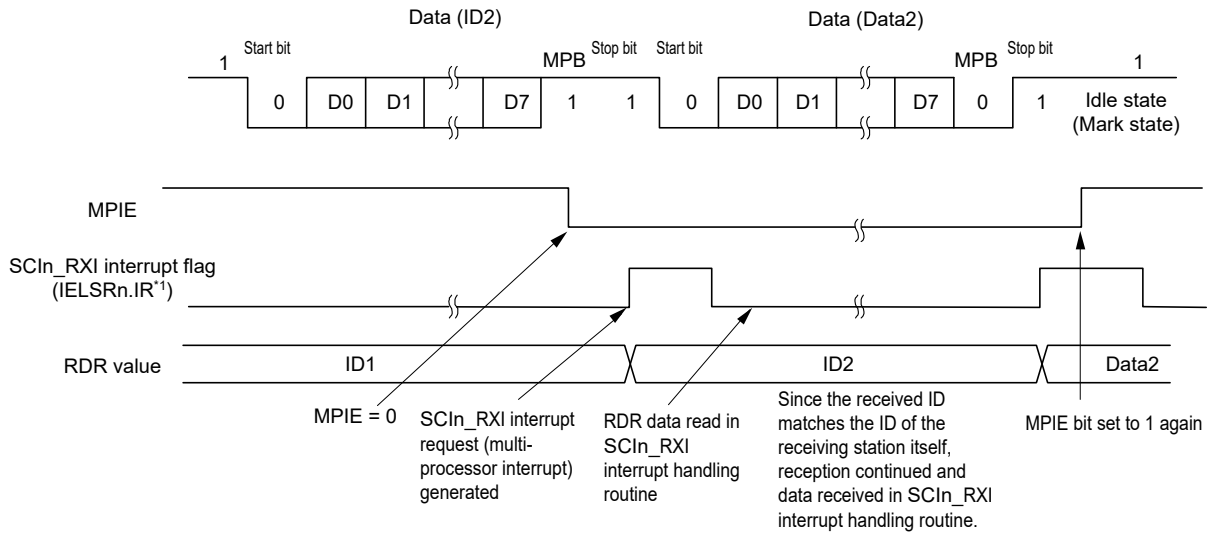
(1) Non-FIFO selected

Figure 31.40 and Figure 31.41 are example flows of multi-processor serial reception. When the CCR0.MPIE bit is set to 1, reading communication data is skipped until reception of communication data in which the multi-processor bit is set to 1. When communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to the RDR register, and the SCIn_RXI interrupt request is generated. The rest of the operations are the same as operations in Asynchronous mode.

Figure 31.39 shows an example operation for data reception.



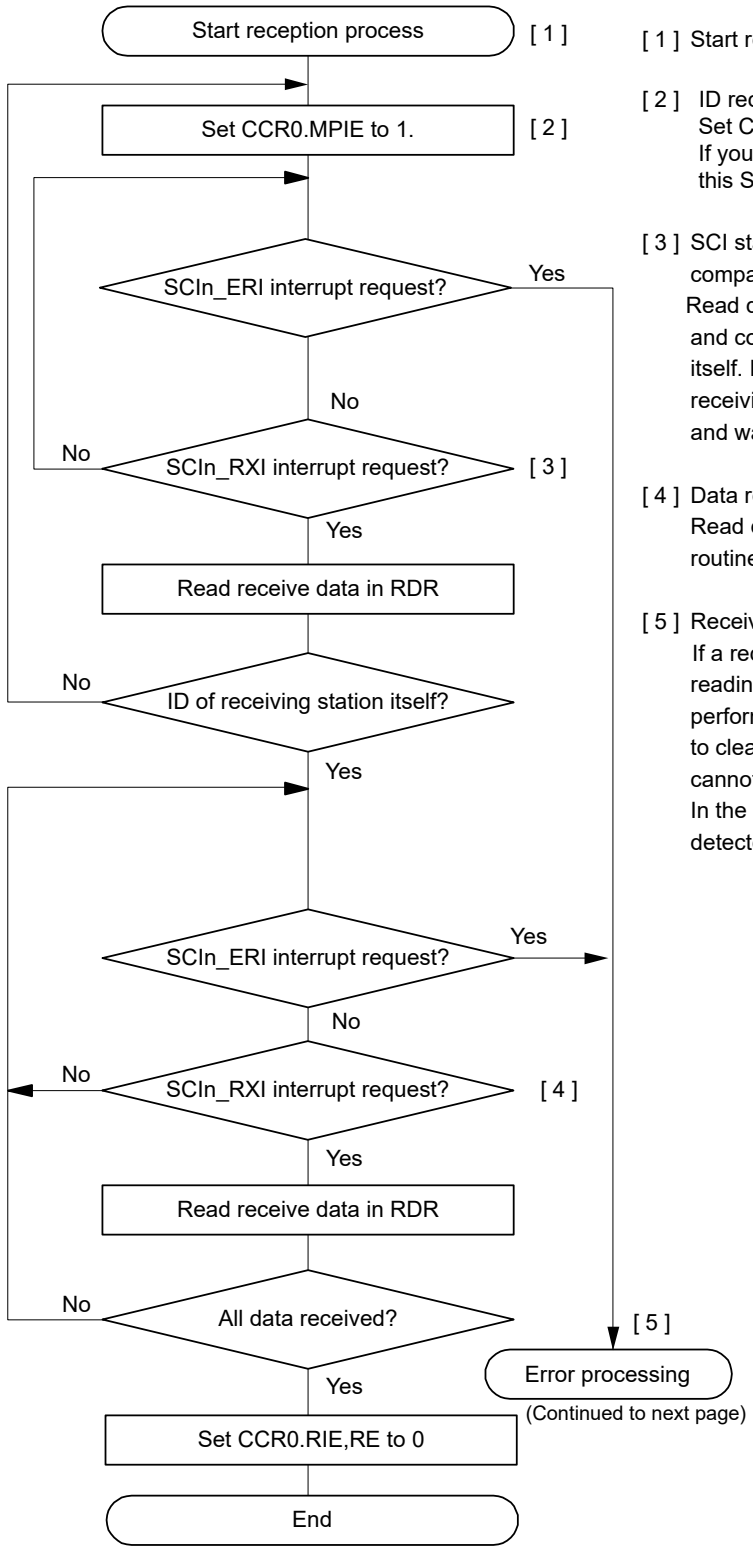
(a) When the received ID does not match the ID of the receiving station itself



(b) When the received ID matches the ID of the receiving station itself

Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.39 Example of SCI reception with 8-bit data, multi-processor bit, and 1 stop bit



[1] Start reception process

[2] ID reception cycle:
Set CCR0.MPIE to 1. and wait for ID reception.
If you set this bits at initialization([1]), you can skip this Step.

[3] SCI status confirmation and reception and comparison of ID:
Read data in RDR at the first SCIn_RXI interrupt and compare it with the ID of the receiving station itself. If the ID does not match the ID of the receiving station itself, set the MPIE bit to 1 again, and wait for another SCIn_RXI interrupt request.

[4] Data reception at an SCIn_RXI interrupt:
Read data in RDR once in the SCIn_RXI interrupt routine.

[5] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading ORER and FER flags in CSR. After performing the appropriate error processing be sure to clear the ORER and FER flags. Reception cannot be resumed if any of these flags is set to 1. In the case of a framing error, a break can be detected by reading CSR.RXDMON bit.

Figure 31.40 Example flow of multi-processor serial reception with non-FIFO selected (1)

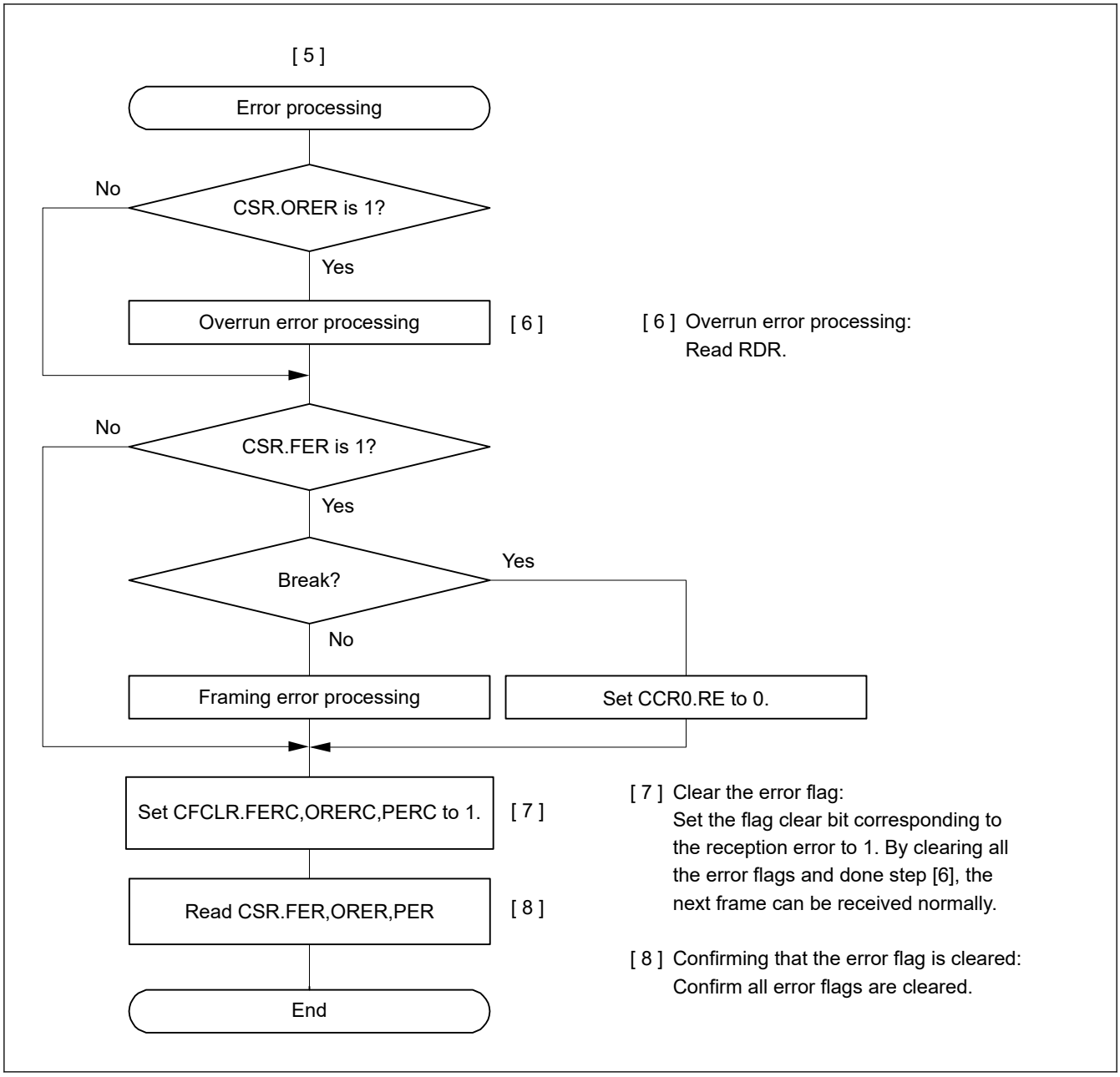


Figure 31.41 Example flow of multi-processor serial reception with non-FIFO selected (2)

(2) FIFO selected

Figure 31.42 shows an example of a data format that is written to receive-FIFO (RDR) in multi-processor mode.

In multi-processor mode, the MPB value that is a part of the receive data is written to the RDR.MPB bit. A value of 0 is written to the RDR.FPER and PER flags. Data is written to receive-FIFO (RDR) with the correct data length. Unused bits are written with 0. When software reads the receive-FIFO (RDR) register, the SCI updates RDR.FFER, FPER, and MPB flags, and receive data (RDAT[8:0]) in receive-FIFO (RDR) with the next data. The FER, PER, and ORER flags in the receive-FIFO (RDR) register always reflect the associated flags in the CSR and FRSR registers.

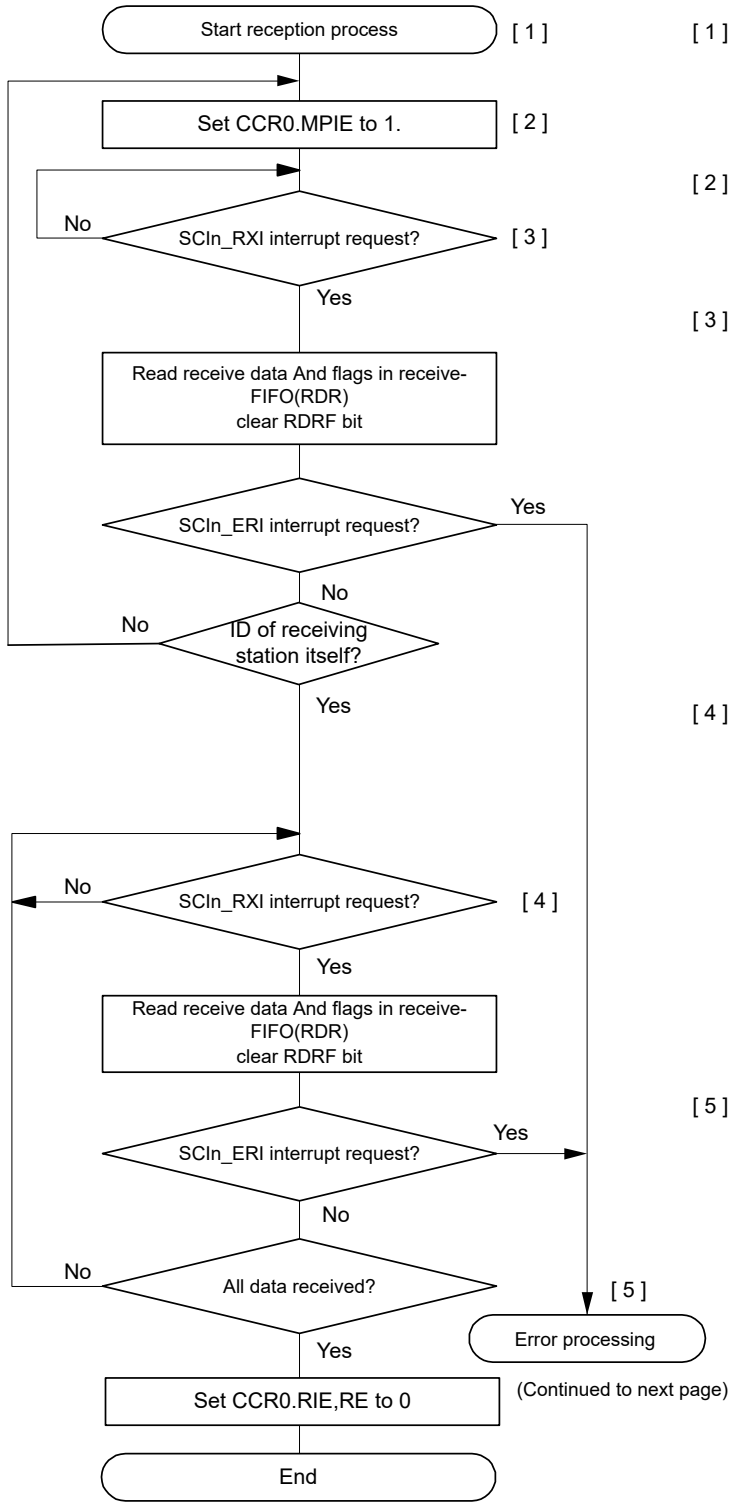
Data Length	Register Setting CCR3.CHR[1:0]		Receive data in RDR[31:0] b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7bit	1	1	0	0	0	FFER	FPER	DR	MPB	0	0	RDAT[6:0]						
8bit	1	0	0	0	0	FFER	FPER	DR	MPB	0	RDAT[7:0]							
9bit	0	Don't Care	0	0	0	FFER	FPER	DR	MPB	RDAT[8:0]								
Data Length	CCR3.CHR[1:0]		b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
7bit	1	1	0	0	0	FER	PER	0	0	ORER	0	0	0	0	0	0	0	0
8bit	1	0	0	0	0	FER	PER	0	0	ORER	0	0	0	0	0	0	0	0
9bit	0	Don't Care	0	0	0	FER	PER	0	0	ORER	0	0	0	0	0	0	0	0

Note: When data length is 7bit, it can read always 0 in RDAT[8:7].
When data length is 8bit, it can read always 0 in RDAT[8].

Figure 31.42 Data format stored in receive-FIFO (RDR) in multi-processor mode with FIFO selected

Figure 31.43 shows an example flow of multi-processor data reception with FIFO selected. When the CCR0.MPIE is set to 1, reading communication data is skipped until reception of communication data in which the multi-processor bit is set to 1. When communication data in which the multi-processor bit is set to 1 is received, the received data, MPB and associated errors are transferred to the receive-FIFO (RDR) register. The CCR0.MPIE bit is automatically cleared and normal reception continues.

If a framing error occurs and the CSR.FER flag is set to 1, the SCI continues data reception. The rest of the operations are the same as operations in Asynchronous mode with FIFO selected.



[1] Start reception process
To generate an SCIn_RXI interrupt request with MP detection, please set FCR.RTRG = 1 and FCR.DRES = 0.

[2] ID reception cycle:
Set CCR0.MPIE to 1. and wait for ID reception. If you set this bits at initialization([1]), you can skip this Step.

[3] SCI status confirmation and reception and comparison of ID:
Skip the communication data until the MPB bit receives data 1. When the MPB bit receives data 1, an SCIn_XRI interrupt request is generated. When an SCIn_RXI interrupt request is received, read the data in the receiving FIFO (RDR register) and compare it to the ID of my station. If it is not my station's ID, set the MPIE bit to 1 again and wait for the SCIn_RXI interrupt request. For my station's ID, set the received FIFO threshold to the necessary value.

[4] Data reception at an SCIn_RXI interrupt:
When the number of received data stored in the received FIFO (RDR register) exceeds the received FIFO threshold, the CSR.RDRF bit becomes 1 and an SCIn_RXI interrupt request is generated. In addition, if the number of received data stored in the received FIFO (RDR register) is 1 or more and less than the received FIFO threshold, the FRSR.DR bit becomes 1 after 15 etus, and an SCIn_RXI interrupt request is generated. The SCIn_RXI interrupt routine reads the data from the received FIFO (RDR register).

[5] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading ORER and FER flags in CSR. After performing the appropriate error processing be sure to clear the ORER and FER flags. In the case of a framing error, a break can be detected by reading the value of the RXDn pin.

Figure 31.43 Example flow of serial reception in multi-processor mode with FIFO selected

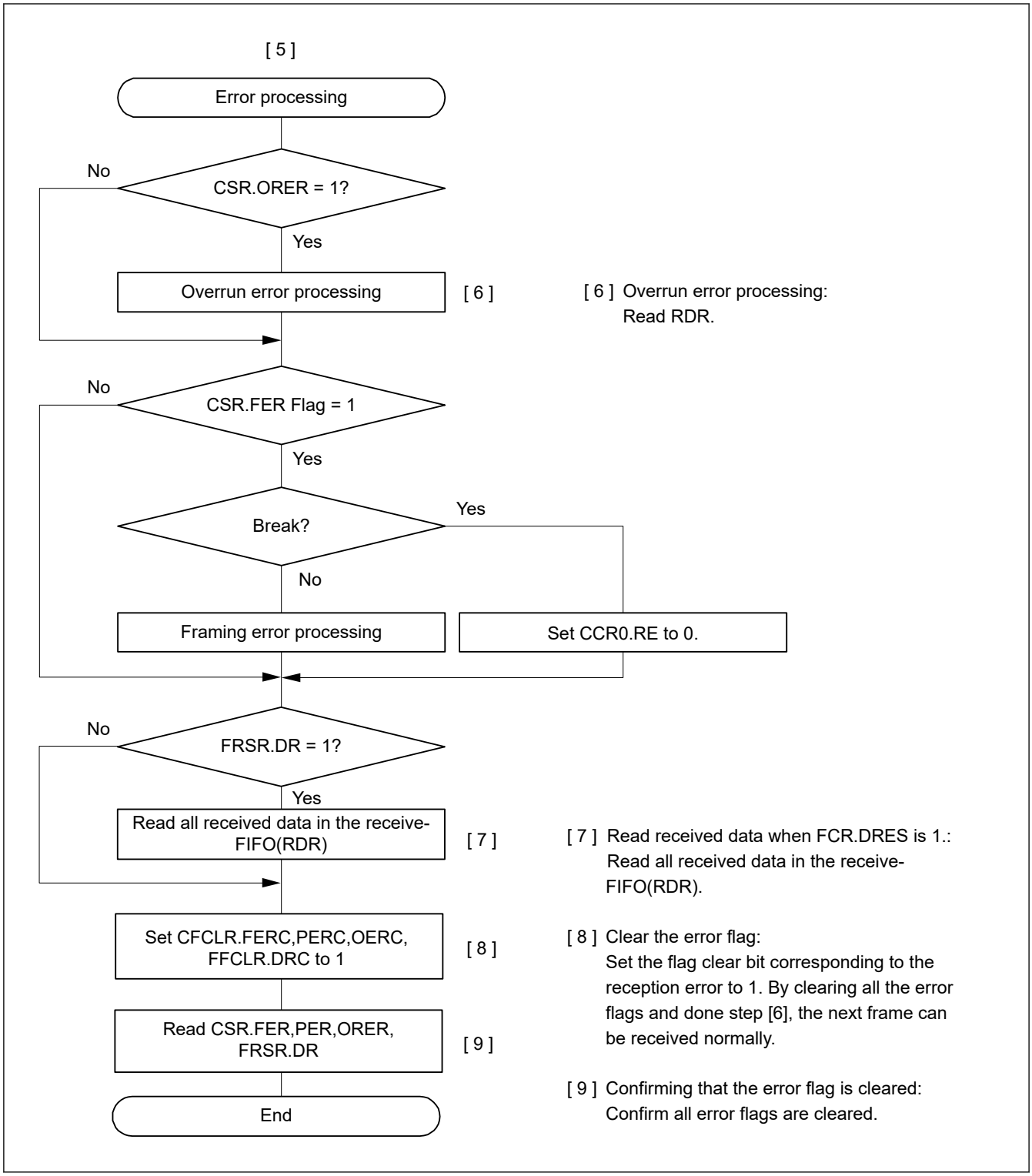


Figure 31.44 Example Flowchart of Serial Reception in Multi-Processor Mode (2) (FIFO selected)

31.5 Operation in Manchester mode

In Manchester mode, the transmit or receive serial data is coded in Manchester encoding.

Figure 31.45 shows the conceptual image of Manchester encoding.

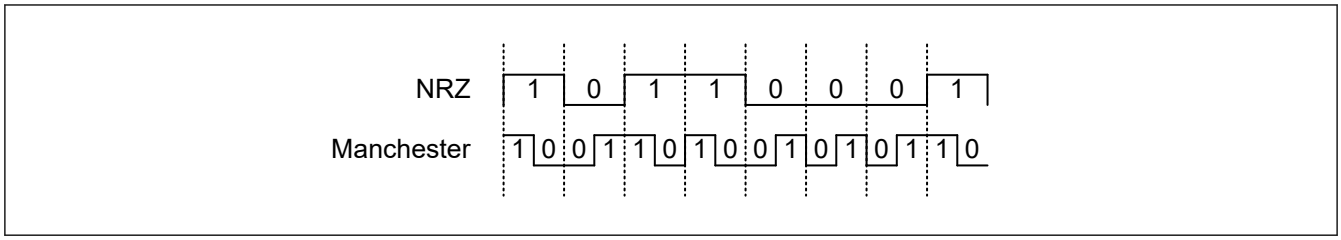


Figure 31.45 Example of Manchester Encoding

In Manchester mode, a preface and a start bit area are added to the transmit data in the register to configure a transmit frame. For transmission, data is encoded in Manchester encoding. When data is received, frames having the same format as transmitted frames are detected and Manchester decoding is performed.

For details on the frame format, see section 31.5.1. Frame Format.

31.5.1 Frame Format

Figure 31.46 shows the frame format in Manchester mode.

In the upper half of the figure, relevant setting registers are shown.

The preface area and the data area are encoded in Manchester encoding.

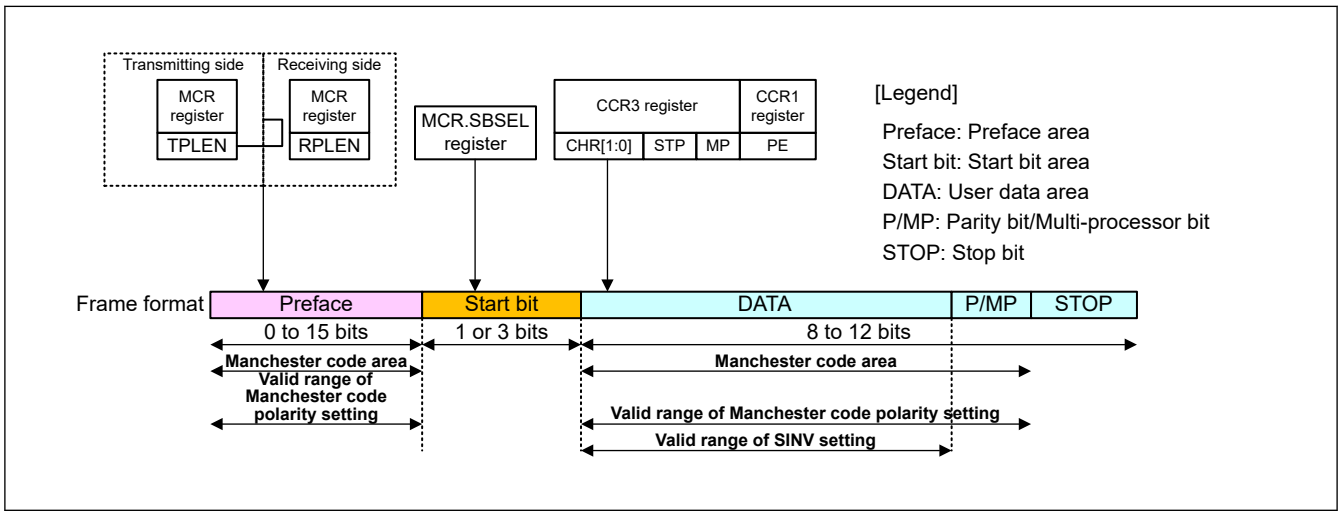


Figure 31.46 Frame Format in Manchester Mode

(1) Preface area

This is a fixed pattern area located at the beginning of each frame.

Different registers are used to set the preface area for transmission and reception. The preface length is determined by setting MCR.TPLEN[3:0] for transmission. It is determined by setting MCR.RPLEN[3:0] for reception.

If it is set to 0, the transmit preface is disabled and is not added.

If it is set to 1d to 15d, a preface whose length is determined by this setting is added.

(For example, if it is set to 1d, a 1-bit preface is added. If it is set to 15d, a 15-bit preface is added.)

The preface pattern is set with MCR.TPPAT[1:0] for transmission and MCR.RPPAT[1:0] for reception, and is selected from four types of patterns.

Figure 31.47 shows how the preface pattern is set. The preface area and the start bit area are added for each communication frame.

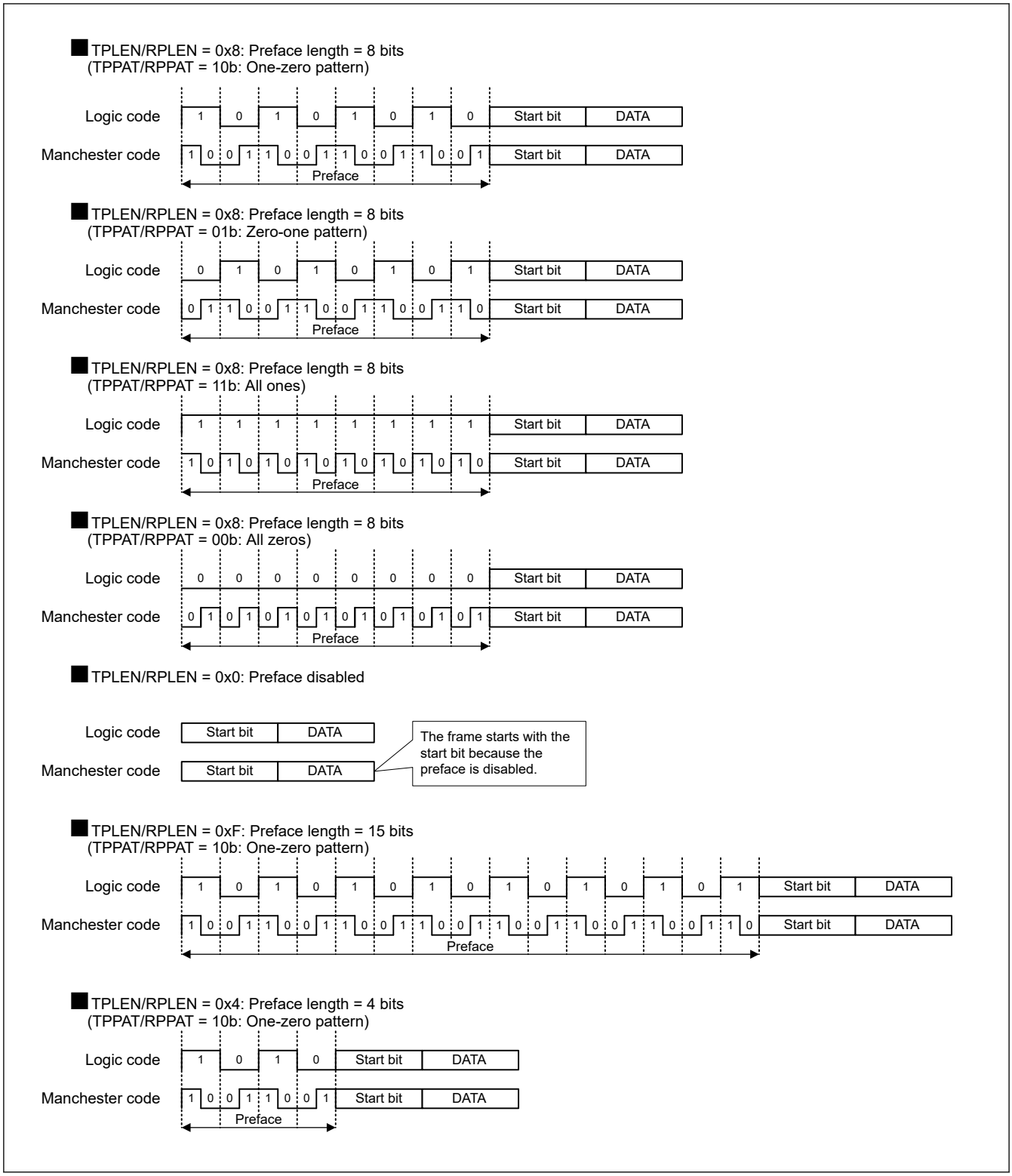


Figure 31.47 Preface Pattern Setting Example

(2) Start bit area

This is an area indicating the start of valid data in a frame. It is added after the preface area.

The start bit length is determined by MCR.SBSEL setting. When MCR.SBSEL = “0”, the start bit length is 1 bit.

When MCR.SBSEL = “1”, the start bit length is 3 bits.

When MCR.SBSEL = “1”, the SYNC type can be selected from command SYNC and data SYNC.

Command SYNC means the three start bits are added as a one-to-zero transition.

Data SYNC means the three start bits are added as a zero-to-one transition.

The SYNC type is determined by the MCR.SYNSEL, MCR.SYNVAL and TDR.TSYNC settings.

(When receiving, the received result is applied to MSR.RSYNC.)

When MCR.SBSEL = “0”, the start bit is added as a zero-to-one or one-to-zero transition.

The selection is determined by the MCR.SYNVAL setting.

The MCR.SYNSEL bit specifies the destination to be referred to when setting for transmission.

When the MCR.SYNSEL bit is set to 1, the MCR.SYNVAL setting is referred to. When the MCR.SYNSEL bit is set to 0, the TDR.TSYNC setting is referred to.

Figure 31.48 shows the state of the start bit area according to the settings in the MCR.SYNSEL, MCR.SYNVAL and TDR.TSYNC registers in the case of transmission. Figure 31.49 shows that in the case of reception.

The start bit(s) is not affected by the MCR.TMPOL or MCR.RMPOL setting.

*d.c. = don’t care, Preface = preface area, START = start bit area, Data = user data area,
P = parity bit, MP = multi-processor bit, STOP = stop bit

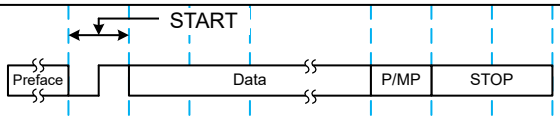
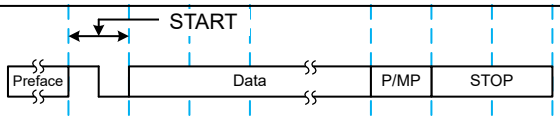
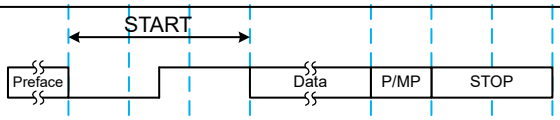
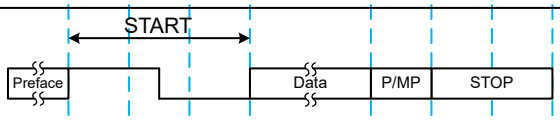
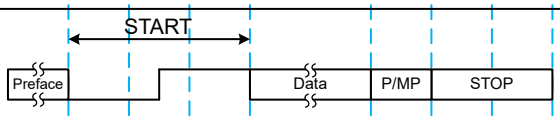
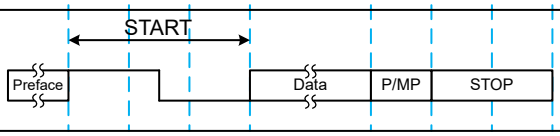
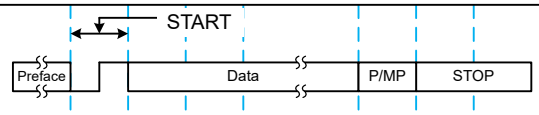
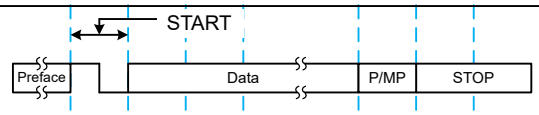
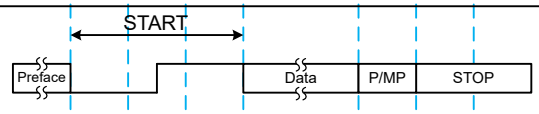
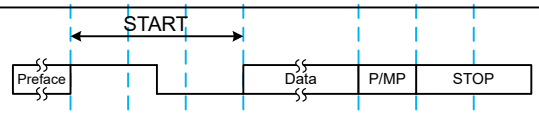
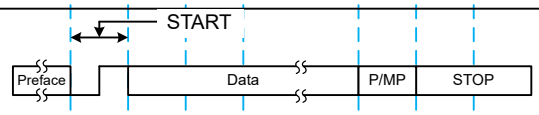
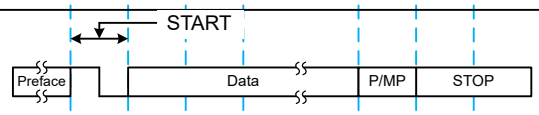
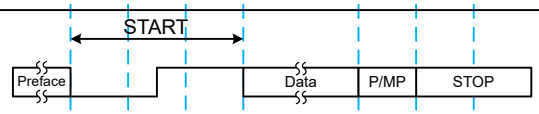
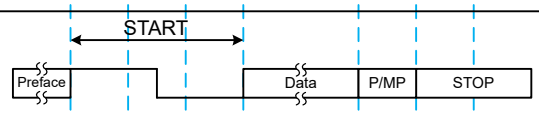
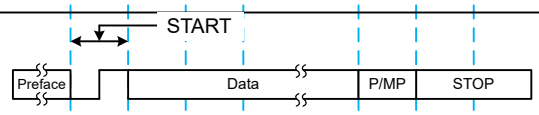
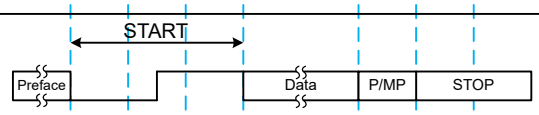
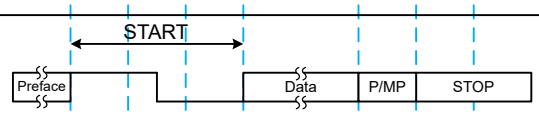
Register setting				Output signal	Output Start bit
MCR			TDR	Output TXDn waveform	
SBSEL	SYNSEL	SYNVAL	TSYNC		
0	d.c.	0	d.c.		1bit 0 to 1 transition
0	d.c.	1	d.c.		1bit 1 to 0 transition
1	0	0	d.c.		3bit 0 to 1 transition (DATA SYNC)
1	0	1	d.c.		3bit 1 to 0 transition (COMMAND SYNC)
1	1	d.c.	0		3bit 0 to 1 transition (DATA SYNC)
1	1	d.c.	1		3bit 1 to 0 transition (COMMAND SYNC)

Figure 31.48 Settings Related to and Format of the Start Bit Area at Transmission

d.c. = don't care, Preface = Preface area, START = Start bit area, Data = Data area
P = Parity bit, MP = Multi-processor bit, STOP = Stop bit

Register setting				Input signal	Start bit detection result*1	Register indication
MCR			TDR	RXDn input waveform		MSR.RSYNC
SBSEL	SYNSEL	SYNVAL	TSYNC			
0	d.c.	0	d.c.		Normal start bit (1 bit: 0-to-1 transition)	0
					Start bit error	0
					Start bit error	0
					Start bit error	0
0	d.c.	1	d.c.		Start bit error	0
					Normal start bit (1 bit: 1-to-0 transition)	0
					Start bit error	0
					Start bit error	0
1	d.c.	d.c.	d.c.		Start bit error	0
					Start bit error	0
					Data SYNC	0
					Command SYNC	1

Note 1. Data other than the start bit is assumed to be normal.

Figure 31.49 Settings Related to and Judgment of the Start Bit Area at Reception

(3) DATA

Since the format of the data area is the same as that of the Asynchronous mode, see section 31.3.1. Serial Data Transfer Format.

As shown in Figure 31.45, Frame Format in Manchester Mode, the stop bit is not included in the Manchester encoding range.

31.5.2 Clock

As the transfer clock in Manchester mode, the clock generated by the on-chip baud rate generator is used by setting the CCR2.CKS[1:0] bit.

Also it is possible to set the oversampling (transfer rate of one-bit period) by CCR2.ABCS bit.

When the CCR2.ABCS bit is set to 0, oversampling x16 is selected with the one-bit period being 16 cycles of the base clock. When the CCR2.ABCS bit is set to 1, oversampling x8 is selected with the one-bit period being 8 cycles of the base clock.

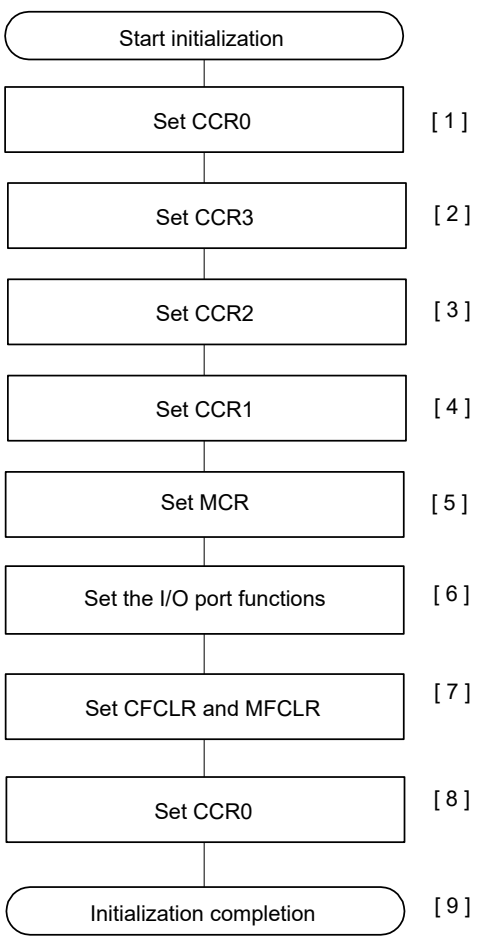
31.5.3 Initialization of the SCI in Manchester Mode

Before transferring data, write 0 to CCR0.TE and CCR0.RE (or write the initial value to CCR0 register) and initialize the SCI following the example of flowchart shown in Figure 31.50.

Whenever the operating mode or transfer format is changed, the CCR0 register must be initialized before the change is made.

Note that setting the CCR0.RE bit to 0 initializes none of the ORER, FER, PER, MER, RDRF, SYER, PFER and SBER flags, and the RDR registers.

Note also that switching the value of CCR0.TE from 0 to 1 when CCR0.TIE is 1 generates a SCIn_TXI interrupt request.



[1] Set the following bits of CCR0 to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set the driver control, Multi-processor communication, the communication mode, transmission/reception format, and Clock enable.
Do not use function bits such as CPOL and CPHA at the initial value.

[3] Set the clock selection, Bit rate setting.
Leave unused function bits at their initial values.(ex. BRME)

[4] Set the noise filter, loopback control, communication pin status, parity check, and CTS / RTS function.

[5] Set Manchester transmission / reception setting.
Start bit setting, retiming setting, Manchester polarity setting, transmission preface setting, reception preface setting.

[6] Set the I / O port so that the required pin function is valid among the TXDn, RXDn, and SCKn pins.

[7] Set 1 to the following clear bits in the CFCLR and MFCLR registers to clear the target flag.
CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
MFCLR.MERC, SBERC, SYERC, PFERC
This procedure can be omitted for the initial setting after reset.

[9] Set the TE bit or RE bit to 1.
To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time, and multiprocessor function. The TXDn and RXDn pins can be used by setting the TE and RE bits.

Figure 31.50 SCI Initialization Flow in Manchester Mode

31.5.4 Double-speed operation

When the ABCS bit in CCR2 is set to 1 and eight pulses of the base clock for a 1-bit period is selected, the SCI operates on the bit rate twice that of when ABCS is set to 0.

When the BGDM bit in CCR2 is set to 1, the cycle of the base clock is reduced to half and the SCI operates on the bit rate twice that of when ABCS is set to 0.

When the ABCS and the BGDM bits in CCR2 are set to 1, the SCI operates on the bit rate four times that of when the ABCS and the BGDM bits in CCR2 are set to 0.

31.5.5 CTS and RTS functions

The CTS function uses input on the CTSn pin in transmission control. Setting the CTSE bit in CCR1 to 1 enables the CTS function. The CTSn_RTSn pin can be set as a multiplexed pin which allows one pin to be used for either CTS or RTS function, or as dedicated pins with each pin at CTSn pin for CTS function and CTSn_RTSn pin for RTS function. Use the CTSPEN bit in CCR1 for this setting.

When the CTS function is enabled, reception starts only when the CTSn pin is at the low level.

Applying a high level to the CTSn pin after transmission starts does not affect transmission of the current frame, which continues.

The RTS function uses output on the CTSn_RTSn pin to request transmission. When the SCI is ready to receive, it outputs a low level to the CTSn_RTSn pin, Conditions for output of the low level and high level are as follows:

[Conditions for low-level output]

When all conditions listed below are satisfied:

- The value of the RE bit in CCR0 is 1.
- When The SCI is ready to receive next.
  - When there is no received data yet to be read and not receiving
  - All of the following flags are set to 0: CSR.ORER, FER, and PER, MSR.MER, SYER (when SYEREN = 1), PFER (when PFEREN = 1) and SBER flags (when SBEREN = 1).

[Conditions for high-level output]

- When the conditions for low output are not satisfied

31.5.6 Serial data transmission in Manchester mode

The SCI encodes data in Manchester encoding and sends the resultant data in Manchester mode.

When the polarity setting (MCR.TMPOL) set to 0, logic 0 is coded as a zero-to-one transition in Manchester code and logic 1 is coded as a one-to-zero transition in Manchester code.

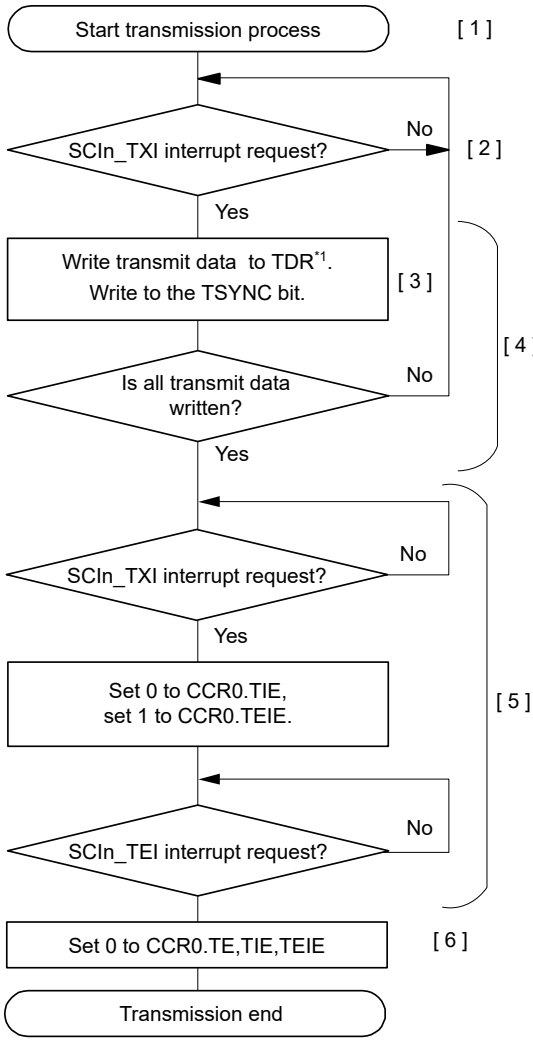
When the polarity setting (MCR.TMPOL) set to 1, logic 0 is coded as a one-to-zero transition in Manchester code and logic 1 is coded as a zero-to-one transition in Manchester code.

For this reason, a level transition occurs with the Manchester encoded data in the middle of individual logic data. (See Figure 31.45).

The transmitter constructs transmit frames in a specific format by adding a preface area to data and setting the start bit(s) according to the polarity setting and sends resultant serial data.

For details on the frame format, see section 31.5.1. Frame Format.

Figure 31.51 shows the flowchart in transmission. At transmission starts, set the CCR0.TIE and CCR0.TE bits to 1 simultaneously with one instruction. Then, a SCIn_TXI interrupt request is generated. Figure 31.52, Figure 31.53, and Figure 31.54 show examples of the operation for serial transmission in Manchester mode.



[1] Start transmission process

[2] SCIn_TXI interrupt:
A transmit data empty interrupt (SCIn_TXI) request is generated when CCR0.TIE = 1 and TE = 0->1, and when transmit data is transferred from TDR to TSR.

[3] TDR setting:
Write the transmit data to TDR.
When TSYNC is used, also set TDR.TSYNC.

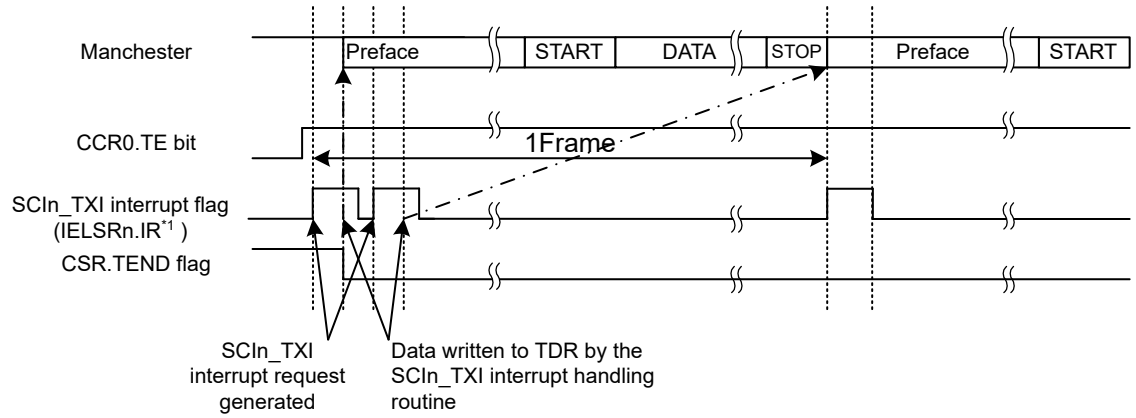
[4] Serial transmission continuation procedure:
To continue serial transmission, write the transmit data to TDR*1 once by using an SCIn_TXI interrupt.
The transmit data can also be written to TDR*1 by activating the DMAC or DTC in the SCIn_TXI interrupt request.

[5] When SCIn_TEI interrupt requests are used, set the CCR0.TIE bit to 0 and the CCR0.TEIE bit to 1 after the last of the data to be transmitted is written to TDR and it's transmission is started.

[6] Set 0 to CCR0.TE, TIE and TEIE.

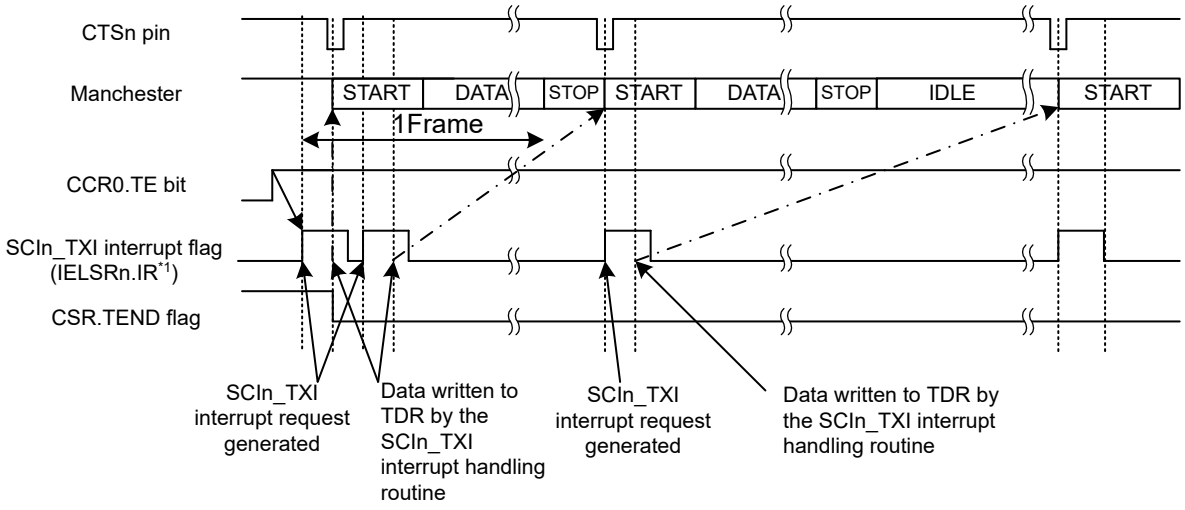
Note 1. If TDR[15:8] requires data write and byte access is required, write the transmit data in the order of TDR [15:8] and TDR [7:0].

Figure 31.51 Example of Serial Transmission Flowchart in Manchester Mode



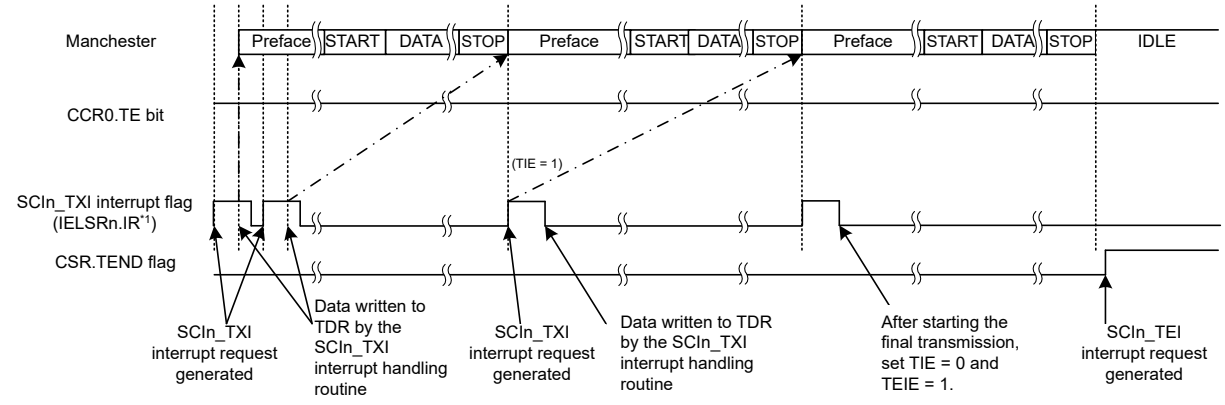
Note 1. For corresponding interrupt numbers, see section 13, Interrupt Controller Unit (ICU)

Figure 31.52 Example of Start-of-Transmission Operation for Serial Transmission in Manchester mode (with Preface but Without the CTS Function)



Note 1. For corresponding interrupt numbers, see section 13, Interrupt Controller Unit (ICU)

Figure 31.53 Example of Start-of-Transmission Operation for Serial Transmission in Manchester Mode (Without Preface but with the CTS Function)



Note 1. For corresponding interrupt numbers, see section 13, Interrupt Controller Unit (ICU)

Figure 31.54 Example of End-of-Transmission Operation for Serial Transmission in Manchester Mode (with Preface but Without the CTS Function)

31.5.7 Serial Data Reception in Manchester Mode

In Manchester mode, the SCI operates on a base clock with a frequency of 16 times*1 the bit rate. Reception starts by sampling the falling edges of received data at the base clock. As shown in Figure 31.55, reception starts at a falling edge of the received data and it continues if the received data keeps low for the duration of 1/4 bit. If the received data goes high within the duration of 1/4 bit, the SCI judges it as an error and waits for a falling edge again.

If a high level is expected in the first half of a bit in the received data, the SCI judges a low level that continues for one base clock cycle as an error and ignores the change to the low level.

Note 1. This is the case when CCR2.ABCS = 0. When CCR2.ABCS = 1, the SCI operates on a base clock with a frequency of 8 times the bit rate.

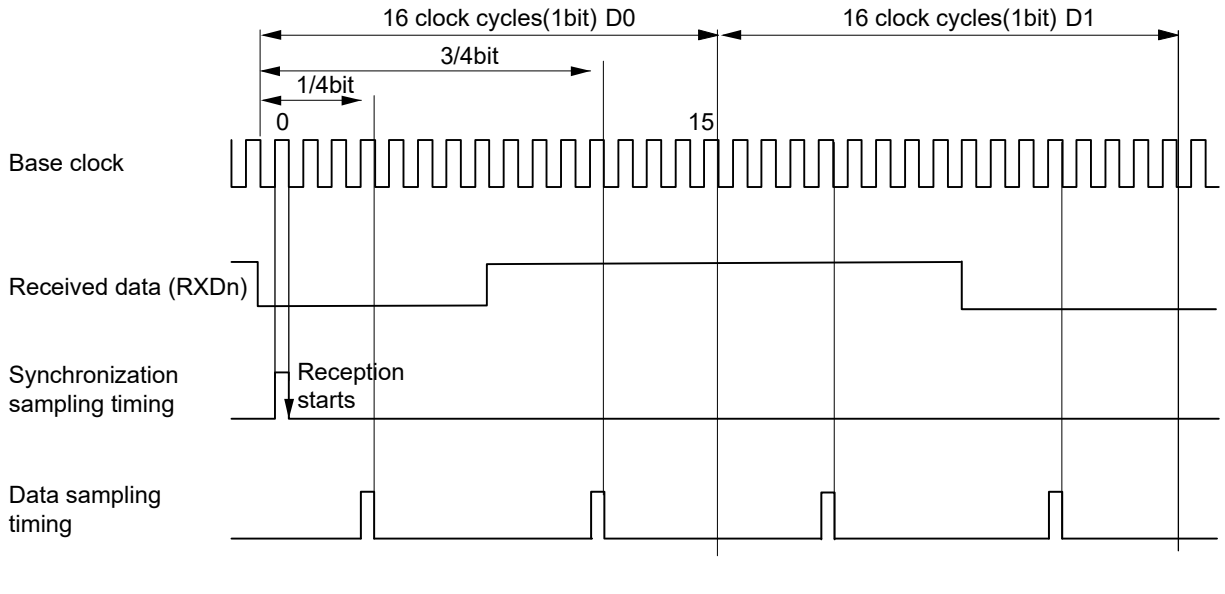


Figure 31.55 Data Reception Sampling Timing in Manchester Mode

In Manchester mode, data reception starts with detection of a preface and start bit area.

The SCI checks the input from the RXDn pin to see whether a preface is added based on the value of MCR.RPLEN.

If the preface is disabled (MCR.RPLEN = 0), it moves on to the detection of a start bit area without detecting a preface.

When a preface is enabled, it identifies a preface pattern setting according to the set value in MCR.RPPAT, and compares it with the RXDn input for a pattern match to detect a preface pattern.

Upon detection of a preface pattern match, it judges it as a normal preface and moves on to the detection of a start bit area.

If detecting a preface pattern mismatch or a Manchester code error in the preface area, it judges it as a preface error and asserts a preface error (PFER).

For start bit detection, the SCI selects an expected value based on the register settings (MCR.SBSEL and SYNVAL), compares it with the RXDn input for a pattern match to detect a start bit area. Upon detection of a start bit pattern match, it judges it as a normal start bit area and moves on to the data processing.

Only when a preface and a start bit area are detected normally, it moves on to the next phase of data reception.

Upon detection of a start bit pattern mismatch, it asserts a start bit error flag (SBER).

In data processing, the SCI shifts the data by the expected received data length based on the register settings (CCR3.CHR[1:0]) through the RSR register. If two sampling points in a bit of the received data are identical, the SCI judges this as a Manchester code error.

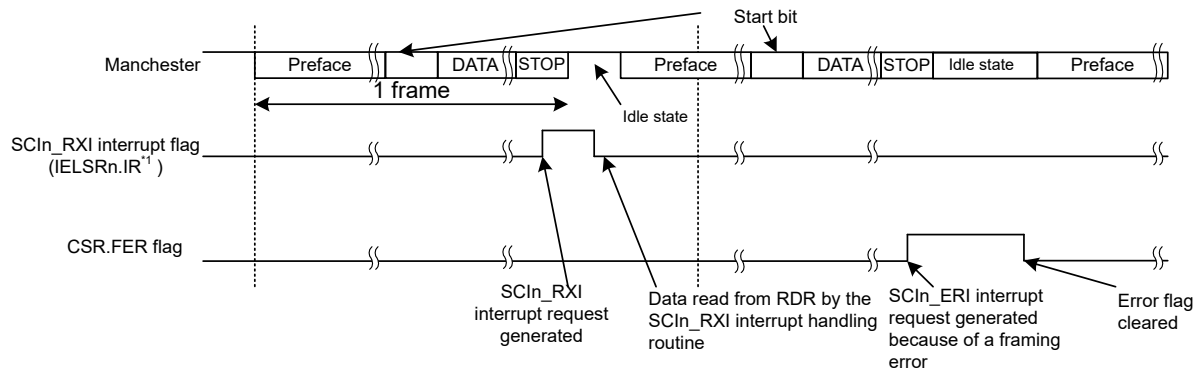
For details, see section 31.5.11. Errors in Manchester Mode (4).

When the parity function is disabled (CCR1.PE = 0), the SCI moves on to the next phase of stop bit detection. When the parity function is enabled (CCR1.PE = 1), the SCI performs parity checking. If detecting a parity error, it asserts a parity error flag (PER), and then moves on to stop bit detection.

In stop bit detection, the SCI checks the following in the stop bit area of the received frame:

It has two sampling points in a bit. If both points are at the high level, the bit is recognized as a normal stop bit and the data is stored in the RDR register. At least one low-level point is judged as an abnormal stop bit, causing a framing error flag (FER) to be set. Even when an error is detected, the received data is stored in the RDR register as abnormal data.

Figure 31.56 shows an example of the operation for serial data reception in Manchester mode.



Note 1. For corresponding interrupt numbers, see section 13, Interrupt Controller Unit (ICU).

Figure 31.56 Example of Operation for Serial Data Reception in Manchester mode (with a Preface)

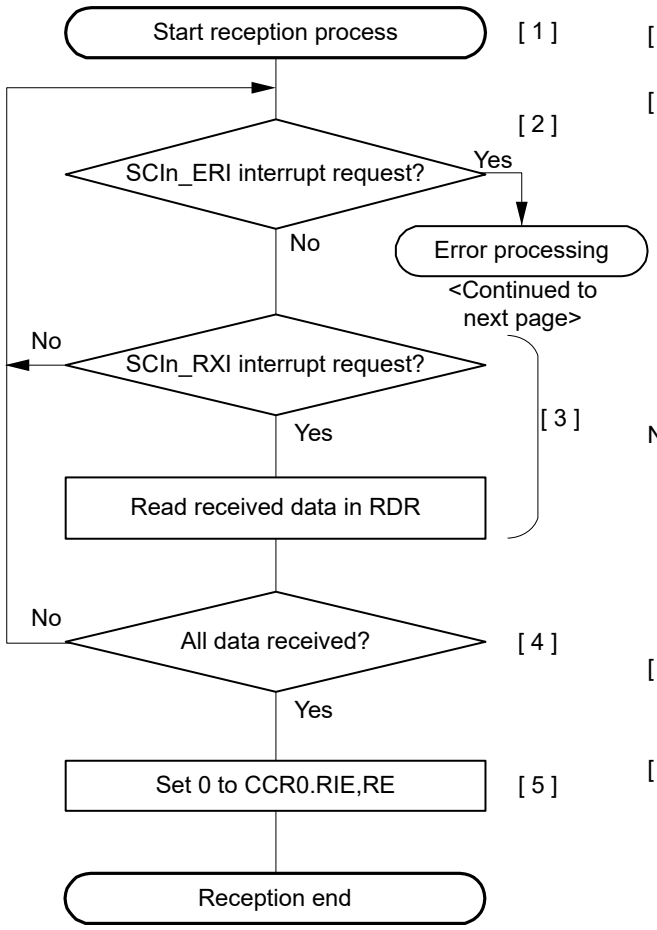
For the state of each status flag in the CCR0 register and RXDn input processing when a receive error is detected, see section 31.5.11. Errors in Manchester Mode.

If a receive error is detected, an SCIn_ERI interrupt request is generated but an SCIn_RXI interrupt request is not generated.

Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, PER, MER, SYER*1, PFER*1, and SBER*1 flags to 0 before resuming reception. Also, be sure to read the RDR register during overrun error processing. When a reception is forcibly terminated by setting the CCR0.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in the RDR register.

Figure 31.57 and Figure 31.58 show examples of serial data reception flowchart in Manchester mode.

Note 1. Effective when the corresponding bit is enabled.



[1] Start reception process

[2] Receive error processing:
When a receive error occurs, SCI generates an SCIn_ERI interrupt . An error is identified by reading the ORER, PER, FER in CSR and MER, SBER, SYER, PFER in MSR. After performing the appropriate error processing, always set the ORER, PER, FER, MER, SBER, SYER, PFER flags to 0. Reception cannot be resumed if any of these flags is set to 1.

Note: Each of the PFER, SYER, SBER bits in MSR has an enable bit (PFEREN, SYEREN, and SBEREN in MECR). If the corresponding enable bit is set to 0 (disable), the flag value does not affect the communication. Perform error processing as required.

[3] Read the received data in RDR once in the SCIn_RXI interrupt handling routine.

[4] Serial reception continuation procedure:
To continue serial reception, before the stop bit of a frame is received, read data from RDR in the SCIn_RXI interrupt handling routine. The RDR data can also be read by activating the DMAC or DTC.

[5] Write 0 to CCR0.RIE and RE.

Figure 31.57 Example of Serial Data Reception Flowchart in Manchester Mode (Normal Reception)

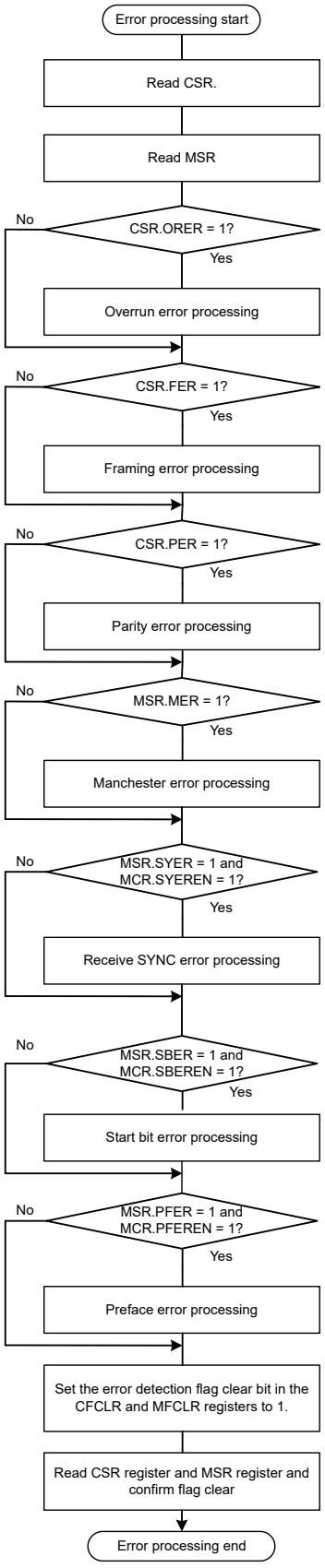


Figure 31.58 Example of Serial Reception Flowchart in Manchester Mode (Error Processing)

31.5.8 Operation When Multi-Processor Bit Is Used

See section 31.4. Multi-Processor Communication Function (1) for the operation in Manchester mode when using multi-processor mode because the operation is the same.

A preface and a start bit area are added to the frame format in Manchester mode. See Figure 31.58 for error processing in Manchester mode for the reception flowchart (Figure 31.41). See Table 31.37 for the operation status when detecting various errors.

31.5.9 Receive Retiming

This function corrects the timing for each central edge of the bit, taking advantage of the fact that each bit has an edge in the center in Manchester code.

The receive retiming function can be turned on or off by setting the ERTEN bit in the MCR register.

When the receive retiming function is turned off (MCR.ERTEN = 0), retiming is not performed, causing misalignment between the internal clock and the RXDn input to be accumulated and the receive margin to be reduced.

When the receive retiming function is turned on (MCR.ERTEN = 1), retiming is performed for the preface area, the start bit area*1, and the data area (excluding the stop bit).

Note 1. Retiming is not performed for the start bit area if the preface length is 0 and the start bit length is 3.

As an example, the receive retiming when oversampling x16 is selected is shown below.

When detecting an RXDn input edge two to four cycles before the expected receive cycle, the receive processing is shortened by one sampling CLK cycle.

When detecting a RXDn input edge two to three cycles after the expected receive cycle, the receive processing is extended by one sampling CLK cycle.

(Even if the clock is misaligned with the data by more than two cycles, one cycle is corrected for each bit.)

Figure 31.59 shows the conceptual image of receive retiming range.

When detecting an edge in the tolerance area in the figure, data is received as is without making correction.

When detecting an edge in the SyncJump area in the figure, data is corrected for reception.

When detecting an edge in the SyncError area in the figure, data is received as abnormal data with no correction made.

For a Manchester code error (data matches at the 1/4-phase and 3/4-phase sampling points), the SCI reports a code error.

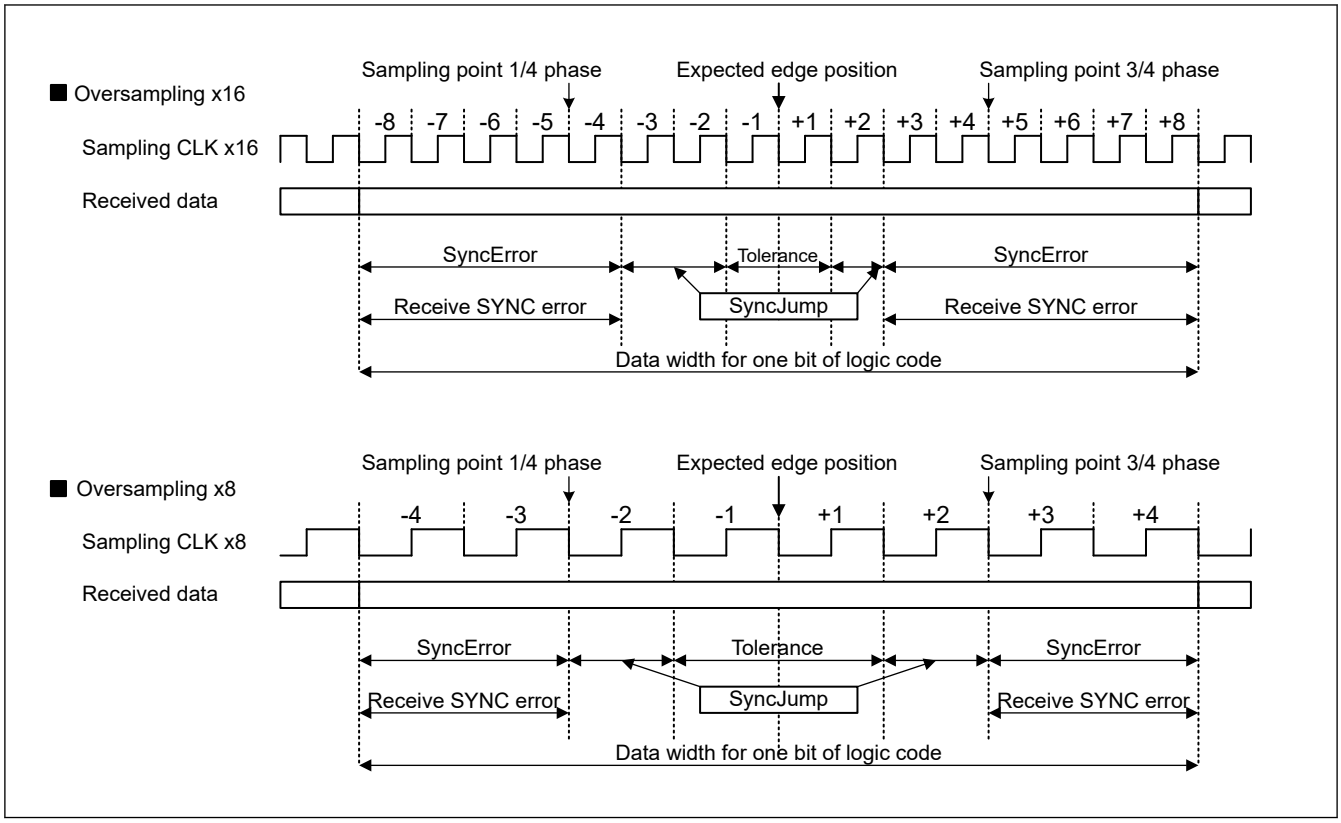


Figure 31.59 Conceptual Image of Reception Retiming Range

31.5.10 Polarity Setting for Manchester Code

The polarity of the Manchester code can be set with the Manchester Control Register (MCR).

It can be set separately for transmission and reception. Use the MCR.TMPOL bit to set the polarity for transmission and the MCR.RMPOL bit to set the polarity for reception.

The Manchester code polarity setting is valid for the preface area, the data area, and the parity or multi-processor area.

When the initial settings (TMPOL/RMPOL = 0) are used for the polarity of Manchester code, logic 0 is encoded as a zero-to-one transition in Manchester code and logic 1 is encoded as a one-to-zero transition in Manchester code. If the settings are changed to TMPOL/RMPOL = 1, logic 0 is encoded as a one-to-zero transition in Manchester code and logic 1 is encoded as a zero-to-one transition in Manchester code. Figure 31.60 shows the conceptual image of the settings and operation.

Separately from the function above, the transmitted and received data in the data area can be inverted by the transmitted/received data inversion function (CCR3.SINV). Since the polarity of Manchester code (MCR.TMPOL/RMPOL) can be set separately from the transmitted/received data invert function (CCR3.SINV), if both are set to inversion (MCR.TMPOL/RMPOL = 1 and CCR3.SINV = 1), the transmitted and received data are set to initial state (inversion + inversion = normal).

The polarity of the start bit area can be set by a register different from the ones mentioned above.

Since a different register is used, the polarity of the start bit area is not affected by the polarity setting for Manchester code mentioned above.

For details on the setting for the start bit area, see section 31.5.1. Frame Format (2).

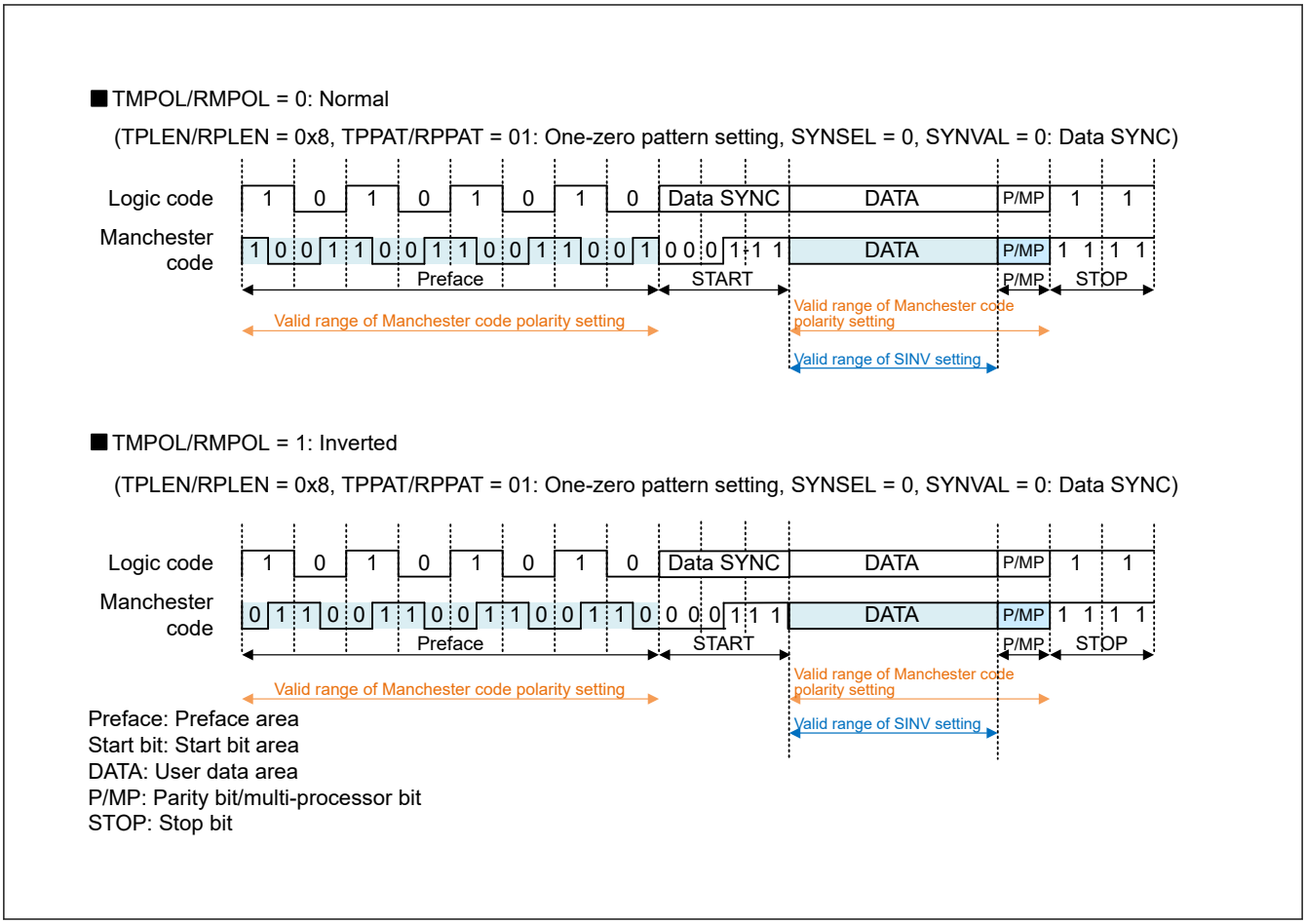


Figure 31.60 Valid Range of the Manchester Code Polarity Setting

31.5.11 Errors in Manchester Mode

There are the following errors in Manchester mode:

1. Parity error
2. Over run error
3. Framing error
4. Manchester error
5. Preface error
6. Start Bit error
7. Receive SYNC error

For errors (1) to (3), see section 31.3.9. Serial Data Reception in Asynchronous Mode (1) because they are the same as in Asynchronous mode.

Each errors are judged in each area, but they are reflected on flags and operations at the timing of 3/4-bit sampling of the STOP bit area. If a preface error or start bit error is detected, subsequent data will not be received. Therefore, no other error detection is performed, and the error flag holds the previous information.

Table 31.35 lists the states of the serial status register when detecting errors and judgment about whether to store data in the RDR.

Table 31.36 lists the errors that can be detected in each area of a Manchester frame.

If a Preface error or Start bit error is detected, subsequent data will not be received. Therefore, no other error detection is performed, and the error flag holds the result of the previous frame reception. Also, if an error is detected in the previous

frame, data will not be received, but errors in the pre-face area and start bit area will update that flag. Table 31.37 shows the flags and actions in this case.

(4) Manchester error

A Manchester error is generated when a Manchester code error is detected.

In Manchester code, there must be an edge (transition) in the center of the bit.

In the data area of a received frame (including the parity/multi-processor bit), the values of the 1/4-bit and 3/4-bit sampling points are checked in each received 1-bit data, and a Manchester code error is determined if these two values match.

If a Manchester code error is detected, the Manchester error flag (MSR.MER) is asserted.

If a Manchester error occurs, it is handled as an interrupt source and event source. If a Manchester error is detected, the next reception is not performed until the corresponding error flag is cleared.

(5) Preface error

A preface error is generated when the preface pattern does not match or a Manchester code error is detected in the preface area. If a preface error is detected, the preface error flag (MSR.PFER) is asserted.

It is possible to set whether to use this error flag as an interrupt source with the setting of the MCR register.

When MCR.PFEREN = 1, a preface error is handled as an interrupt source or event source. If a preface error is detected, the next reception is not performed until the corresponding error flag is cleared.

When MCR.PFEREN = 0, a preface error is not handled as an interrupt source or event source, and the next reception is not halted. However, a preface error is notified to MSR.PFER.

(6) Start bit error

A start bit error is generated when a mismatch is detected between the start bit area in the received frame and the preset start bit pattern. Upon detection of a start bit error, a start bit error flag (MSR.SBER) is asserted.

It is possible to set whether to use the start bit error as an interrupt source with the setting of the MCR register.

When MCR.SBEREN = 1, a start bit error is handled as an interrupt source or event source. If a start bit error is detected, the next reception is not performed until the corresponding error flag is cleared.

When MCR.SBEREN = 0, a start bit error is not handled as an interrupt source or event source, and the next reception is not halted. However, a start bit error is notified to MSR.SBER.

(7) Receive SYNC error

When the receive retiming function described in section 31.5.9. Receive Retiming is enabled, the receive retiming operation is performed.

If no edges are detected within the receive retiming range (SyncError area in Figure 31.59) when receive timing operation is being performed, a receive SYNC error is generated. Upon detection of a receive SYNC error, a receive SYNC error flag (MSR.SYER) is asserted. In areas not subject to retiming, receive SYNC errors are not detected.

The preface area[*1], the start bit area[*1],[*2], and the data area (excluding the stop bit) for which receive retiming operation is performed are checked.

It is possible to set whether to use the receive SYNC error as an interrupt source with the setting of the MCR register.

When MCR.SYEREN = 1, a receive SYNC error is handled as an interrupt source or event source. If a receive SYNC error is detected, the next reception is not performed until the corresponding error flag is cleared.

When MCR.SYEREN = 0, a receive SYNC error is not handled as an interrupt source or event source, and the next reception is not halted. However, a receive SYNC error is notified to MSR.SYER.

Note 1. In the case of a frame that starts with a pattern that expects the first half of the bit to be High, it is excluded from retiming.

Note 2. In the start bit area, when there is no preface length and 3 bit start bit is set, it is not subject to retiming.

Also, the 1st bit and the 2nd bit in the start bit area when 3 bit start bit is set are not subject to retiming.

Table 31.35 Flags in the CSR Register and Receive Data Handling in Manchester Mode

Flag in the CSR register			Flag in the MRS register				Received data	Received error status (SCIn_ERI interrupt / event generation)
ORER	FER	PER	MER	SBER*1	PFER*1	SYER		
0	0	0	0	0	0	0	transfer to RDR	No error
0	1	0	0	0	0	0	transfer to RDR	Framing error
0	0	1	0	0	0	0	transfer to RDR	Parity error
0	1	1	0	0	0	0	transfer to RDR	Framing error + Parity error
0	0	0	1	0	0	0	transfer to RDR	Manchester error
0	1	0	1	0	0	0	transfer to RDR	Framing error + Manchester error
0	0	1	1	0	0	0	transfer to RDR	Parity error + Manchester error
0	1	1	1	0	0	0	transfer to RDR	Framing error + Parity error + Manchester error
1	0	0	0	0	0	0	Lost	Overrun error
1	1	0	0	0	0	0	Lost	Overrun error + Framing error
1	0	1	0	0	0	0	Lost	Overrun error + Parity error
1	1	1	0	0	0	0	Lost	Overrun error + Framing error + Parity error
1	0	0	1	0	0	0	Lost	Overrun error + Manchester error
1	1	0	1	0	0	0	Lost	Overrun error + Framing error + Manchester error
1	0	1	1	0	0	0	Lost	Overrun error + Parity error + Manchester error
1	1	1	1	0	0	0	Lost	Overrun error + Framing error + Parity error + Manchester error
0	Combination of above			0	0	1	transfer to RDR	Errors above + Receive SYNC error*2
1				0	0	1	Lost	Errors above + Receive SYNC error*2
hold	hold	hold	hold	0	1	0	Lost	Preface error*3
hold	hold	hold	hold	1	0	0	Lost	Start bit error*3
hold	hold	hold	hold	0	1	1	Lost	Preface error*3 + Receive SYNC error*2
hold	hold	hold	hold	1	0	1	Lost	Start bit error*3 + Receive SYNC error*2

Note 1. Start bit error and Preface error never become 1 at the same time.
Note 2. When MCR.SYEREN = 1, SCIn_ERI interrupt / event is generated by SYER factor.
Note 3. If MCR.PFEREN = 1 or MCR.SBEREN = 1, an SCIn_ERI interrupt / event is generated when the corresponding flag is set.

Table 31.36 Errors Detectable in Each Area

	Preface error (PFER)	Start Bit error (SBER)	Manchester error (MER)	Receive SYNC error (SYER)	Parity error (PER)	Framing error (FER)
Preface area	✓	—	—*1	✓*2	—	—
Start Bit area	—	✓	—	✓*2	—	—
Data area	—	—	✓	✓	—	—
Parity area	—	—	✓	✓	✓	—
Multi-processor area	—	—	✓	✓	—	—
Stop Bit area	—	—	—	—	—	✓

Note: ✓: Detected, —: Not detected
Note 1. When an Manchester code error occurs in the preface area, it is defined as a preface error.
Note 2. It may not be subject to Receive SYNC error detection. For details see the text section 31.5.11. Errors in Manchester Mode (7)

Table 31.37 Operation status due to presence / absence of error in previous frame and operation status list in multiprocessor mode (1 of 2)

Previous frame	Each area of the Frame					PFEREN	SBEREN	SYEREN	received data	Error flag	Interrupt request	Event signal
	preface	start bit	data	parity	stop							
No Error	PFER	No Error	Don't Care	Don't Care	Don't Care	0	Don't Care	Don't Care	Lost	set PFER*1	not output	not output
	No SYER*1					1					output	output
	No Error	SBER	Don't Care	Don't Care	Don't Care	Don't Care	0	Don't Care	Lost	set SBER*1	not output	not output
		No SYER*1					1				output	output
	SYER	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0	transfer to RDR	set SYER	not output	not output
	No PFER							1	Lost		output	output
	No Error	SYER	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0	transfer to RDR	set SYER	not output	not output
		No SBER						1	Lost		output	output
	No Error	No Error	SYER		No Error	Don't Care	Don't Care	0	transfer to RDR	set SYER	not output	not output
								1			output	output
	No Error	No Error	MER		No Error	Don't Care	Don't Care	Don't Care	transfer to RDR	set MER	output	output
	No Error	No Error	Don't Care	PER	No Error	Don't Care	Don't Care	Don't Care	transfer to RDR	set PER	output	output
	No Error	No Error	Don't Care	Don't Care	FER	Don't Care	Don't Care	Don't Care	transfer to RDR	set FER	output	output
	There is some error ORER					Don't Care	Don't Care	Don't Care	Lost	set some flags*2	output	output
	No Error	No Error	No Error	No Error	No Error ORER	Don't Care	Don't Care	Don't Care	Lost	set ORER	output	output

Table 31.37 Operation status due to presence / absence of error in previous frame and operation status list in multiprocessor mode (2 of 2)

Previous frame	Each area of the Frame					PFEREN	SBEREN	SYEREN	received data	Error flag	Interrupt request	Event signal
	preface	start bit	data	parity	stop							
some error*3 *6	PFER No SYER*1	No Error	Don't Care	Don't Care	Don't Care	0	Don't Care	Don't Care	Lost	set PFER*1	output*4	not output*5
						1						
	No Error	SBER No SYER*1	Don't Care	Don't Care	Don't Care	Don't Care	0	Don't Care		set SBER*1		
							1					
	SYER No PFER	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0		set SYER		
								1				
	No Error	SYER No SBER	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0		set SYER		
								1				
	No Error	No Error	SYER		No Error	Don't Care	Don't Care	0		don't set any flags		
								1				
	No Error	No Error	MER		No Error	Don't Care	Don't Care	Don't Care				
	No Error	No Error	Don't Care	PER	No Error	Don't Care	Don't Care	Don't Care				
	No Error	No Error	Don't Care	Don't Care	FER	Don't Care	Don't Care	Don't Care				
	There is some error ORER					Don't Care	Don't Care	Don't Care				
	No Error	No Error	No Error	No Error	No Error ORER	Don't Care	Don't Care	Don't Care				

Note 1. If SYER is detected, the SYER flag is also set. Other operations are as shown in this table.
Note 2. Other detected error flags including ORER are also set.
Note 3. If all the error flags are cleared before the STOP bit is judged, the operation will be the same as the case where there is no error in the previous frame of this table.
Note 4. Since the SCIn_ERI interrupt request is level output, it remains active due to errors in the previous frame regardless of the presence or absence of error in the relevant frame.
Note 5. Since the error cause is continuously detected, the SCIn_ERI event is not newly output regardless of the presence or absence of errors in the relevant frame.
Note 6. For PFER, SBER, and SYER, when each enable bit is set to disable, it is treated as no error.

Table 31.38 Operation when MPIE = 1 in multi-processor mode (MPIE = 0)

MPB*1	Each area of the frame					PFEREN	SBEREN	SYEREN	received data	Error flag	Interrupt request	Event signal
	preface	start bit	data	parity	stop							
1	No Error	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	transfer to RDR	set some flags	output*2	output*2
	No PFER	No SBER	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0				
	SYER*3	SYER*3						1	Lost	don't set any flags	not output	not output
	PFER	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care				
	No Error	SBER	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care					

Note 1. If the received MPB bit is 0, it is not received the frame, and the operation is the same as lost of the reception data of this table.
Note 2. If no error is detected, SCIn_RXI interrupt request or event is output, and if it is detected, SCIn_ERI interrupt request or event is output.

Note 3. When SYER is detected in the preface area or the start bit area, the behavior of handling as an error depending on the SYEREN bit changes.

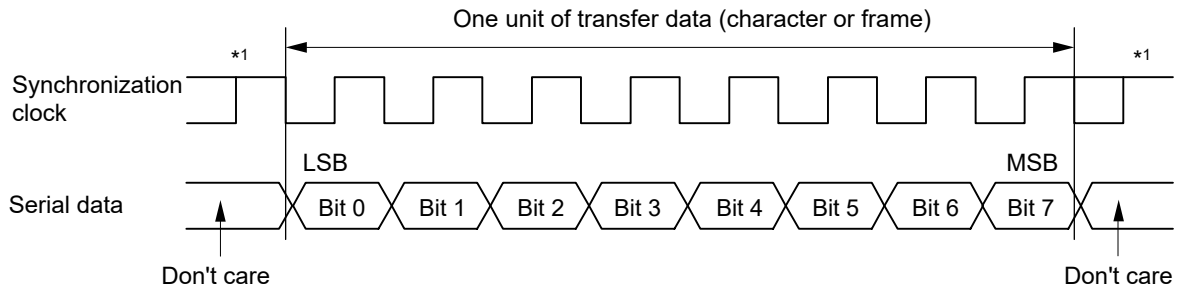
31.6 Operation in Clock Synchronous Mode

Figure 31.61 shows the data format for clock synchronous serial data communications.

In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. For single-character data transfer, data consists of 8-bit. In clock synchronous mode, no parity bit can be added.

In data transmission when CPHA = 1 and CPOL = 1, the SCI outputs data from one falling edge of the synchronization clock to the next falling edge. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the transmission line holds the last bit as output state. When the CPHA bit is 0 in slave mode, the transmission line holds the first bit output state.

Within the SCI, the transmitter and receiver are independent units, enabling full-duplex communications by using a shared communication clock of the transmitter and the receiver. Furthermore, because both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.



Note 1. Holds a high level except during continuous transfer.

Figure 31.61 Data format in clock synchronous serial communications with LSB-first order

31.6.1 Clock

If the maximum speed of SCK = 1/2TCLK is set in Clock Synchronous and Simple SPI mode, Do not make PCLK less than half the speed of TCLK. If PCLK is made slower than this, malfunction may occur.

1. When the internal clock is selected
   When the SCI operates on an internal clock (CCR3.CKE[1:0] bits are set to 00b or 01b (master mode)), the synchronization clock is output from the SCKn pin. Eight synchronization clock pulses are output during single-character transmission/reception. When no data transfer is performed, the clock is held at high level.*1 In transmission-only or transmission/reception, the synchronization clock is not output unless transmit data is prepared.
   When the internal clock is selected, the clock with a delay from the SCKn signal is used for the master reception sampling clock. This ensures the data setup time and hold time for high-speed communication.

   Note 1. The signal is held high while (CCR3.CPHA = 0 and CCR3.CPOL = 1) or (CCR3.CPHA = 1 and CCR3.CPOL = 1). It is held low while (CCR3.CPHA = 0 and CCR3.CPOL = 0) or (CCR3.CPHA = 1 and CCR3.CPOL = 0).

2. When the external clock is selected
   When the CCR3.CKE[1:0] bits are set to 10b or 11b (slave mode), data is transmitted and received using the external clock that is input from the SCKn pin.

31.6.2 CTS and RTS Functions

In the CTS function, the CTSn_RTSn pin input controls the start of data reception or transmission when the clock source is the internal clock. Setting the CCR1.CTSE bit to 1 enables the CTS function. In clock synchronous communication, the CTS function can be used for the internal clock and the RTS function can be used for the external clock, so the CTS function and RTS function cannot be used at the same time. When the CTS function is enabled, setting the CTSn_RTSn pin low causes data reception or transmission to start.

Setting the CTSn_RTSn pin high while the data transmission or reception is in progress does not affect transmission or reception of the current frame.

In the RTS function, the CTSn_RTSn pin output is used to request the start of data reception or transmission when the clock source is an external synchronizing clock. The CTSn_RTSn output goes low when serial communication is enabled. Conditions for output of the CTSn_RTSn low and high are shown as follows:

[Conditions for low output]

Satisfaction of all the following conditions:

Non-FIFO selected when all of the following conditions are satisfied

- The value of the CCR0.RE bit or the CCR0.TE bit is 1
- Next serial communication is enabled.
  - No receive data is present before reading and not receiving. (when CCR0.RE bit = 1)
  - When the transmission data written in TDR is ready for transmission.*1 (when CCR0.TE bit = 1)
- The CSR.ORER flag is 0

Note 1. The CTSn_RTSn pin will be High after starting transmission.

FIFO selected when all of the following conditions are satisfied

- The value of the CCR0.RE bit or the CCR0.TE bit is 1
- Next serial communication is enabled.
  - The number of receive data stored in the receive FIFO (RDR register) is less than the value set in FCR.RSTRG[4:0] (when CCR0.RE bit = 1)
  - When the transmission data written in the transmission FIFO (TDR register) is ready for transmission.*1 (when CCR0.TE bit = 1)
- The CSR.ORER flag is 0

Note 1. The CTSn_RTSn pin will be High after the last data transmission starts.

[Condition for high output]

- The conditions for low output are not satisfied

31.6.3 SCI Initialization in Clock Synchronous Mode

Before transmitting and receiving data, start by writing the initial value 0x00 to the CCR0 register, then continue through the SCI initialization procedure given in the sections describing non-FIFO and FIFO selection in section 31.6.2. CTS and RTS Functions. Anytime the operating mode or transfer format is to be changed, the CCR0.TE and CCR0.RE must write to 0 before the change can be made.

Note: Setting the CCR0.RE bit to 0 initializes neither the ORER, FER, PER, and RDRF flags in CSR nor the RDR register. When the TE bit is set to 0, the TEND flag for the selected FIFO buffer is not initialized.

Note: Switching the value of the CCR0.TE bit from 1 to 0 when the CCR0.TIE bit is 1 generates an SCIn_TXI interrupt request.

Table 31.39 Example flow of SCI initialization in clock synchronous mode with non-FIFO selected (1 of 2)

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set the CCR0.TEIE, TIE, RIE, TE, and RE bits to 0*1. If you have not changed from the initial settings, you can skip this step.
3	Set FCR	Set the TFRST and RFRST to 1 to empty FIFO. Set the TTRG[4:0], RTRG[4:0], and RSTRG[4:0] bits.

Table 31.39 Example flow of SCI initialization in clock synchronous mode with non-FIFO selected (2 of 2)

No.	Step Name	Description
4	Set CCR3 except MOD[2:0]	Set CCR3 except of communication mode. ● FIFO use/no-use ● Transmission/reception format ● Clock setting ● Leave unused bits (CHR[1:0], STP, RXDSEL, MP, DE, ACS0, GM, BLK) at their initial values.
5	Set CCR3.MOD[2:0]	Set communication mode (MOD[2:0] = 010b)*2.
6	Set CCR2	Select clock, set bit rate *3. Leave unused bits (BCP[2:0], ABCS, ABCSE, BRME, MDDR[7:0]) at their initial values.
7	Set CCR1	Set up the loop-back function, communication pin status and the CTS/RTS function.
8	Set CCR4	Set up the adjust sampling timing function. Leave unused bits (CMPD[8:0]) at their initial values.
9	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
10	Set CFCLR, FFCLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC FFCLR.BRKC, DRC
11	Set CCR0	Set the TE or RE bit to 1.*1 *4 To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Setting the TE and RE bits allows TXDn and RXDn to be used.
12	Initialization completion	—

Note 1. In simultaneous transmit and receive operations, the TE and RE bits in CCR0 should both be to 0 or set to 1 simultaneously.
Note 2. Set CPOL and CPHA before setting the communication mode.
Note 3. If you use an external clock, you do not need to set it.
Note 4. When using the internal clock (master), the setting of reception only is prohibited.

31.6.4 Serial Data Transmission in Clock Synchronous Mode

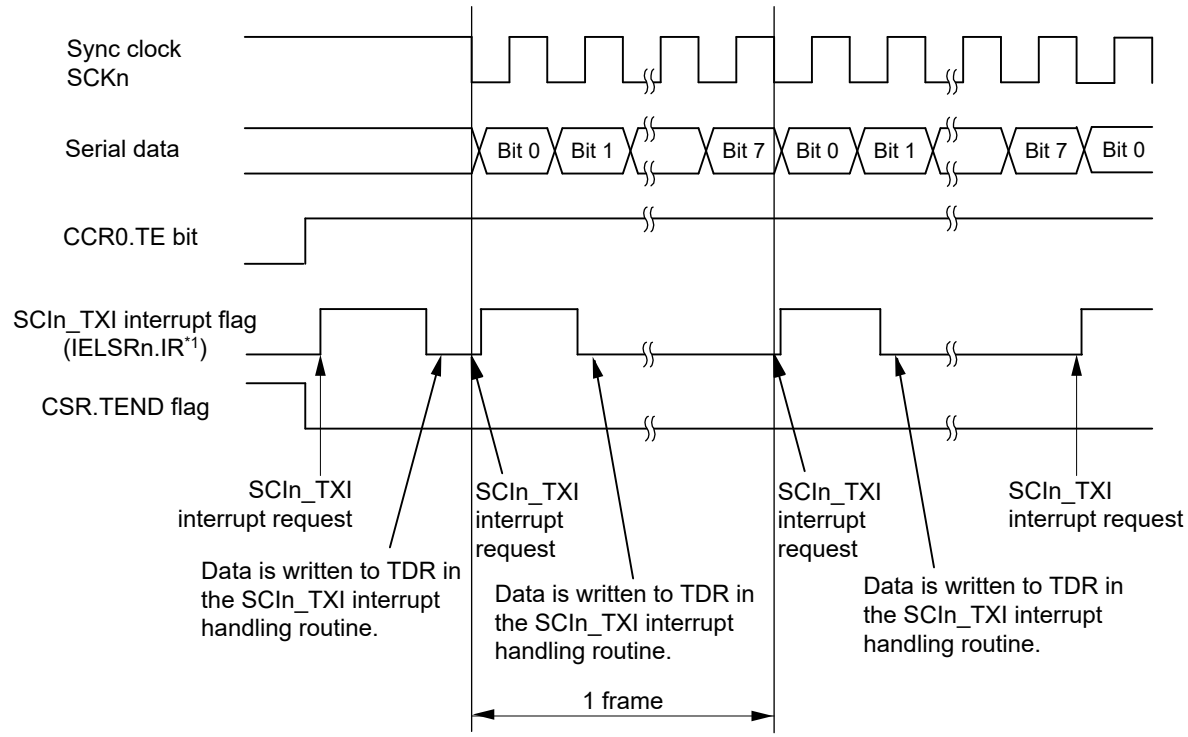
(1) Non-FIFO selected

Figure 31.62, Figure 31.63, and Figure 31.64 show examples of serial transmission in clock synchronous mode.

In serial data transmission, the SCI operates as follows:

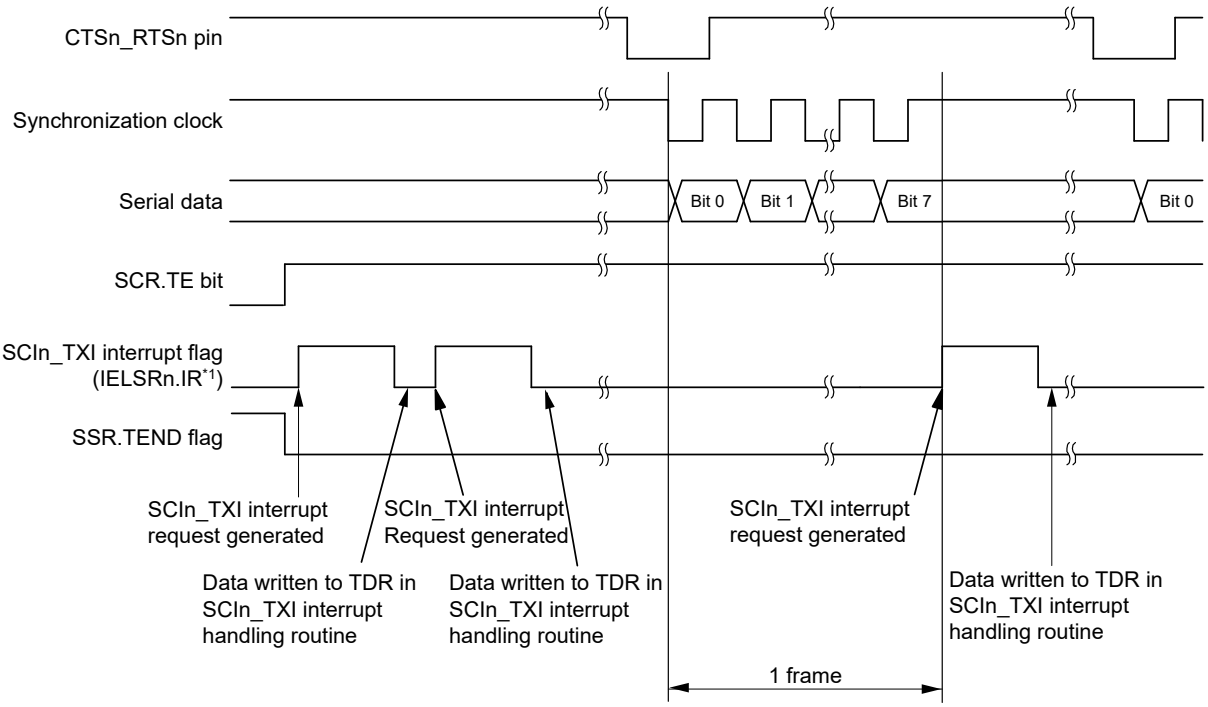
1. The SCI transfers data from the TDR register to the TSR register when data is written to TDR in the SCIn_TXI interrupt handling routine. When starting data transmission, set the CCR0.TIE bit and the CCR0.TE bit to 1 simultaneously by a single instruction. Then a SCIn_TXI interrupt request is generated.
2. After transferring data from TDR to TSR, the SCI starts transmission. When the CCR0.TIE bit is set to 1, an SCIn_TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to TDR in the SCIn_TXI interrupt handling routine before transmission of the current transmit data finishes. When SCIn_TEI interrupt requests are in use, set the CCR0.TIE bit to 0 and the CCR0.TEIE bit to 1 after the last of the data to be transmitted is written to the TDR register from the handling routine for SCIn_TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when the clock output mode is specified and in synchronization with the input clock when the use of an external clock is specified. Output of the clock signal is suspended until the input CTS signal is low when the CCR1.CTSE bit is 1 (CTS function enabled).
4. The SCI checks for update to the TDR register on output of the last bit.
5. When the TDR register is updated, the next transmit data is transferred from TDR to TSR, and serial transmission of the next frame starts.
6. If TDR is not updated, the CSR.TEND flag is set to 1. The TXDn pin retains the output state of the last bit. If the CCR0.TEIE bit is 1, an SCIn_TEI interrupt request is generated and the SCKn pin is held high.

Figure 31.62, Figure 31.63, and Figure 31.64 show examples of serial data transmission.



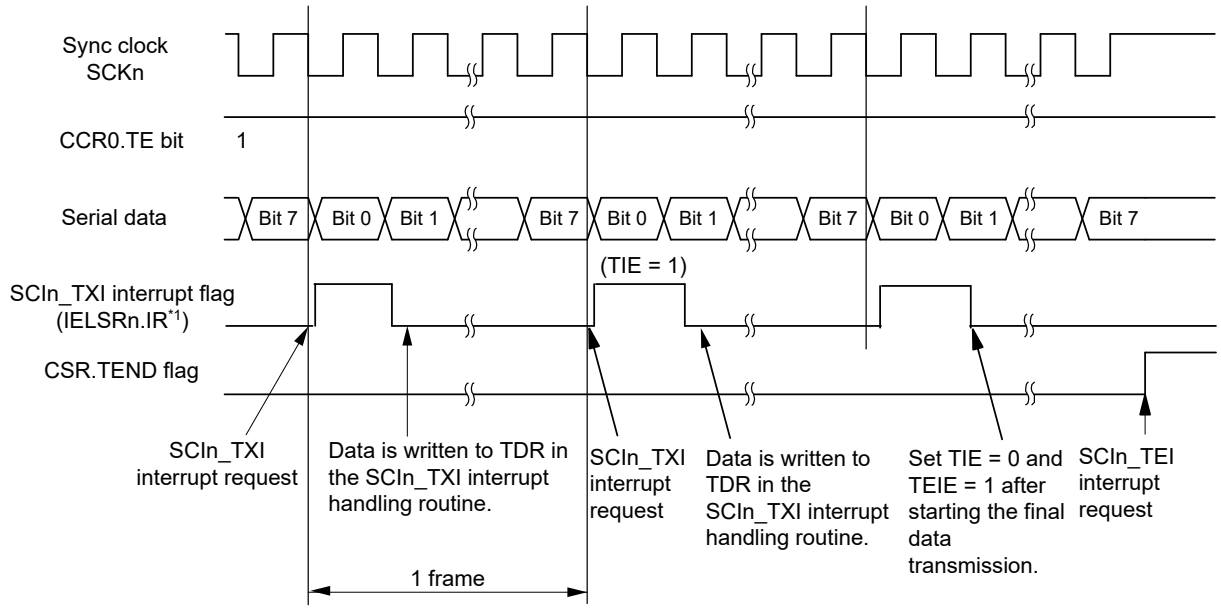
Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.62 Example of serial data transmission in clock synchronous mode when the CTS function is not used at the beginning of transmission



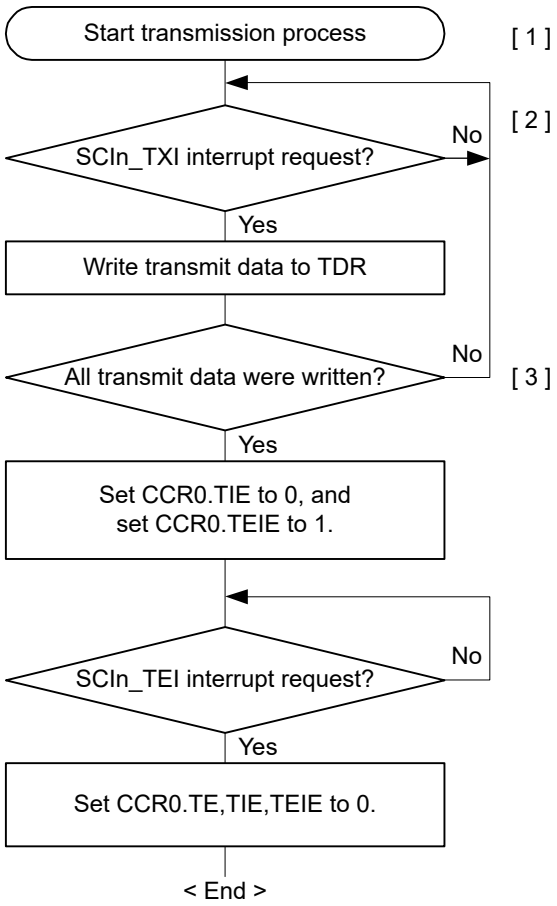
Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.63 Example of serial data transmission in clock synchronous mode when the CTS function is used at the beginning of transmission



Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.64 Example of serial data transmission in clock synchronous mode from the middle of transmission until transmission completion



[1] Start transmission process

[2] Write transmit data to TDR by using SCIn_TXI:
The transmit-data-empty-interrupt(SCIn_TXI) is Generated when transmission-data forwards to TSR from TDR. Write transmit data to TDR once by using a SCIn_TXI.

[3] Serial transmission continuation procedure:
To continue serial transmission write transmit data to TDR by using a SCIn_TXI.
Transmit data can also be written by activating the DMAC or DTC.
If you use SCIn_TEI, set CCR0.TIE = 0 and CCR0.TEIE = 1 after writing the last-transmit-data to TDR.

Note: When the external clock is in use (the value of the CCR3.CKE[1:0] is 10b or 11b), the rising edge on the SCK pin for the last bit sets the CSR.TEND flag to 1. Setting TE to 0 immediately after this, then it may lead to insufficient received data hold time on the receiver side.

Figure 31.65 Example flow of serial transmission in clock synchronous mode with non-FIFO selected

(2) FIFO selected

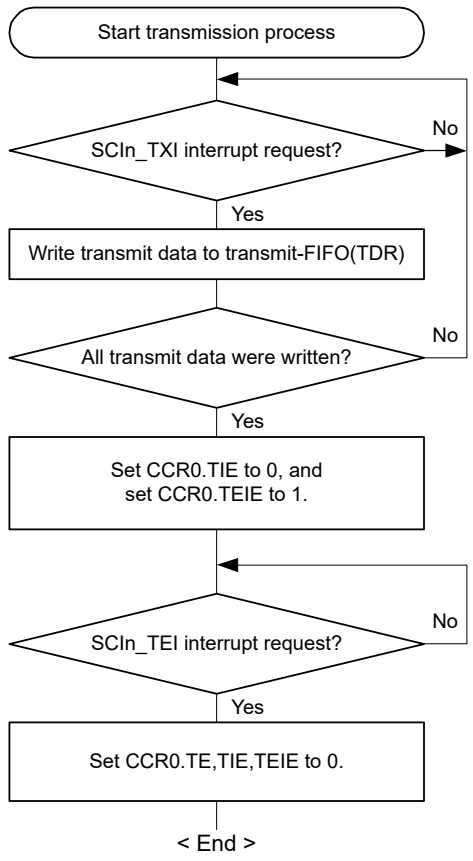
Figure 31.66 shows an example of serial transmission in clock synchronous mode with FIFO selected.

In serial data transmission, the SCI operates as follows:

1. The SCI transfers data from the transmit-FIFO (TDR register) to the TSR register when data is written to transmit-FIFO (TDR register) in the SCIn_TXI interrupt handling routine. The amount of data that can be written to transmit-FIFO (TDR register) is 16 - FTSR.T[5:0] bytes. In addition, when starting data transmission, set the CCR0.TIE bit and the CCR0.TE bit to 1 simultaneously by a single instruction. Then a SCIn_TXI interrupt request is generated.
2. After transferring data from transmit-FIFO (TDR register) to TSR, the SCI starts transmission. When the amount of transmit data written in transmit-FIFO (TDR register) is equal to or less than the specified transmit triggering number, the CSR.TDRE is set to 1. When the CCR0.TIE bit is set to 1, an SCIn_TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to transmit-FIFO (TDR register) in the SCIn_TXI interrupt handling routine before transmission of the current transmit data has finished. When SCIn_TEI interrupt requests are in use, set the CCR0.TIE bit to 0 and the CCR0.TEIE bit to 1 after the last of the data to be transmitted is written to the transmit-FIFO (TDR register) from the handling routine for SCIn_TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when the clock output mode is specified and in synchronization with the input clock when the use of an external clock is specified. Output of the clock signal is suspended until the input CTS signal is low when the CCR1.CTSE bit is 1 (CTS function enabled).
4. The SCI checks whether non-transmitted data remains in transmit-FIFO (TDR register)*1 on output of the stop bit.
5. When data is remaining in the transmit-FIFO (TDR register), the next transmit data is transferred from transmit-FIFO (TDR register) to TSR and serial transmission of the next frame starts.

6. If no data is remaining in the transmit FIFO (TDR register), the CSR.TEND flag is set to 1. The TXDn pin retains the output state of the last bit. If the CCR0.TEIE bit is 1, an SCIn_TEI interrupt request is generated and the SCKn pin is held high.

Note 1. The number of unsent transmit data stored in the TDR register (transmit FIFO) can be monitored by reading the FTSR.T[5:0] bits.



[1] [1] Start transmission process

[2] [2] Write transmit data to transmit-FIFO(TDR) by using SCIn_TXI: The transmit-data-empty-interrupt(SCIn_TXI) is generated when transmission-data forwards to TSR from the transmit-FIFO(TDR) and the number of transmit-data written in the transmit-FIFO is equal to or less than the transmit-FIFO threshold.

[3] [3] Serial transmission continuation procedure: To continue serial transmission, write transmit data to the transmit-FIFO(TDR) by using a SCIn_TXI interrupt and clear CSR.TDRE flag to 0. Transmit data can also be written to the transmit- FIFO(TDR) by activating the DMAC or DTC. In this case, the TDRE flag is cleared automatically, so please don't operate the TDRE flag. If you use SCIn_TEI, set CCR0.TIE = 0 and CCR0.TEIE = 1 after writing the last-transmit-data to TDR.

Note: When the external clock is in use (the value of the CCR3.CKE[1:0] is 10b or 11b), the rising edge on the SCK pin for the last bit sets the CSR.TEND flag to 1. Setting CCR0.TE to 0 immediately after this, then it may lead to insufficient received data hold time on the receiver side.

Figure 31.66 Example flow of serial transmission in clock synchronous mode with FIFO selected

31.6.5 Serial Data Reception in Clock Synchronous Mode

(1) Non-FIFO selected

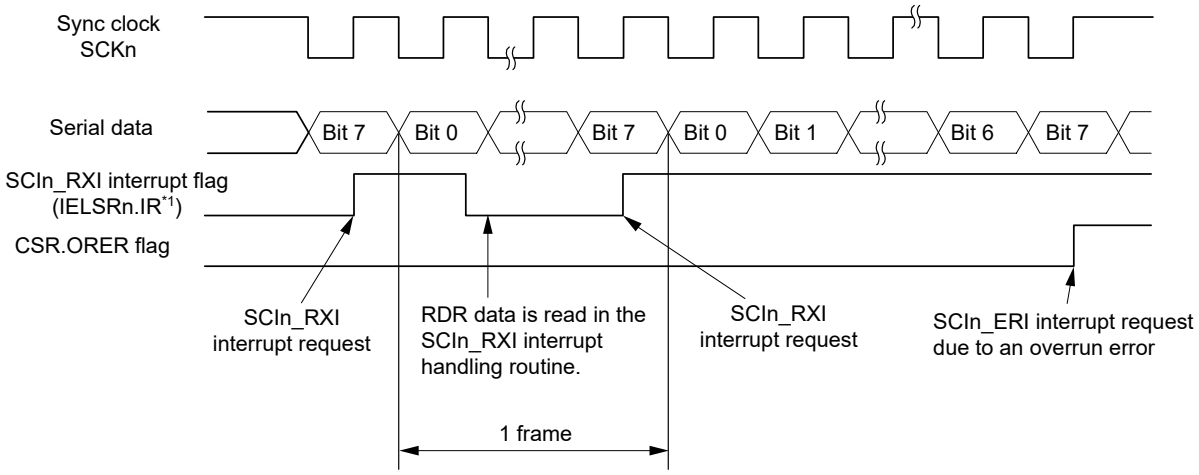
Figure 31.67 and Figure 31.68 show examples of SCI operation for serial reception in clock synchronous mode.

In serial data reception, the SCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

1. When the value of the CCR0.RE bit becomes 1, the CTSn_RTSn pin goes low (when the RTS function is used).
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in the RSR register.
3. If an overrun error occurs, the CSR.ORER flag is set to 1. If the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to the RDR register.
4. When reception completes successfully, receive data is transferred to the RDR register. If the CCR0.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the received data transferred to the

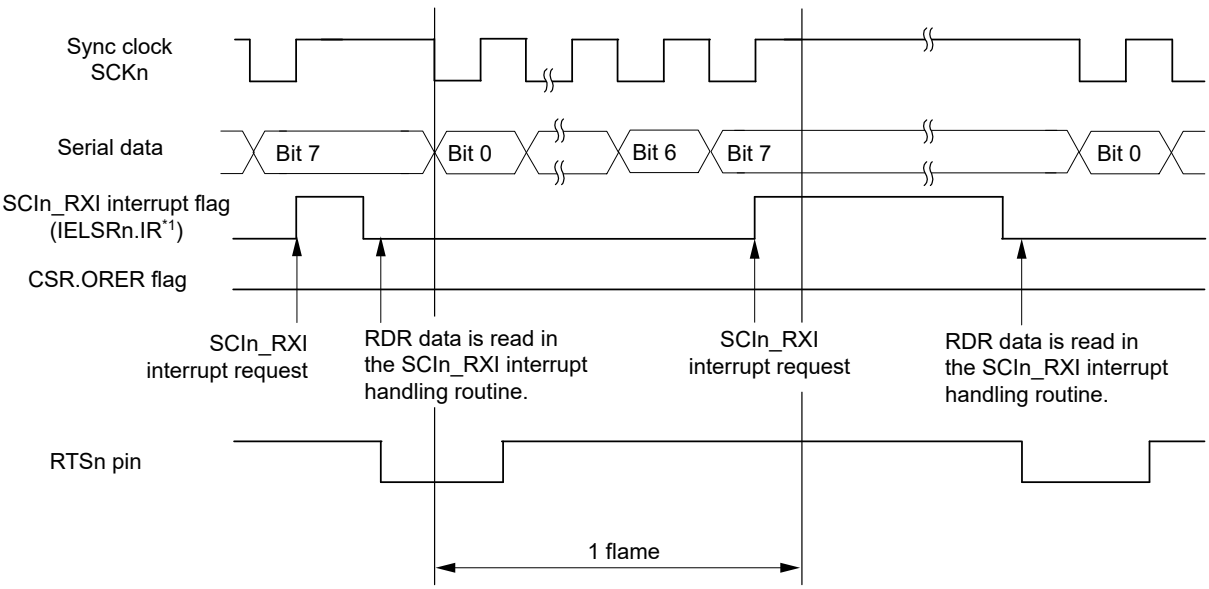
RDR register in the SCIn_RXI interrupt handling routine before reception of the next receive data completes. Reading the received data that is transferred to RDR causes the CTSn_RTSn pin to output low (when the RTS function is used).

If you want to prevent the CTSn_RTSn pin output from turning low level after the final data is received, clear the CCR0.RE bit to 0 and then read the RDR register.



Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.67 Example operation for serial reception in clock synchronous mode (1) when the RTS function is not used

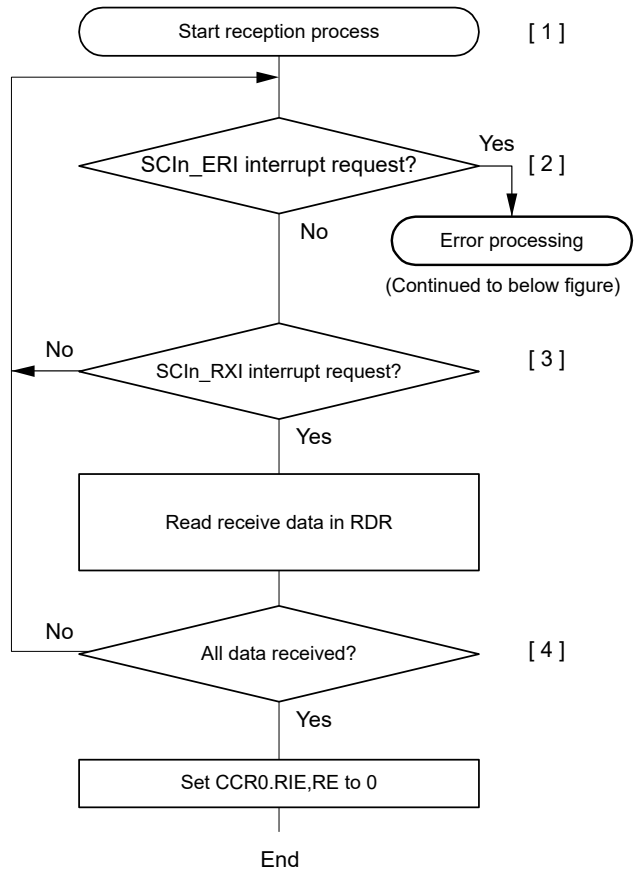


Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.68 Example operation for serial reception in clock synchronous mode (2) when RTS function is used

Data transfer cannot resume while the receive error flag is 1. Therefore, clear the ORER, FER, and PER flags in the CSR register to 0 before resuming data reception. Additionally, always read the RDR register during overrun error processing. When a data reception is forced to terminate by a 0 write to the CCR0.RE bit during operation, read the RDR register because received data that is not yet read might be left in the RDR register.

Figure 31.69 shows an example flow of serial data reception.

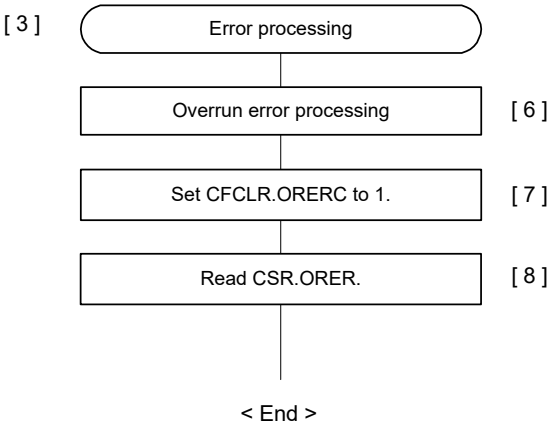


[1] Start reception process

[2] Receive error processing:
If a receive error occurs, after performing the appropriate error processing, be sure to CSR.ORER is 0. Reception cannot be resumed if ORER is 1.

[3] Read the receive data in RDR once in the SCIn_RXI interrupt handling routine.

[4] Serial reception continuation procedure:
To continue serial reception, before the MSB bit (bit7) of the current frame is received, read data from RDR in the SCIn_RXI interrupt handling routine. The RDR data can also be read by activating the DMAC or DTC.



[6] Overrun error processing:
Read RDR.

[7] Clear the error flag:
Set the flag clear bit corresponding to the reception error to 1. Done this step and step [6], the next frame can be received normally.

[8] Confirming that the error flag is cleared:
Confirm all error flags are cleared.

Figure 31.69 Example flow of serial reception in clock synchronous mode with non-FIFO selected

(2) FIFO selected

Figure 31.70 shows an example of serial reception in clock synchronous mode with FIFO selected.

In serial data reception, the SCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

1. When the value of the CCR0.RE bit becomes 1, the CTSn_RTSn pin goes low (when the RTS function is used).
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and transfers the received data to the receive-FIFO (RDR register).
3. If an overrun error occurs, the CSR.ORER flag is set to 1. If the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated. Received data is not transferred to the receive-FIFO (RDR register)*1.

4. When data reception completes successfully, the receive data is transferred to the receive-FIFO (RDR register)*1. The CSR.RDRF flag is set to 1 when the amount of the receive data stored in receive-FIFO (RDR register) is equal to or greater than the specified receive triggering number. If the CCR0.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous data reception is enabled by reading the receive data transferred to receive-FIFO (RDR register)*1 in the SCIn_RXI interrupt handling routine before an overrun error occurs. If the amount of received data that is transferred to receive-FIFO (RDR register) is less than the specified receive triggering number, the CTSn_RTSn pin goes low (when the RTS function is used).

Note 1. In clock synchronous mode, RDR.RDAT[8] is not used.

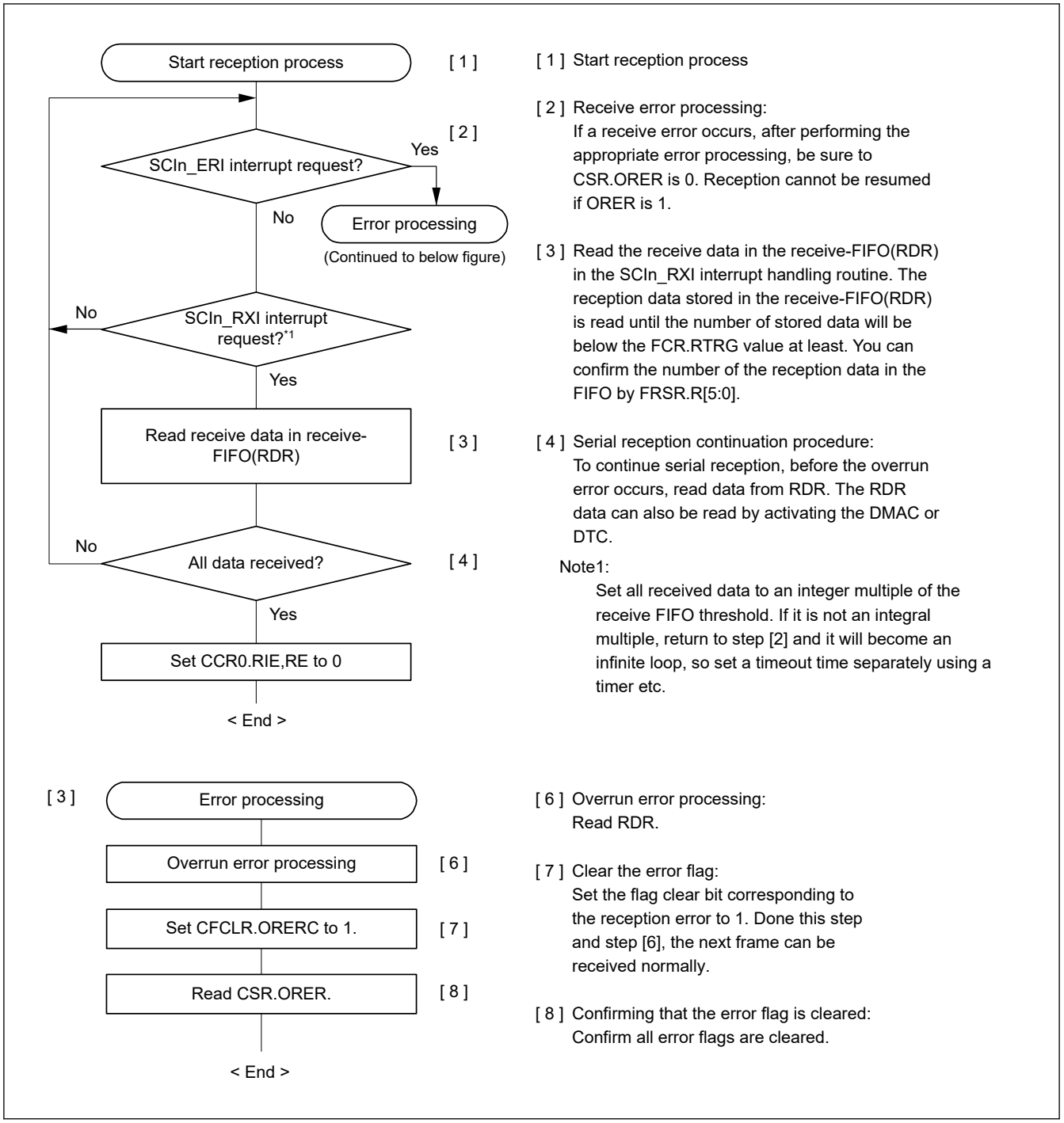


Figure 31.70 Example flow of serial reception in clock synchronous mode with FIFO selected

31.6.6 Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode

(1) Non-FIFO selected

Figure 31.71 shows an example flow of simultaneous serial transmit and receive operations in clock synchronous mode. After initializing the SCI, use the following procedure for simultaneous serial data transmit and receive operations.

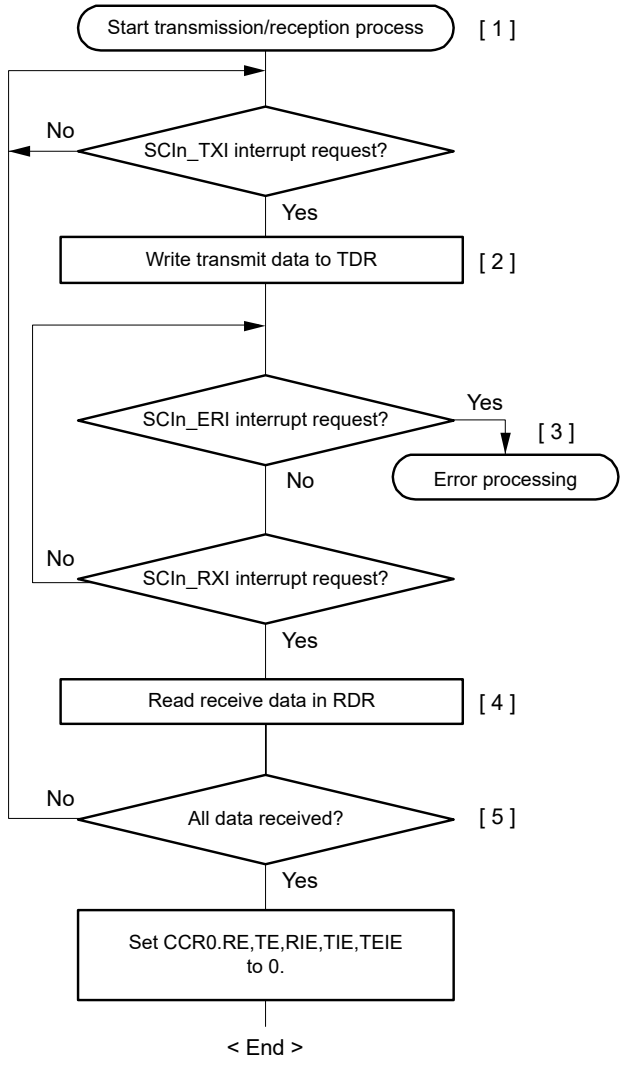
To switch from transmit mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the data transmission by verifying that the CSR.TEND flag is set to 1.
2. Initialize the CCR0 register, and then set the TIE, RIE, TE, and RE bits in the CCR0 register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the data reception.
2. Set the CCR0.TE and RE bits to 0, and then check that the receive error flag (ORER, FER, and PER) in the CSR register is 0.
3. Set the TIE, RIE, TE, and RE bits in the CCR0 register to 1 simultaneously by a single instruction.

When the RTS function is used in the concurrent transmission/reception operation, if you want to prevent the CTSn_RTSn pin output from turning to low after the final data is received as in the reception operation, clear the RE and TE bits in CCR0 to 0 simultaneously, and then read the RDR register.



[1] Start transmission/reception process

[2] Transmit data write:
Write transmit data to TDR in the SCIn_TXI Interrupt request handling routine.

[3] Receive error processing:
If a receive error occurs, after performing the appropriate error processing, be sure to CSR.ORER is 0. Reception cannot be resumed if ORER is 1.

[4] Reading receive data:
The reception data stored in RDR once in the SCIn_RXI interrupt request handling routine.

[5] Serial transmission/reception continuation procedure
To continue serial reception, before the MSB bit (bit7) of the current frame is received, read data from RDR in the SCIn_RXI interrupt handling routine. And before the MSB(bit7) of the current frame is transmitted, write transmit data to TDR in the SCIn_TXI interrupt handling. Transmit data can also be written by activating the DMAC or DTC. And the RDR data can also be read by activating the DMAC or DTC.

Figure 31.71 Example flow of simultaneous serial transmission and reception in clock synchronous mode with non-FIFO selected

(2) FIFO selected

Figure 31.72 shows an example flow of simultaneous serial transmit and receive operations in clock synchronous mode with FIFO selected.

After initializing the SCI, use the following procedure for simultaneous serial data transmit and receive operations.

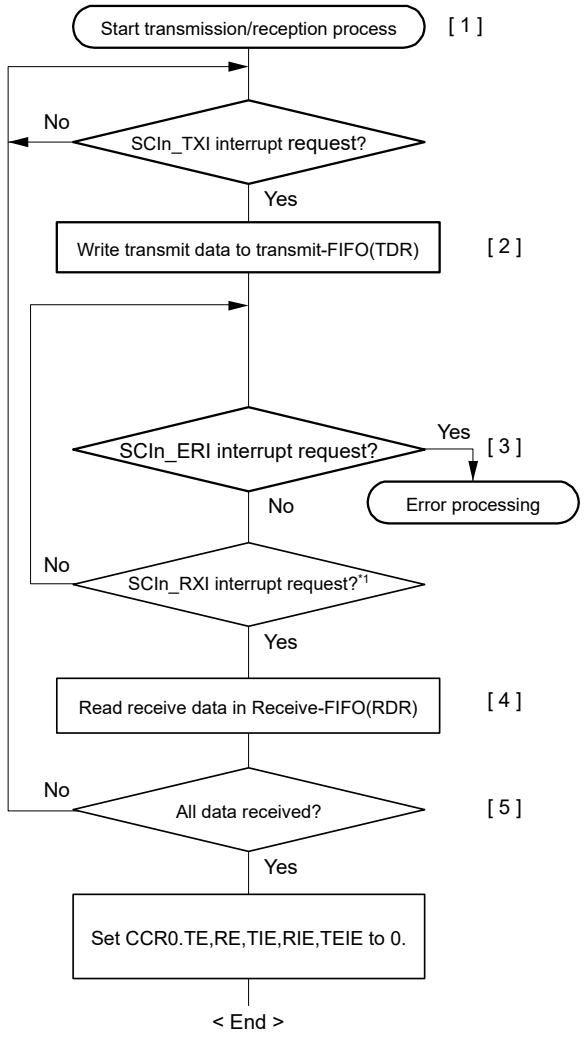
To switch from transmit mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the transmission by verifying that the CSR.TEND flag is set to 1.
2. Initialize the CCR0 register, then set the TIE, RIE, TE, and RE bits in the CCR0 register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the reception.
2. Set the CCR0.TE and RE bits to 0.
3. Check that the receive error flags (ORER, FER, and PER) in the CSR register are 0, and then set the TIE, RIE, TE, and RE bits in the CCR0 register to 1 simultaneously by a single instruction.

Since clock synchronous communication performs transmission and reception at the same time, make sure that the number of data to be transmitted and received is the same.



[1] Start transmission / reception process

[2] Transmit data write:
Write transmit data to Transmit-FIFO(TDR) in the SCIn_TXI Interrupt request handling routine.
Match the number of transmitted data with the FCR.RTRG.

[3] Receive error processing:
If a receive error occurs, after performing the appropriate error processing (Initialization such as stopping communication, reading all received data, clearing CSR.RDRF, etc.), be sure to ORER is 0. Reception cannot be resumed if ORER is 1.

[4] Read receive data:
Read the received data stored in receive-FIFO(RDR) for the number of the FCR.RTRG in the SCIn_RXI interrupt handling routine.

[5] Serial transmission/reception continuation procedure
To continue serial reception, before the overrun error occurs, read data from RDR and clear CSR.RDRF.
To continue serial transmission, write transmit data to the transmit-FIFO(TDR) by using a SCIn_TXI interrupt and clear TDRE flag to 0.
Transmit data can also be written to the transmit- FIFO(TDR) by activating the DMAC or DTC. And The RDR data can also be read by activating the DMAC or DTC. In this case, the TDRE and RDRF are cleared automatically, so please don't operate the TDRE And RDRF.

Note 1. Set all received data to an integer multiple of the receive FIFO threshold. If it is not an integral multiple, return to step [2] and it will become an infinite loop, so set a timeout time separately using a timer etc.

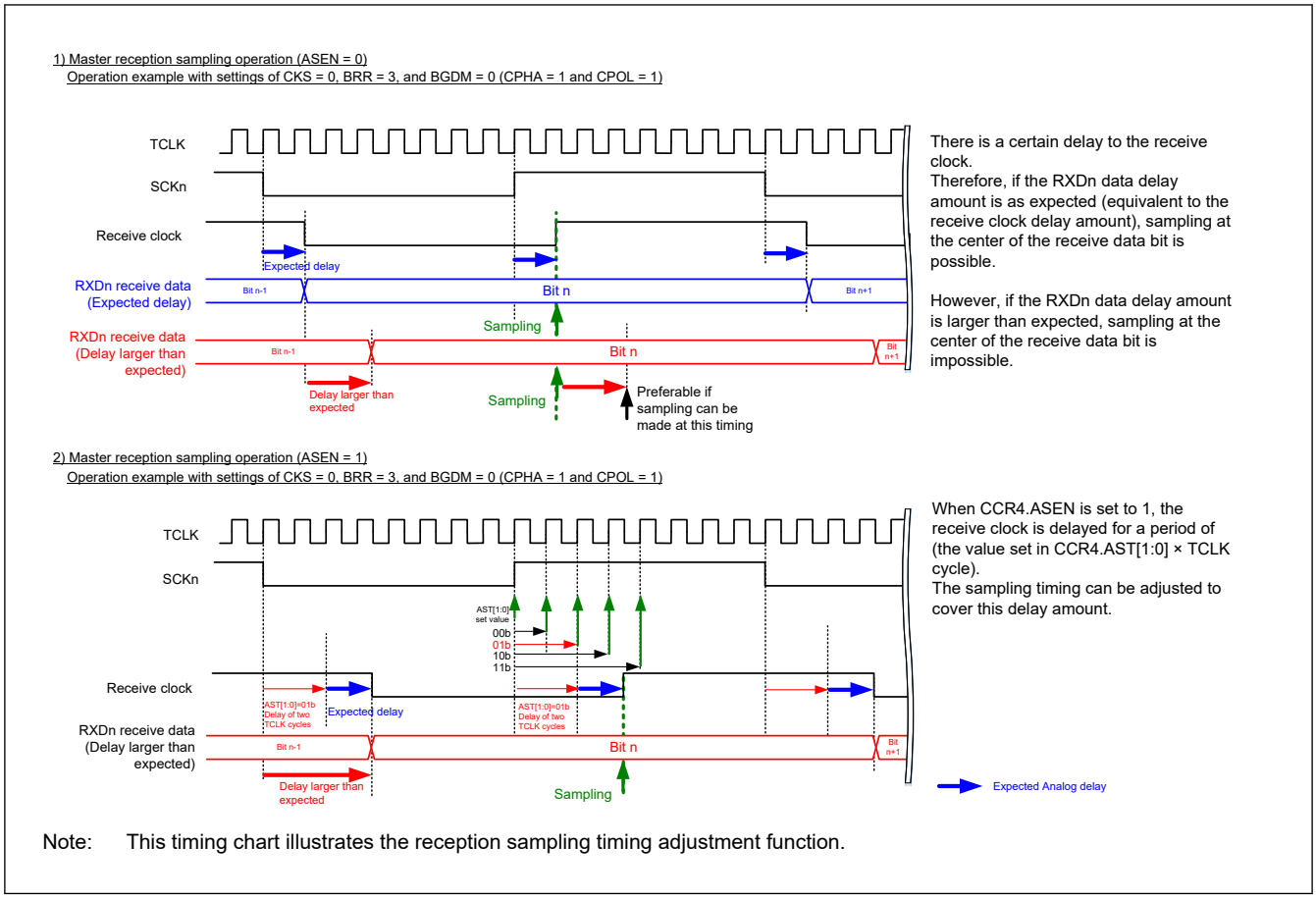
Figure 31.72 Example flow of simultaneous serial transmission and reception in clock synchronous mode with FIFO selected

31.6.7 Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with internal clock used

When the clock synchronous internal clock is used (master mode), the clock enabled by CCR4.SCKSEL bit is used as a reception sampling clock.

This function adjusts the reception sampling timing by delaying the clock by 1 to 4 TCLK and adding a digital delay.

Setting the CCR4.ASEN bit to 1 enables this function. The delay value is set in CCR4.AST[1:0].



Note: This timing chart illustrates the reception sampling timing adjustment function.

Figure 31.73 Reception Sampling Timing Adjustment Operation in Clock Synchronous Mode (Master)

31.7 Operation in Smart Card Interface Mode

The SCI supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards), as an extended function of the SCI.

Smart card interface mode can be selected using the appropriate register.

31.7.1 Example Connection

Figure 31.74 shows an example connection between a smart card (IC card) and the MCU. As shown in Figure 31.74, because the MCU communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor.

Setting the CCR0.TE and CCR0.RE bits to 1 with an IC card disconnected enables closed-loop transmission or reception, allowing self-diagnosis. To supply an IC card with the clock pulses generated by the SCI, input the SCKn pin output to the CLK pin of an IC card.

An output port of the MCU can be used to output a reset signal.

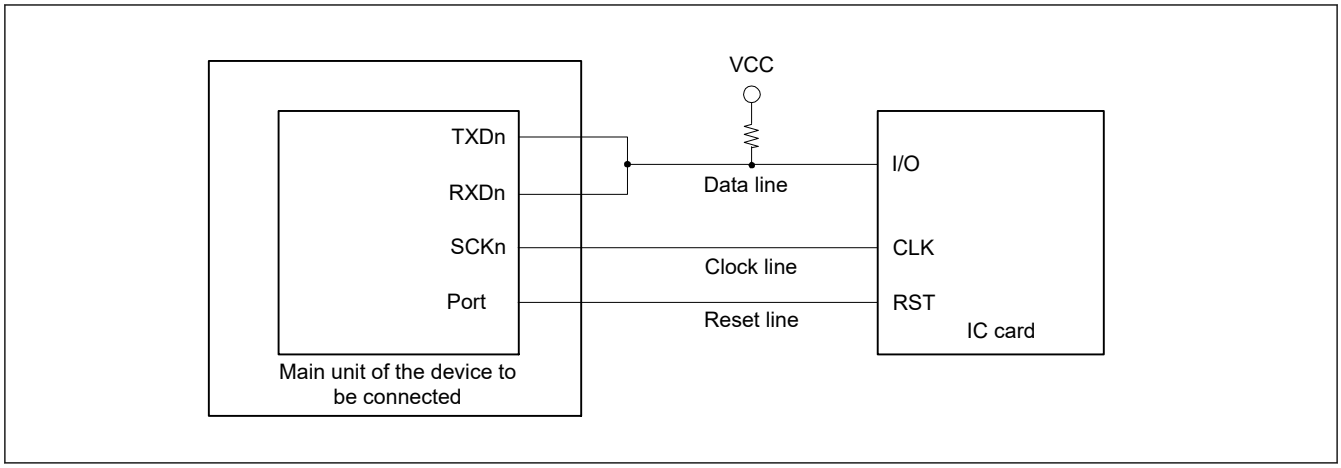


Figure 31.74 Example connection with a smart card (IC card)

31.7.2 Data Format (Except in Block Transfer Mode)

Figure 31.75 shows the data transfer formats in smart card interface mode:

- One frame consists of 8-bit data and a parity bit in Asynchronous mode.
- During transmission, at least 2 etus (elementary time unit – the time required for transferring 1 bit) is set as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low error signal is output for 1 etu after 10.5 etus elapse from the start bit.
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 etus.

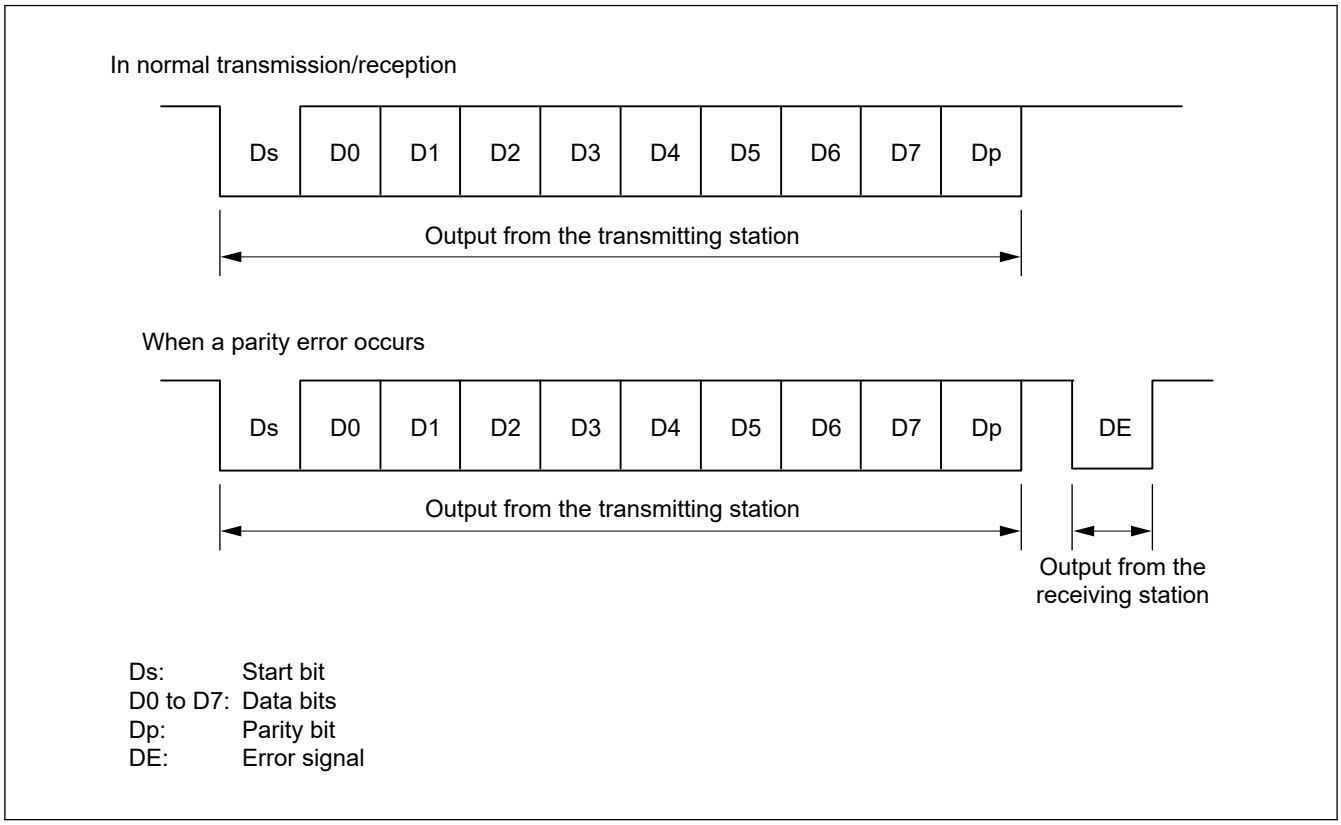


Figure 31.75 Data formats in smart card interface mode

For communications with IC cards of the direct convention type and inverse convention type, follow the procedures in this section.

(1) Direct Convention Type

For the direct convention type, logic levels 1 and 0 indicate the Z and A states, respectively, and data is transferred with LSB-first for the start character, as shown in Figure 31.76. Therefore, data in the start character in the figure is 0x3B.

When using the direct convention type, write 1 to the CCR3.LSBF and write 0 to the CCR3.SINV. Write 0 to the CCR1.PM bit to use even parity, which is prescribed by the smart card standard.

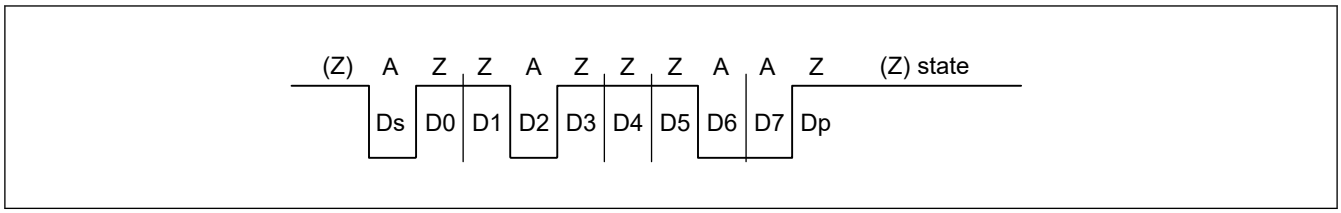


Figure 31.76 Direct convention with LSBF in CCR3 = 1, SINV in CCR3 = 0, and PM in CCR1 = 0

(2) Inverse Convention Type

For the inverse convention type, logic levels 1 and 0 indicate the A and Z states, respectively, and data is transferred with MSB-first for the start character, as shown in Figure 31.77. Therefore, data in the start character in the figure is 0x3F.

When using the inverse convention type, write 0 to the CCR3.LSBF and write 1 to the CCR3.SINV. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to the Z state. Because the SINV bit of the MCU only inverts data bits D7 to D0, write 1 to the PM bit in CCR1 to invert the parity bit for both transmission and reception.

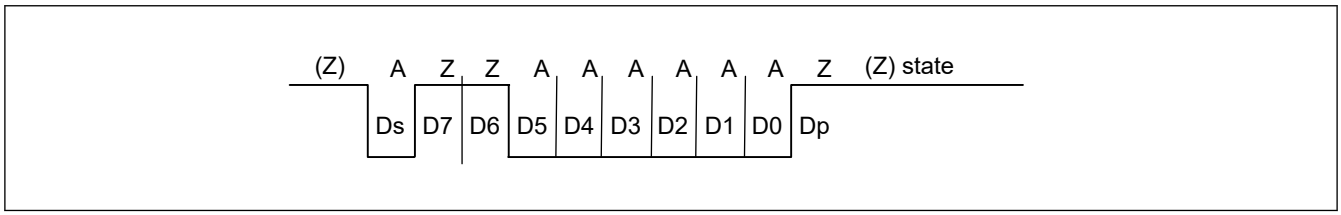


Figure 31.77 Inverse convention with LSBF in CCR3 = 0, SINV in CCR3 = 1, and PM in CCR1 = 1

31.7.3 Block Transfer Mode

Block transfer mode differs from normal smart card interface mode as follows:

- Even if a parity error is detected during reception, no error signal is output. Because the PER flag in CSR is set by error detection, clear the PER flag before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is set as a guard time from the end of the parity bit until the start of the next frame
- Because the same data is not retransmitted, the TEND flag in CSR is set to 11.5 etus after transmission starts
- In block transfer mode, the ERS flag in CSR indicates the error signal status as in normal smart card interface mode, but the flag is read as 0 because no error signal is transferred

31.7.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the on-chip baud rate generator can be used as a transfer clock in smart card interface mode.

In this mode, the SCI can operate on a base clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate set up in the CCR2.BCP[2:0] bits. The frequency is always 16 times the bit rate in normal Asynchronous mode.

For data reception, the falling edge of the start bit is sampled with the base clock to perform internal synchronization.

Receive data is sampled on the 16th, 32nd, 186th, 128th, 46th, 64th, 93rd, and 256th rising edges of the base clock so that it can be latched at the middle of each bit as shown in Figure 31.78. The reception margin is determined by the following formula:

$$M = \left|\left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F)\right| \times 100\ [\%]$$

M: Reception margin (%)

N: Ratio of bit rate to clock (N = 32, 64, 372, 256)

D: Duty cycle of clock (D = 0 to 1.0)

L: Frame length (L = 10)

F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the specified formula, the reception margin is determined using the following formula:

$$M = \{0.5 - 1/(2 \times 372)\} \times 100\ [\%] = 49.866\ \%$$

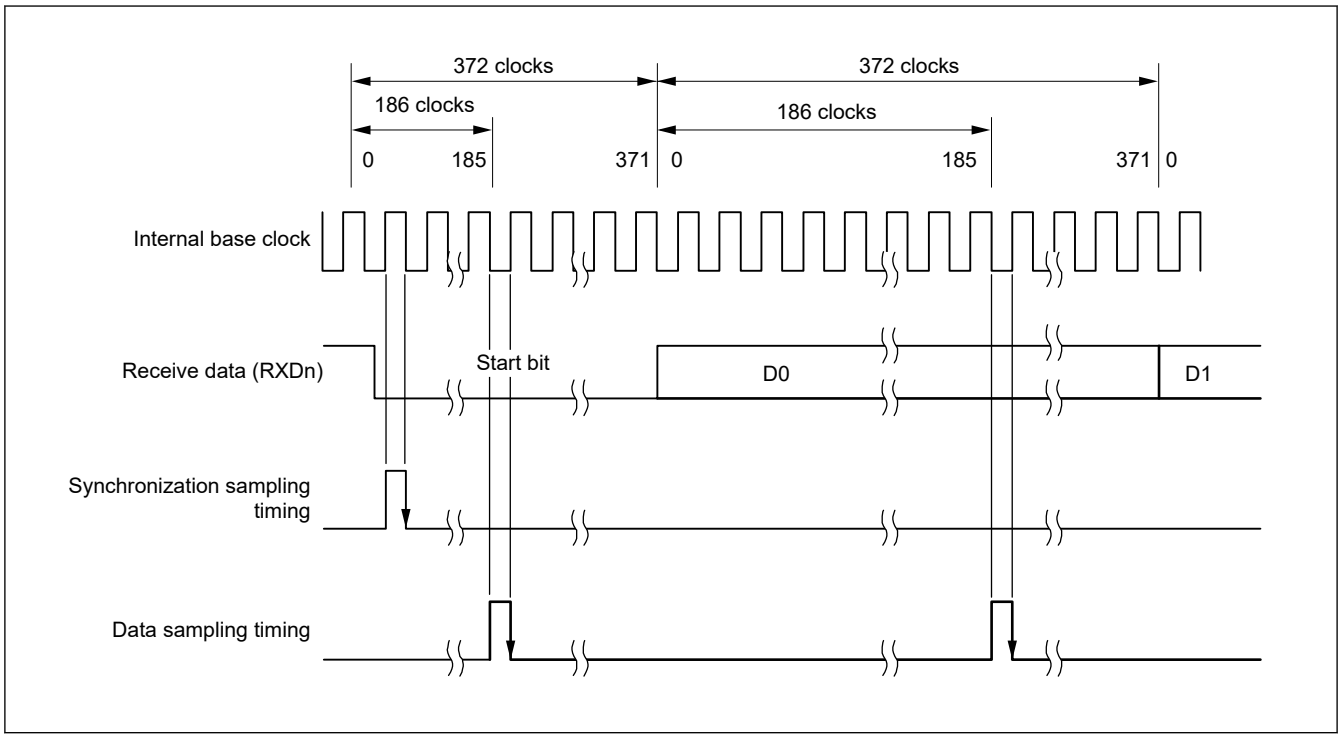


Figure 31.78 Receive data sampling timing in smart card interface mode when the clock frequency is 372 times the bit rate

31.7.5 SCI Initialization (Smart Card Interface Mode)

Before transmitting and receiving data, write the initial value 0x00 in the CCR0 register and initialize the SCI following the example flow shown in Table 31.40.

Always set the initial value in the TIE, RIE, TE, RE, TEIE bits in the CCR0 register before switching from transmission to reception mode or from reception to transmission mode. When CCR0.RE is set to 0, the RDR register is not initialized.

In transmission mode, set 1 to the CCR0.TE bit and CCR0.TIE bit simultaneously, then the SCIn_TXI interrupt request is generated.

To change from reception mode to transmission mode, first check that reception has completed, then initialize the SCI. At the end of initialization, set CCR0.TE = 1 and CCR0.RE = 0. Reception completion can be verified by reading the SCIn_RXI request, ORER, or PER flag in CSR.

To change transmission mode to reception mode, first check that transmission has completed, then initialize the SCI. At the end of initialization, set CCR0.TE = 0 and CCR0.RE = 1. Transmission completion can be verified by reading the TEND flag in CSR.

Table 31.40 Example flow of SCI initialization in smart card interface mode

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set CCR0.TEIE, TIE, RIE, TE, RE to 0. If you have not changed from the initial settings, you can skip this step.
3	Set CCR3	Set communication mode (MOD[2:0] = 001b), BLK, GM, and SINV. Leave other bits at their initial values.
4	Set CCR2	Set clock-select and bit-rate. Set BRME to 0.
5	Set CCR1	Set up the Noise filter function, communication terminal status. Set NFEN, PE, CTSE to 0 and set PE to 1.
6	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
7	Set CCR3	Set CKE[1:0]. When the CKE[0] bit is set to “1” due on GM setting value, the clock is output from the SCKn pin.
8	Set CFCLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
9	Set CCR0	Set the TE or RE to 1. And set the TIE and RIE. Do not simultaneously set the TE and RE bits to 1 if self-diagnosis in not used.
10	Initialization completed	—

Figure 31.79 is a timing chart when data transmission is performed by making transition to the Smart Card Interface mode according to the above flow chart. The figure shows the case when CCR3.GM bit is 0. As shown in the figure, when the pin function is set to the SCKn pin, the SCKn pin is high impedance because the CCR3.CKE [0] bit is 0. When the TXDn pin is set, the TXDn pin is high impedance because the CCR0.TE bit is 0. Start clock output to the SCKn pin with the clock output setting CCR3.CKE [0] to 1, start data transmission by writing transmit data after setting CCR0.TE to 1.

In the smart card interface mode, even if not communicating at CCR0.TE = 0 and CCR0.RE = 0, the clock is continuously output if the clock output setting is used.

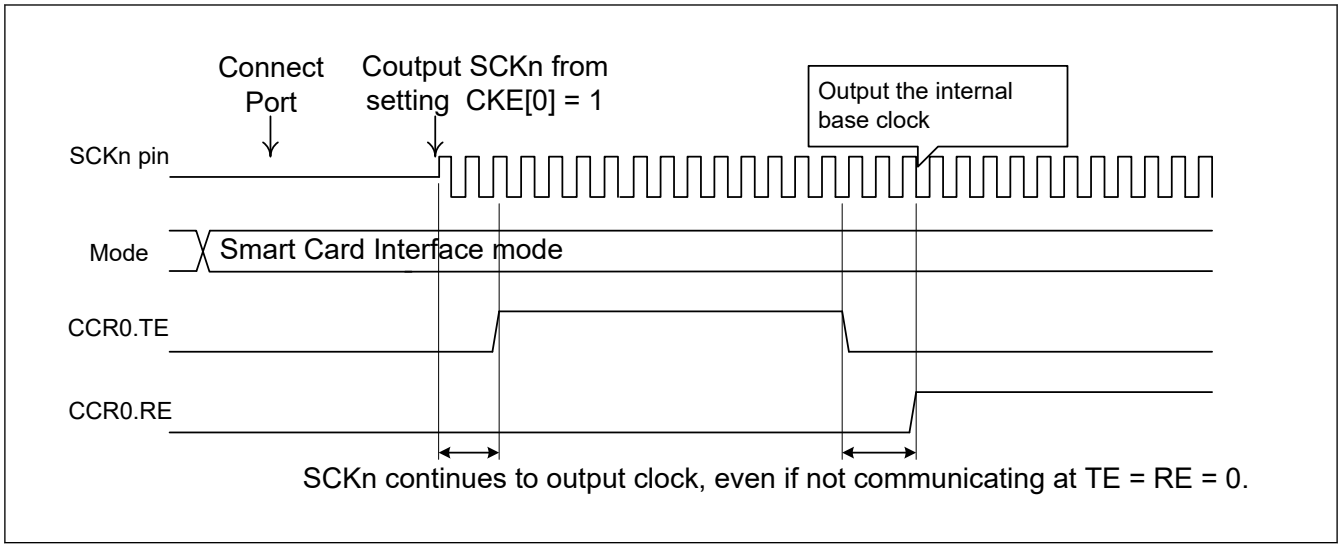


Figure 31.79 Example of Timing chart of data transmission in Smart Card Interface Mode

31.7.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode) is different from that in non-smart card interface mode, in that an error signal is sampled and data can be re-transmitted in smart card mode. Figure 31.80 shows the data re-transfer operation during transmission.

1. When an error signal from the receiver end is sampled after 1-frame data is transmitted, the CSR.ERS flag is set to 1. If the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated. Clear the ERS flag to 0 before the next parity bit is sampled.

2. For a frame in which an error signal is received, the CSR.TEND flag is not set. Data is re-transferred from TDR to TSR, allowing automatic data retransmission.
3. If no error signal is returned from the receiver, the ERS flag is not set to 1.
4. In this case, the SCI determines that transmission of 1-frame data, including the re-transfer, is complete, and the TEND flag is set. If the CCR0.TIE bit is 1, an SCIn_TXI interrupt request is generated. Write transmit data to the TDR to start transmission of the next data.

Figure 31.82 shows an example flow of serial transmission. All the processing steps are automatically performed using an SCIn_TXI interrupt request to activate the DTC or DMAC.

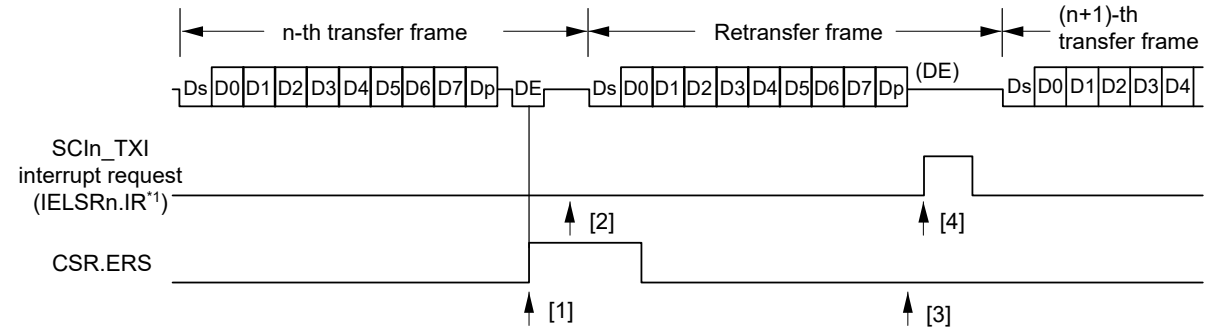
When the CSR.TEND flag is set to 1 in transmission and when the CCR0.TIE bit is 1, an SCIn_TXI interrupt request is generated.

The DTC or DMAC is activated by an SCIn_TXI interrupt request if the SCIn_TXI interrupt request is previously specified as a source of DTC or DMAC activation, allowing the transfer of transmit data. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During this retransmission, the TEND flag is kept at 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission when an error occurs. Because the ERS flag is not automatically cleared, set the RIE bit to 1 before enabling an SCIn_ERI interrupt request to be generated if an error occurs, and clear the ERS flag to 0.

When transmitting or receiving data using the DTC or DMAC, always enable the DTC or DMAC before making the SCI settings.

For DTC or DMAC settings, see section 17, Data Transfer Controller (DTC), section 16, DMA Controller (DMAC).



Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.80 Data re-transfer operation in smart card interface transmission mode

The CSR.TEND flag is set at different timings depending on the CCR3.GM bit setting. Figure 31.81 shows the TEND flag generation timing.

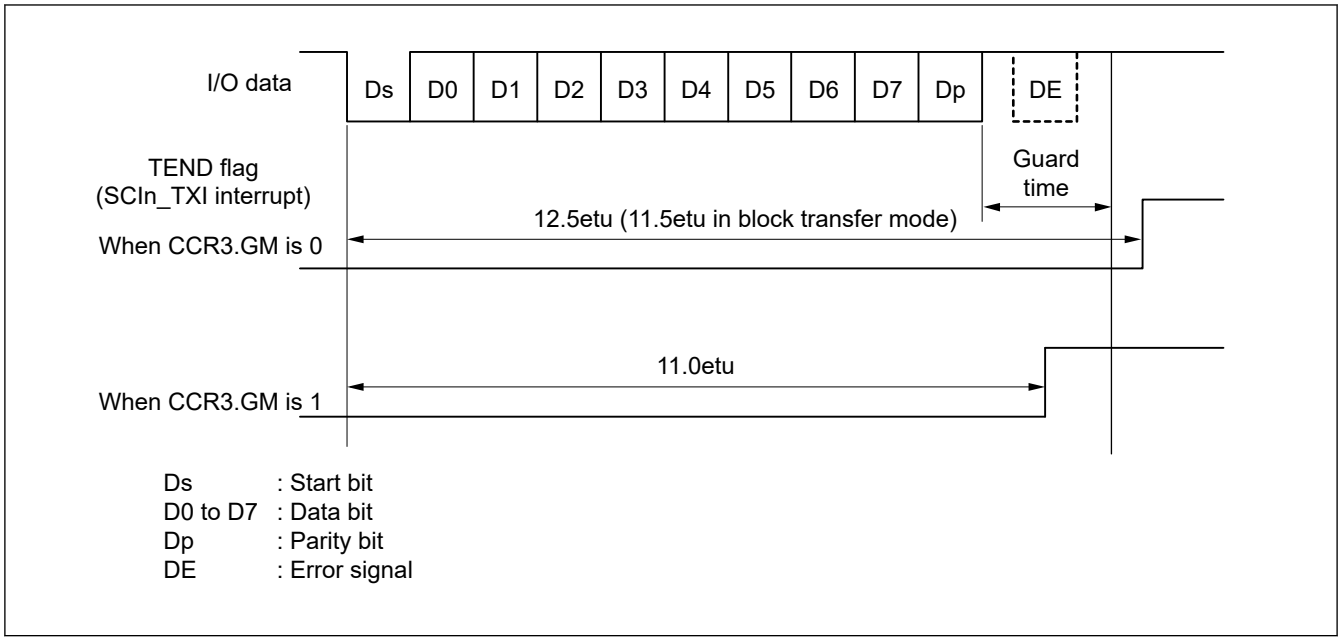


Figure 31.81 CSR.TEND flag generation timing during transmission

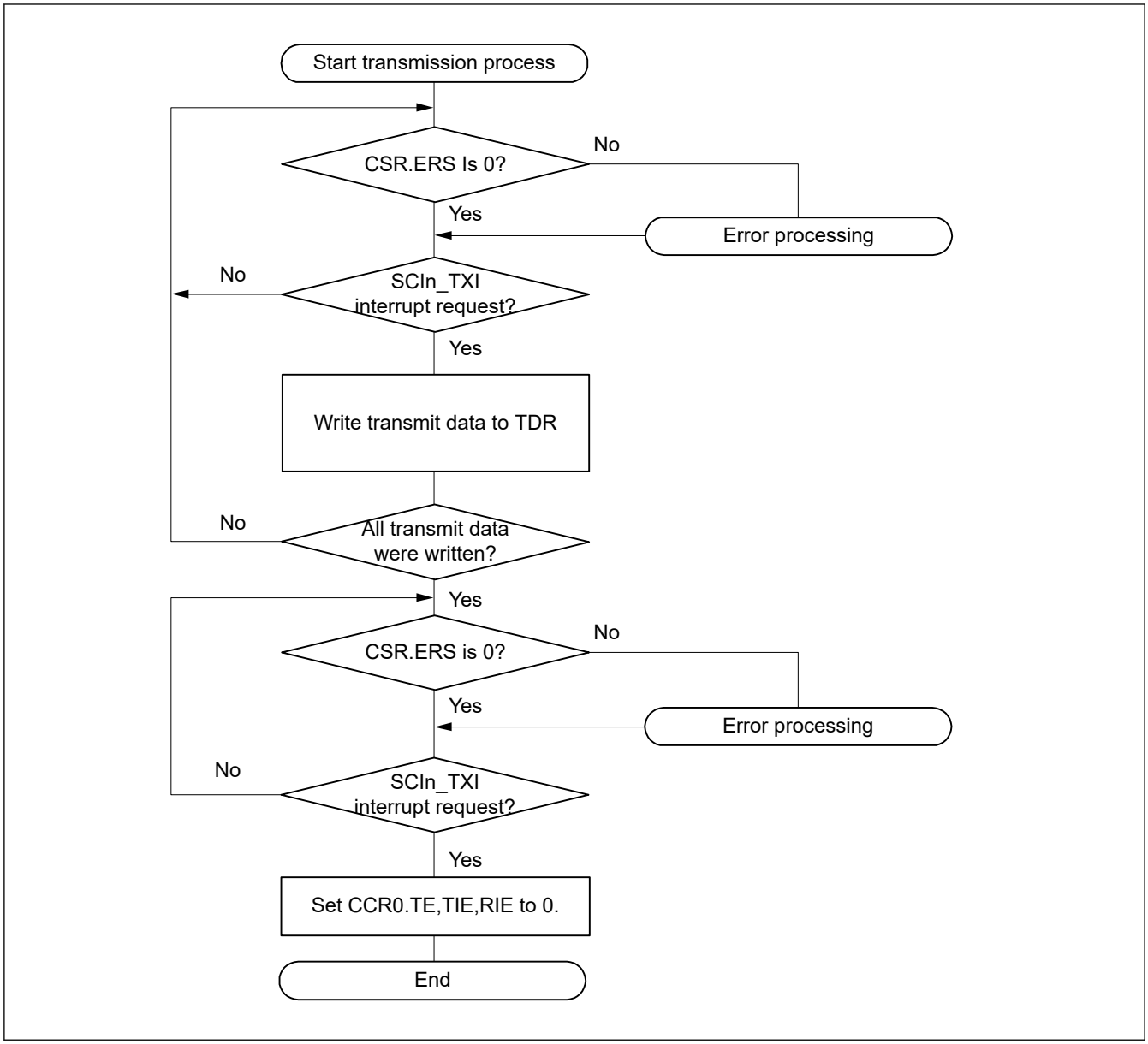


Figure 31.82 Example flow of smart card interface transmission

31.7.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in non-smart card interface mode. Figure 31.83 shows the data re-transfer operation in reception mode.

1. If a parity error is detected in the receive data, the CSR.PER flag is set to 1. When the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated. Clear the PER flag to 0 before the next parity bit is sampled.
2. For a frame in which a parity error is detected, no SCIn_RXI interrupt is generated.
3. When no parity error is detected, the CSR.PER flag is not set to 1.
4. In this case, data is determined to be received successfully. When the CCR0.RIE bit is 1, an SCIn_RXI interrupt request is generated.

Figure 31.84 shows an example flow of serial data reception. All the processing steps are automatically performed using an SCIn_RXI interrupt request to activate the DTC or DMAC.

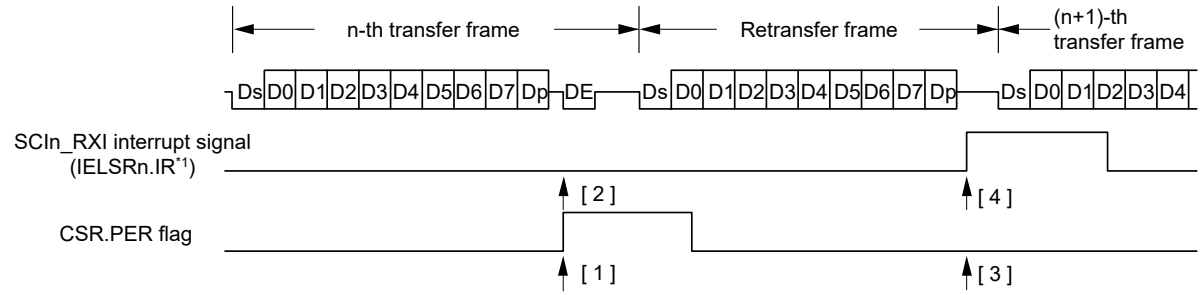
In reception, setting the RIE bit to 1 allows an SCIn_RXI interrupt request to be generated. The DTC or DMAC is activated by an SCIn_RXI interrupt request if the SCIn_RXI interrupt request is previously specified as a source of DTC or DMAC activation, allowing the transfer of receive data.

If an error occurs during reception and either the ORER or PER flag in CSR is set to 1, a receive error interrupt (SCIn_ERI) request is generated. Clear the error flag after the error occurrence. If an error occurs, the DTC or DMAC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC or DMAC is transferred.

If a parity error occurs and the PER flag is set to 1 during reception, the receive data is transferred to RDR, allowing the data to be read.

When a reception is forced to terminate by setting CCR0.RE to 0 during operation, read the RDR register because the received data that is not yet read might be left in the RDR.

Note: For operations in block transfer mode, see section 31.3.9. Serial Data Reception in Asynchronous Mode.



Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.83 Data re-transfer operation in smart card interface reception mode

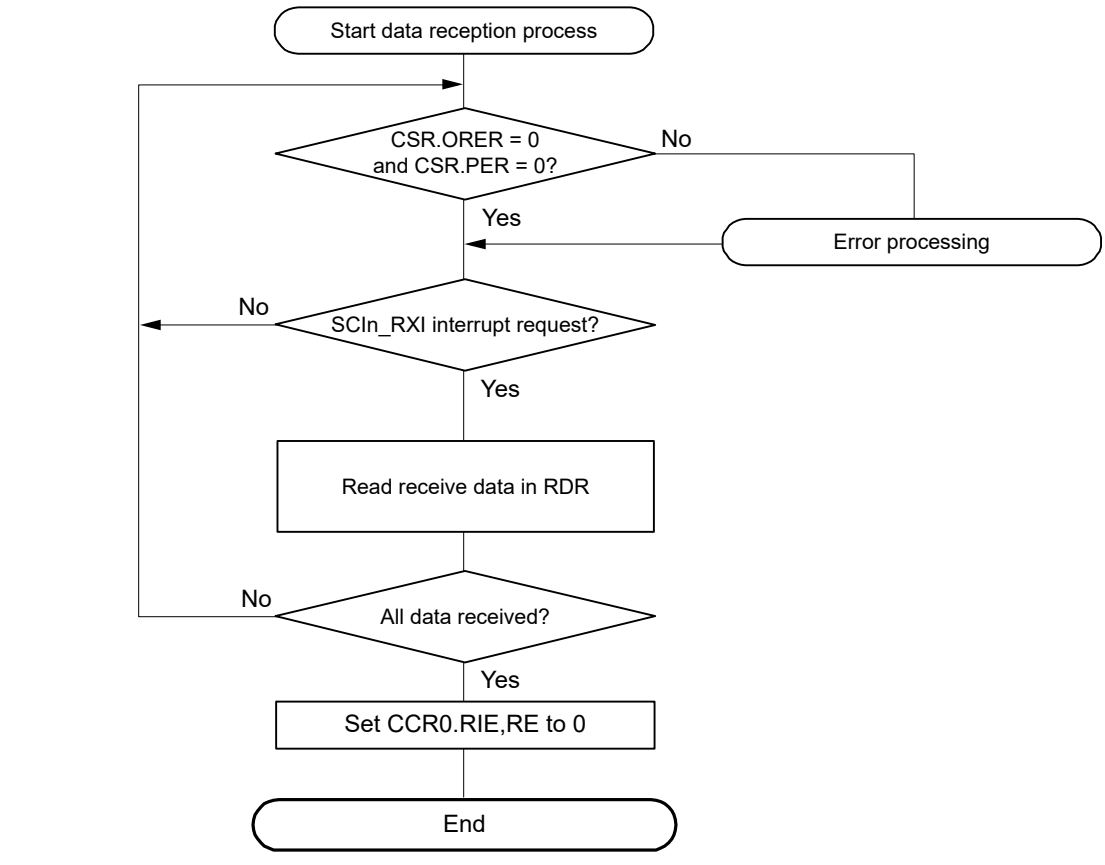


Figure 31.84 Example flow of smart card interface reception

31.7.8 Clock Output Control

When the GM bit in CCR3 is set to 1, the clock output can be controlled by the CKE[1:0] bits in CCR3. For details on the CKE[1:0] bits, see section 31.2.8. CCR3 : Common Control Register 3. When setting the clock output, the base clock described in section 31.7.4. Receive Data Sampling Timing and Reception Margin the bit rate is set by CCR2.CKS, CCR2.BCP[2:0] and BRR[7:0].

Figure 31.85 shows an example timing for the clock output control when the CKE[1] bit in CCR3 is set to 0 and the CKE[0] bit in CCR3 is controlled.

When the GM bit in CCR3 is 0, output control by the CKE[0] bit in CCR3 is immediately reflected on the SCKn pin, so there is a possibility that pulses with an unintended width may be output from the SCKn pin.

When the GM bit in CCR3 is 1, the output pulse control by the CCR3.CKE [0] controls the pulse width set to be based on the state of the base clock.

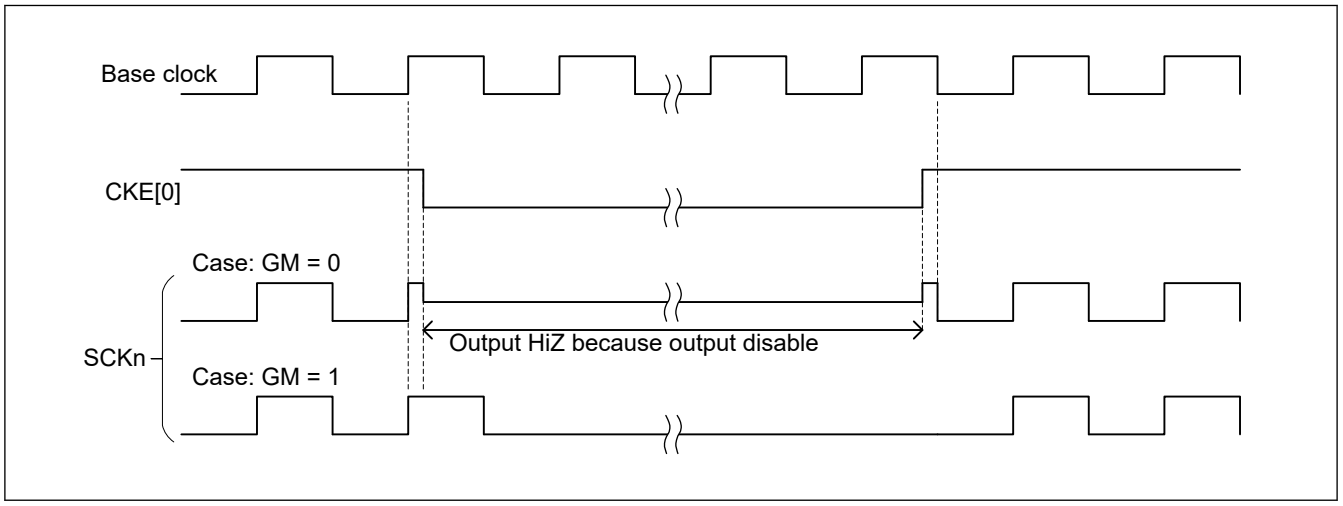


Figure 31.85 Clock Output timing

31.8 Operation in Simple IIC Mode

Simple IIC mode format is composed of 8 data bits and an acknowledge bit. By continuing into a slave-address frame after a start condition or restart condition, a master device can specify a slave device as the partner for communications. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied. The 8 data bits in all frames are transmitted in order from the MSB.

The I^2C bus format and timing of the I^2C bus are shown in Figure 31.86 and Figure 31.87.

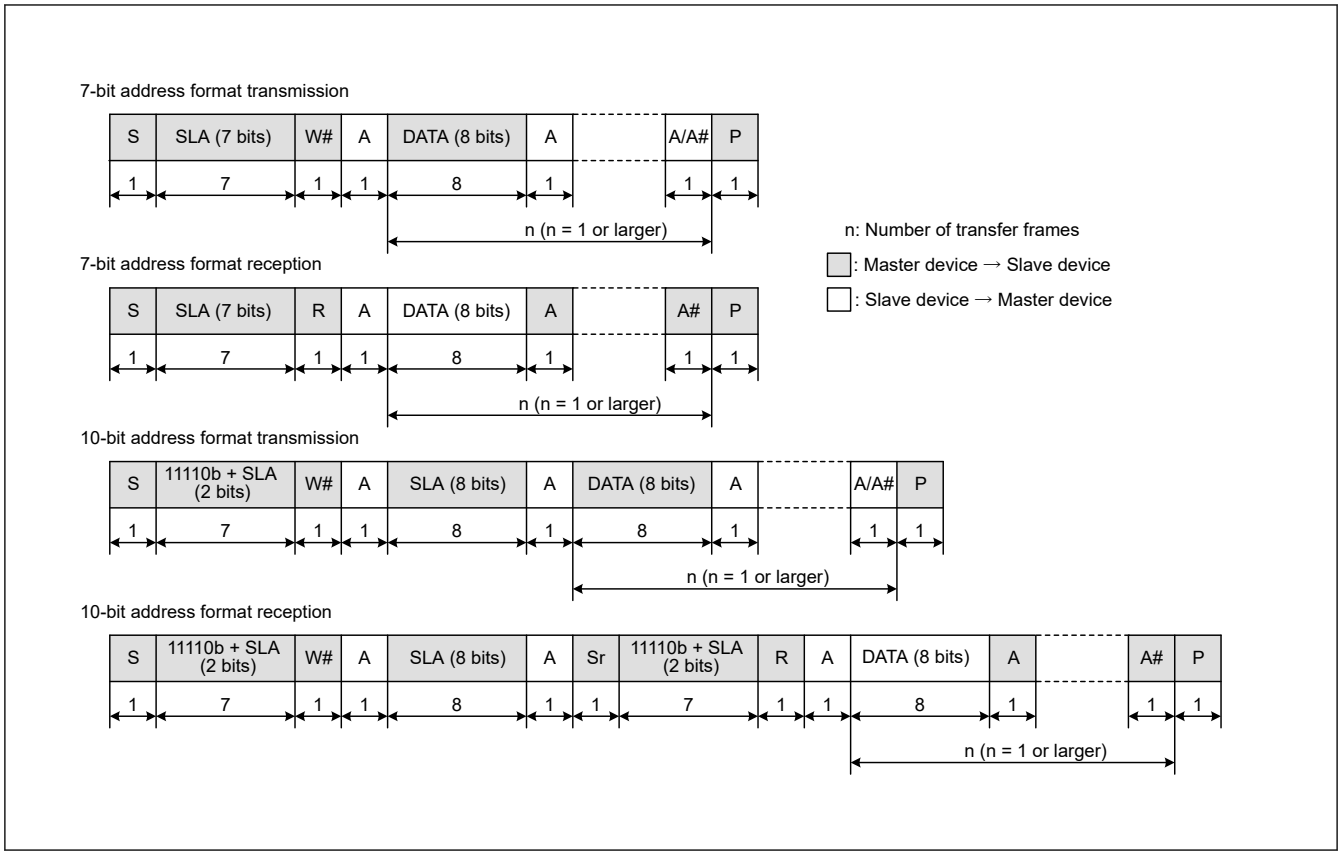


Figure 31.86 **I²C bus format**

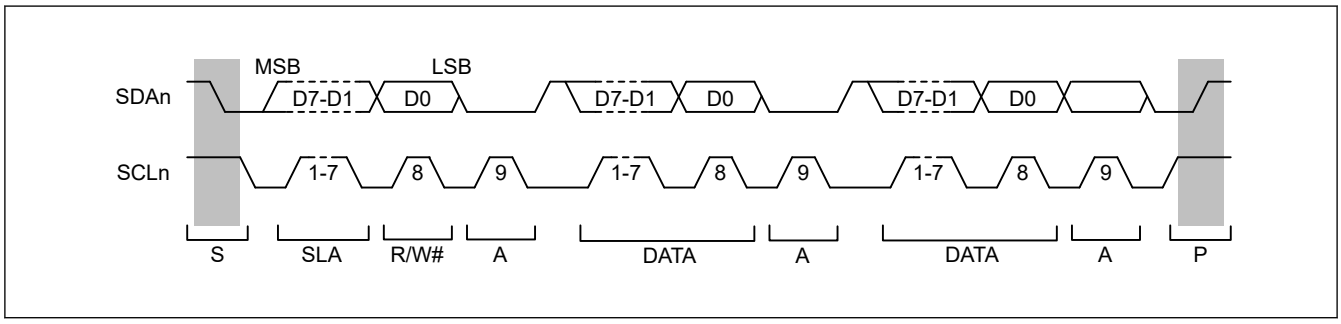


Figure 31.87 **I²C bus timing when SLA is 7 bits**

- S: Indicates a start condition, when the master device changes the level on the SDAn line from high to low while the SCLn line is high
- SLA: Indicates a slave address, by which the master device selects a slave device
- R/W#: Indicates the direction of transfer (reception or transmission). The value 1 indicates transfer from the slave device to the master device and 0 indicates transfer from the master device to the slave device.
- A/A#: Indicates an acknowledge bit. This is returned by the slave device for master transmission and by the master device for master reception. Return low indicates ACK and return high indicates NACK.
- Sr: Indicates a restart condition, when the master device changes the level on the SDAn line from high to low while the SCLn line is high and after the setup time elapses
- DATA: Indicates the data being received or transmitted
- P: Indicates a stop condition, when the master device changes the level on the SDAn line from low to high while the SCLn line is high

31.8.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to the ICR.IICSTAREQ bit causes the generation of a start condition. The generation of a start condition proceeds through the following operations:

- The level on the SDAn line falls (from the high level to the low level) and the SCLn line is kept in the released state
- The hold time for the start condition is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The level on the SCLn line falls (from the high level to the low level), the IICSTAREQ bit in ICR is set to 0, and a start-condition generated interrupt is output

Writing 1 to the IICRSTAREQ bit in ICR causes the generation of a restart condition. The generation of a restart condition proceeds through the following operations:

- The SDAn line is released and the SCLn line is kept at the low level
- The period at low level for the SCLn line is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The SCLn line is released (transition from the low to the high level)
- When a high level is detected on the SCLn line, the setup time for the restart condition is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The level on the SDAn line falls (from the high level to the low level)
- The hold time for the restart condition is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The level on the SCLn line falls (from the high level to the low level), the ICR.IICRSTAREQ bit is set to 0, and a restart-condition generated interrupt is output

Writing 1 to the ICR.IICSTPREQ bit causes the generation of a stop condition. The generation of a stop condition proceeds through the following operations:

- The level on the SDAn line falls (from the high level to the low level) and the SCLn line is kept at the low level
- The period at low level for the SCLn line is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The SCLn line is released (transition from the low to the high level)
- When a high level is detected on the SCLn line, the setup time for the stop condition is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The SDAn line is released (transition from the low to the high level), the ICR.IICSTPREQ bit is set to 0, and a stop-condition generated interrupt is output

Figure 31.88 shows the timing of operations in the generation of start, restart, and stop conditions.

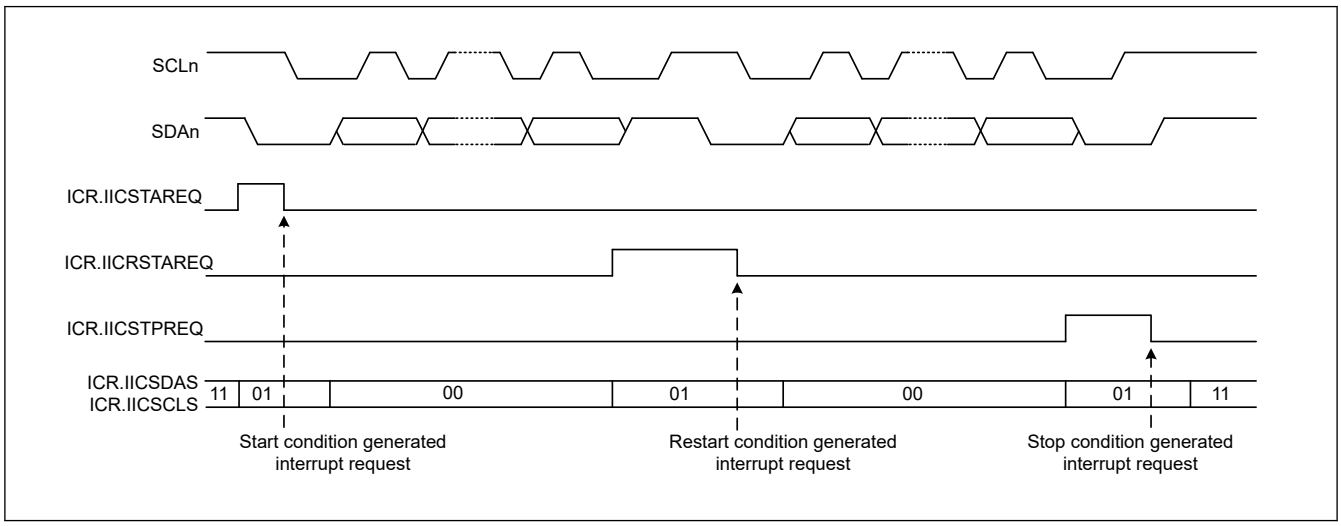


Figure 31.88 Timing of operations in generation of start, restart, and stop conditions

31.8.2 Clock Synchronization

The SCLn line can be driven low if a wait is inserted by a slave device at the other side of the transfer. Setting the ICR.IICCSC bit to 1 applies control to obtain synchronization when a difference arises between the levels of the internal SCLn clock signal and the level being input on the SCLn pin.

When the ICR.IICCSC bit is set to 1, the level of the internal SCLn clock signal changes from low to high. Counting to determine the period at a high level stops while the low level is being input on the SCLn pin. Counting to determine the period at a high level starts after the transition of the input on the SCLn pin to the high level.

The interval from this time until counting to determine the period at high level starts on the transition of the SCLn pin to the high level, is the total time which contains the SCLn input delay, delay for noise filtering of the input on the SCLn pin (2 or 3 cycles of sampling clock for the noise filter), and delay for internal processing (1 or 2 cycles of PCLK). The period at high level of the internal SCLn clock is extended even when other devices do not place the low level on the SCLn line.

If the ICR.IICCSC bit is 1, synchronization is obtained for the transmission and reception of data by taking the logical AND of the input on the SCLn pin and the internal SCLn clock. If the ICR.IICCSC bit is 0, synchronization with the internal SCLn clock is obtained for the transmission and reception of data.

If a slave device inserts a wait period into the interval until the transition of the internal SCLn clock signal from the low to the high level after a request for the generation of a start, restart, or stop condition is issued, the time until generation is prolonged by that period.

If a slave device inserts a wait period after the transition of the internal SCLn clock signal from the low to the high level, although the generation-completed interrupt is issued without stopping the waiting period, generation of the condition itself is not guaranteed.

Figure 31.89 shows an example operation for synchronizing the clocks.

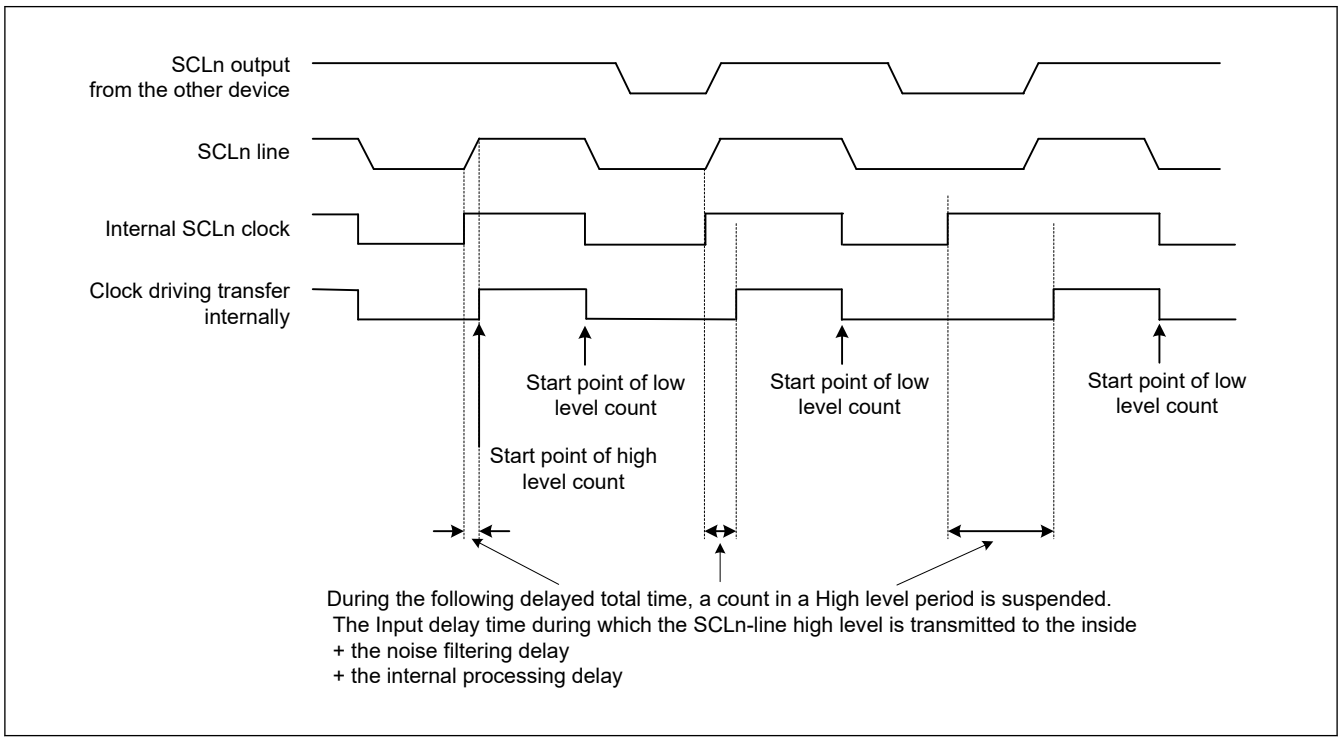


Figure 31.89 Example operations for clock synchronization

31.8.3 SDAn Output Delay

The ICR.IICDL[4:0] bits can be used to set a delay for output on the SDAn pin relative to falling edges of output on the SCLn pin. Delay settings from 0 to 31 are selectable, representing periods of the corresponding numbers of cycles of the clock signal from the on-chip baud rate generator (derived by frequency-dividing the base clock, TCLK, by the divisor selected in the CCR2.CKS[1:0] bits). A delay for output on the SDAn pin applies to the start condition/restart condition/stop condition signal, 8-bit transmit data, and acknowledge bit.

If the SDAn output delay is shorter than the time for the level on the SCLn pin to fall, the change of the output on the SDAn pin starts while the output level on the SCLn pin is falling, creating a possibility of erroneous operation for slave devices. Ensure that settings for the delay of output on the SDAn pin specify times greater than the time output on the SCLn pin takes to fall (300 ns for IIC in normal mode and fast mode).

Figure 31.90 shows the timing of delays in SDAn output.

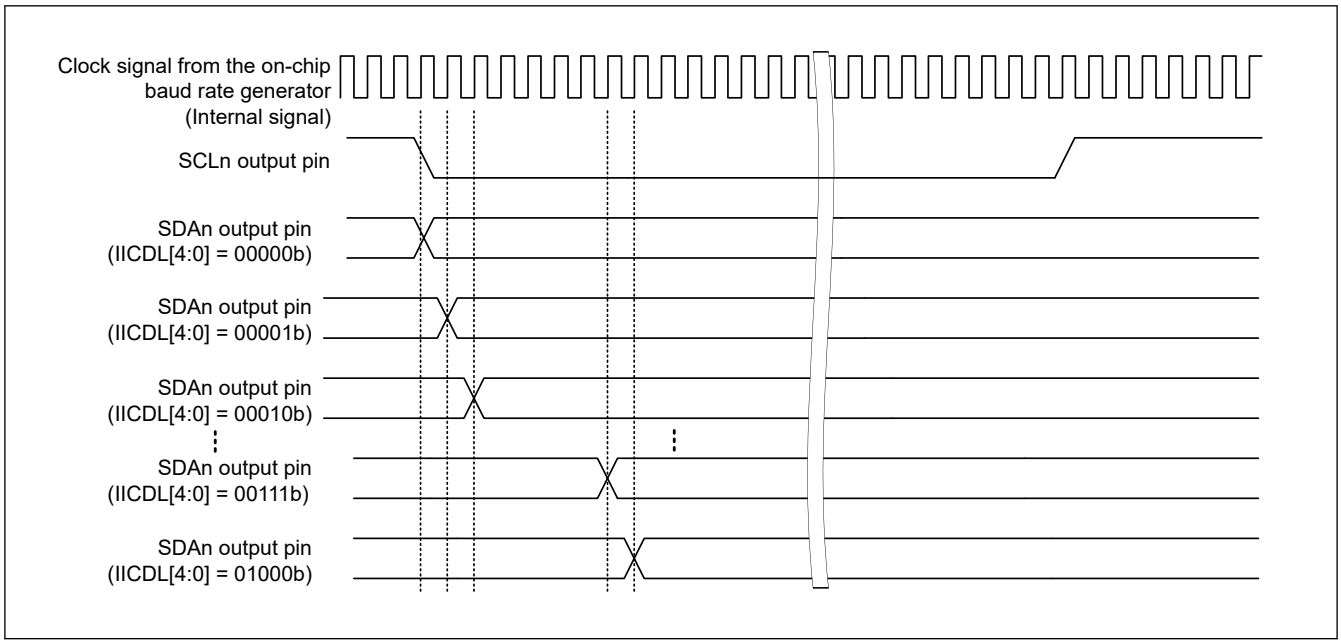


Figure 31.90 Timing of delays in SDAn output

31.8.4 SCI Initialization in Simple IIC Mode

Before transferring data, write the initial value 0x00 to CCR0 and initialize the interface following the example shown in Table 31.41.

Before making any changes to the operating mode or transfer format, be sure to set CCR0 to its initial value. In simple IIC mode, the open-drain setting for the communication ports should be made on the port side.

Table 31.41 Example flow of SCI initialization in simple IIC mode

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set CCR0.TEIE, TIE, RIE, TE, RE to 0. If you have not changed from the initial settings, you can skip this step.
3	Set ICR	Set the IICSDAS[1:0] and IICSCLS[1:0] to 11b. Set the IICDL[4:0] and IICINTM as required. Set the IICACKT and the IICCSC bits to 1.
4	Set CCR3	Set the transmission / reception format as the communication mode (MOD [2:0] = 100b) and CKE [1:0] = 00b.
5	Set CCR2	Set the bit rate modulation function*1, the clock selection, and the bit rate.
6	Set CCR1	Set noise filter, communication pin status, parity check, and CTSn / RTSn function.
7	Set the I/O port functions	Set I/O port settings that allow use (on NMOS open-drain output pins and Hi-Z) of the SCLn and SDAn pin functions.
8	Set CFCLR, ICFCLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC ICFCLR.IICSTIFC
9	Set CCR0 (TE, RE, TIE, RIE)	Set the TE and RE bits to 1. To enable interrupts, set the TE, TIE, RE and RIE bits to 1 with one instruction at the same time (for transmission and when the IICINTM bit is 1, clear the RIE bit). Setting the TE and RE bits to 1 makes the SCLn and SDAn pins functions available.
10	Initialization completed	—

Note: Set the CCR0.TE and RE bits to 0 or 1 at the same time.

Note 1. If you do not use the bit rate modulation function, you do not need to set it.

31.8.5 Operation in Master Transmission in Simple IIC Mode

Figure 31.91 and Figure 31.92 show examples of master transmission and Figure 31.93 shows an example flow of data transmission.

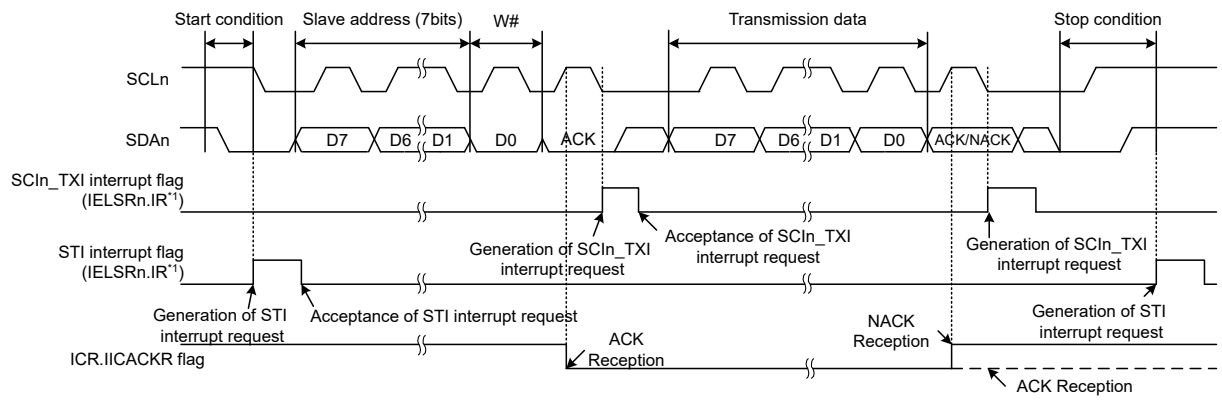
Figure 31.91 shows the operation example when ICR.IICINTM bit is 1 (use reception and transmission interrupts). In this case, you can start DMAC or DTC by SCIn_TXI interrupt. However, if use DMAC or DTC, ACK/NACK cannot be confirmed. If you want to confirm ACK/NACK, prepare the transmit data by CPU.

In simple IIC mode, SCIn_TXI interrupt is generated when communication of one frame is completed. And it is not used SCIn_RXI interrupt in master transmission, so the CCR0.RIE set to 0.

See Table 31.46 for more information on the STI interrupt.

Figure 31.93 shows a flow chart in the case of ICR.IICINTM is 1 and address transmission by CPU and data transmission by DTC or DMAC. When 10-bit slave addresses are in use, steps [3] and [4] are repeated twice.

In simple IIC mode, the transmit data empty interrupt (SCIn_TXI) is generated when communication of one frame is complete, unlike the SCIn_TXI interrupt request generation timing during clock synchronous transmission.



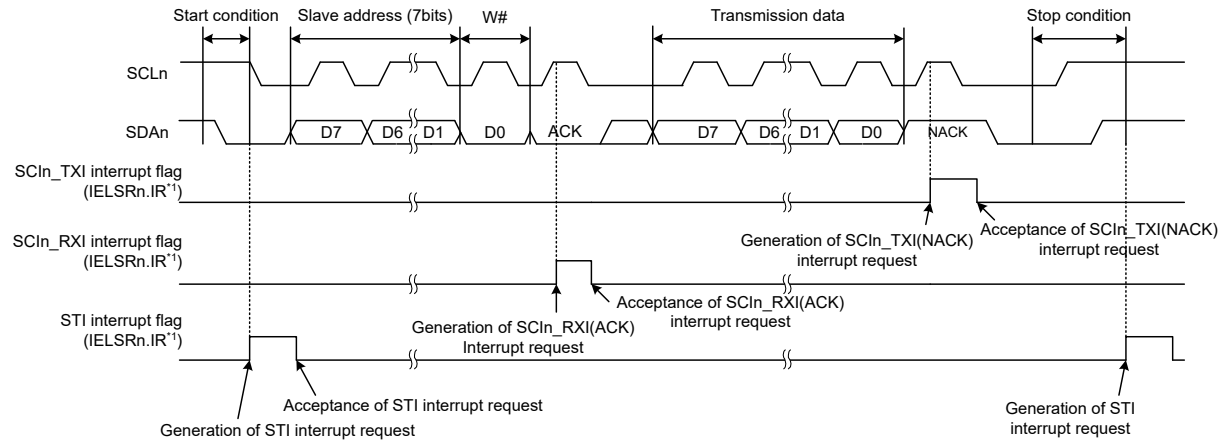
Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.91 Example 1 of operations for master transmission in simple IIC mode with 7-bit slave addresses, transmission interrupts, and reception interrupts (ICR.IICINTM = 1)

When the ICR.IICINTM bit is set to 0 (use ACK/NACK interrupts) during master transmission, the DTC or DMAC is activated by the ACK interrupt as the trigger and required number of data bytes are transmitted. When the NACK is received, error processing such as transmission stop and retransmission is performed using the NACK interrupt as the trigger.

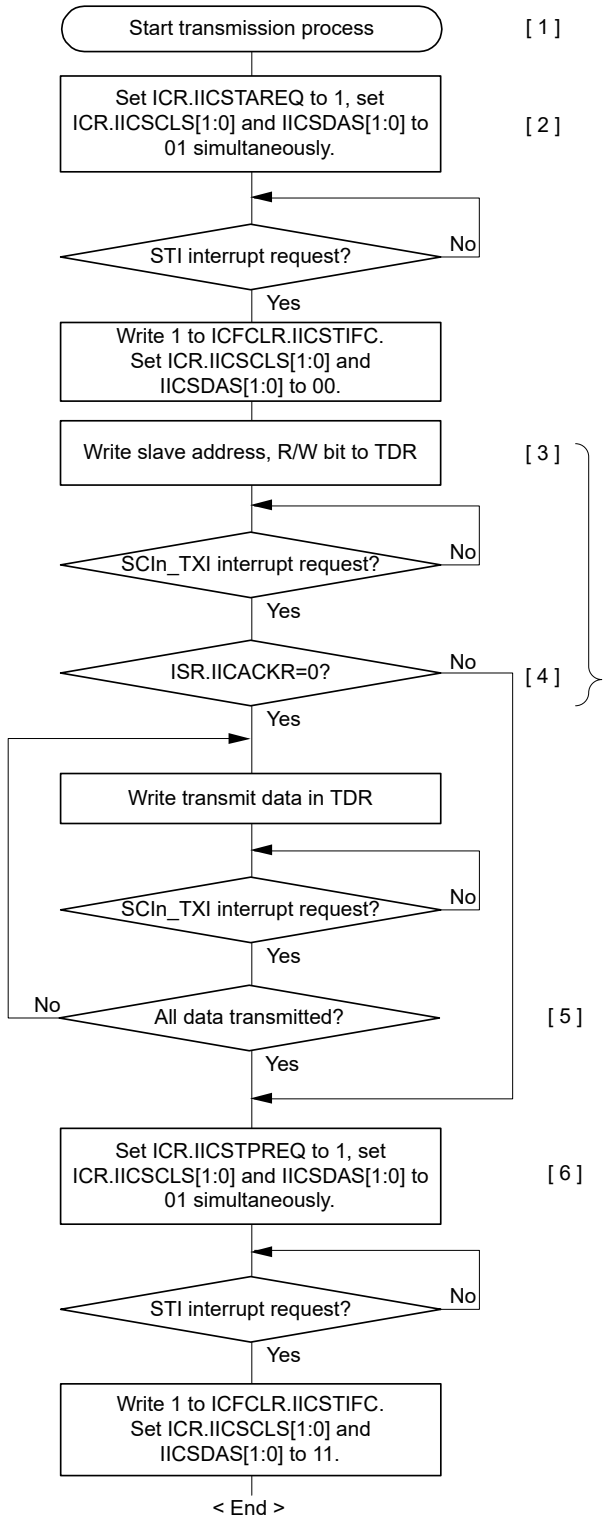
To restart communication for some reason after writing data in the TDR register, use the following procedure:

1. Set the TE and RE bits in the CCR0 register to 0 to stop communication.
2. Set ICR.IICSCLS[1:0] and ICR.IICSDAS[1:0] bits to 11b, release the I^2C bus, and clear the generation of a condition.
3. If the RDRF flag in the CSR register is set to 1, the RDR register is read by dummy and the RDRF bit is set to 0.
4. Set the TE and RE bits in the CCR0 register to 1 and start the next communication.



Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.92 Example 2 of operations for master transmission in simple IIC mode with 7-bit slave addresses, ACK interrupts, and NACK interrupts (ICR.IICINTM = 0)



[1] Start transmission process
For transmission, set the CCR0.RIE to 0.
(SCIn_RXI and SCIn_ERI interrupts requests are disabled)

[2] Generate a start condition.

[3] Writing to TDR:
Writing the slave address and value for the R/W bit to TDR.

[4] Confirm ACK response from the slave device:
Check the ISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

[5] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to TDR. Except for the first data to be transmitted, a SCIn_TXI interrupt request can activate the DMAC or DTC to handle writing of data to TDR.
But it cannot be confirmed status of ACK or NACK in this case.

[6] Generation of a stop condition.

If 10-bits slave address are in use, processing from [3] to [4] is repeated twice.

Note: In simple IIC mode, the SCIn_TXI interrupt request is generated when communication is completed.

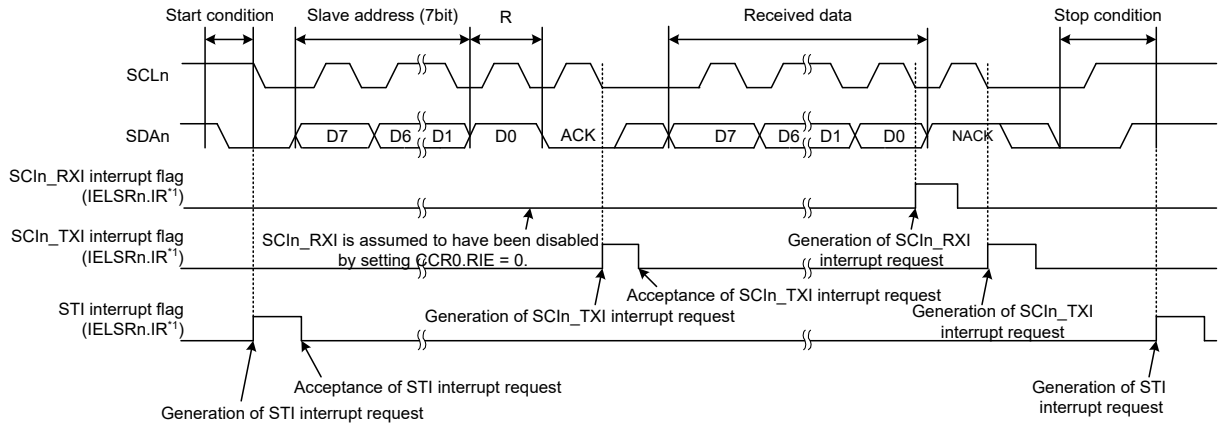
Figure 31.93 Example flow of master transmission in simple IIC mode with transmission interrupts and reception interrupts

31.8.6 Master Reception in Simple IIC Mode

Figure 31.94 shows an example operation in simple IIC mode master reception and Figure 31.96 shows an example flow of master reception.

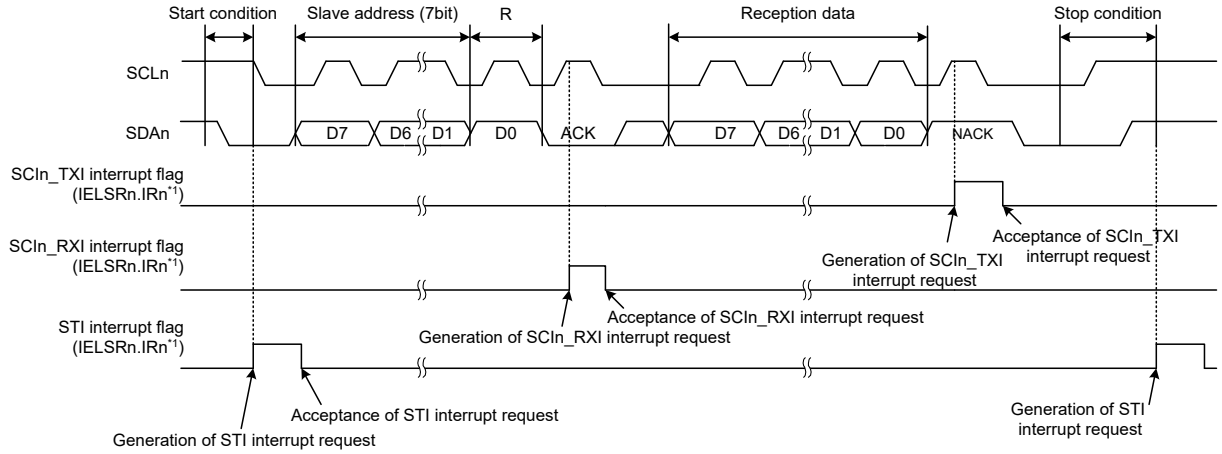
The value of the ICR.IICINTM bit is assumed to be 1 (use reception and transmission interrupts) and 0 (use ACK and NACK interrupts).

In simple IIC mode, the transmit data empty interrupt (SCIn_TXI) is generated when communication of one frame is complete, unlike the SCIn_TXI interrupt request generation timing during clock synchronous transmission.



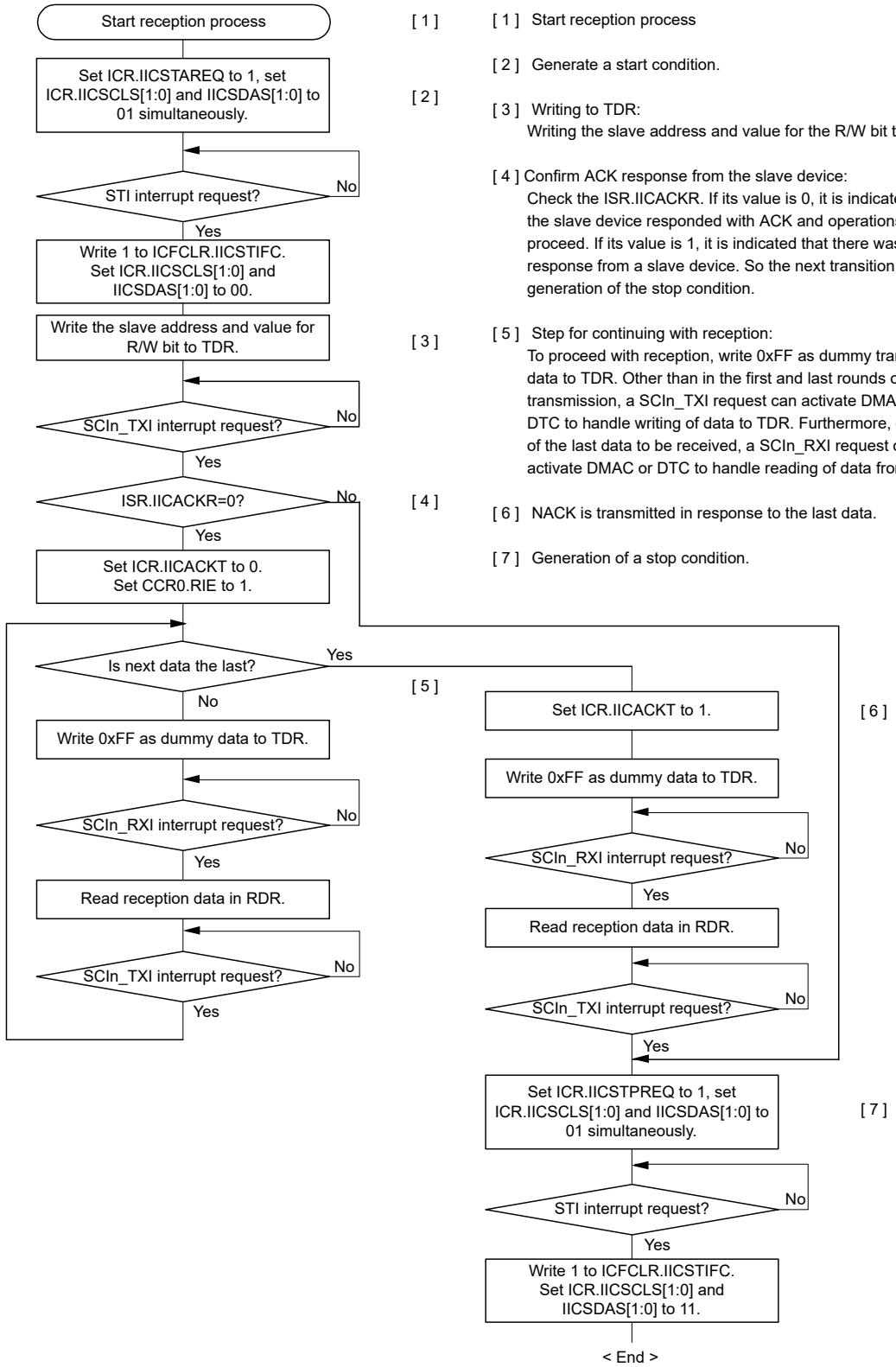
Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.94 Example operations for master reception in simple IIC mode with 7-bit slave addresses, transmission interrupts, and reception interrupts (ICR.IICINTM = 1)



Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.95 Example of Operations for Master Reception in Simple IIC Mode (7bit Slave address, ACK and NACK interrupt in use (ICR.IICINTM = 0))



[1] Start reception process

[2] Generate a start condition.

[3] Writing to TDR:
Writing the slave address and value for the R/W bit to TDR.

[4] Confirm ACK response from the slave device:
Check the ISR.IICACKR. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device. So the next transition is to generation of the stop condition.

[5] Step for continuing with reception:
To proceed with reception, write 0xFF as dummy transmit data to TDR. Other than in the first and last rounds of transmission, a SCIn_TXI request can activate DMAC or DTC to handle writing of data to TDR. Furthermore, except of the last data to be received, a SCIn_RXI request can activate DMAC or DTC to handle reading of data from RDR.

[6] NACK is transmitted in response to the last data.

[7] Generation of a stop condition.

Note: In simple IIC mode, the SCIn_TXI interrupt request is generated when communication is completed.

Figure 31.96 Example flow of master reception in simple IIC mode with transmission interrupts and reception interrupts (ICR.IICINTM = 1)

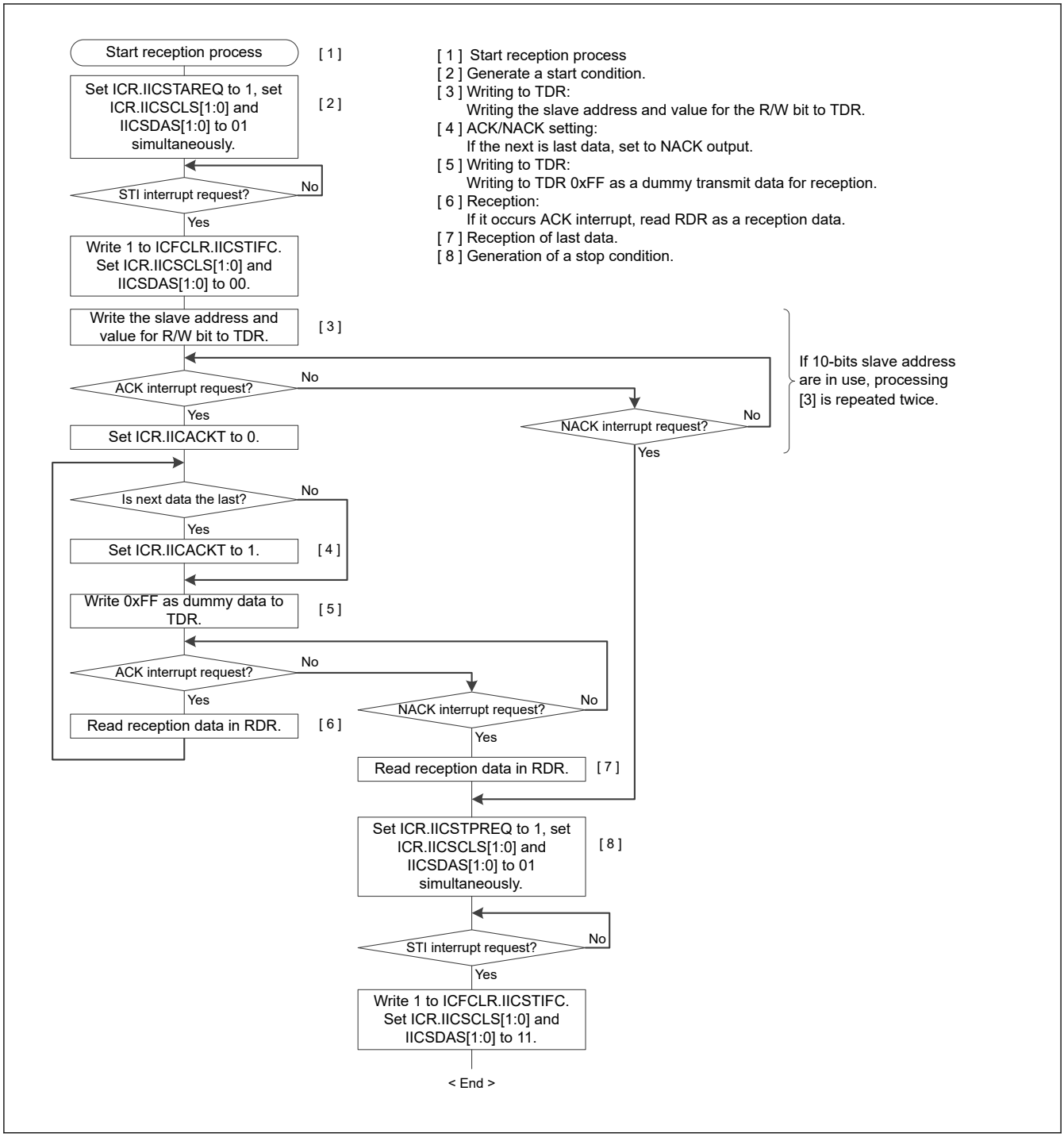


Figure 31.97 Example flow of master reception in simple IIC mode with ACK Interrupts and NACK Interrupts (ICR.IICINTM = 0)

31.9 Operation in Simple SPI Mode

As an extended function, the SCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

Using the settings for Simple SPI mode setting (CCR3.MOD[2:0] bit = 011b) and setting the CCR0.SSE bit to 1 place the SCI in simple SPI mode. However, the SSn pin function on the master side is not required for connection of the device used as the master in simple SPI mode when the configuration only has a single master. Therefore, set the CCR0.SSE bit to 0 in such cases.

Figure 31.98 shows an example of connections for simple SPI mode. Control a general port pin to produce the SSn output signal from the master.

In simple SPI mode, data is transferred in synchronization with clock pulses in the same way as in clock synchronous mode. One character of data for transfer consists of 8 bits of data, and parity bits cannot be appended. The data can be inverted by setting the CCR3.SINV bit to 1.

Because the receiver and transmitter are independent of each other within the SCI module, full-duplex communications are possible, with a shared clock signal. Additionally, because both the transmitter and receiver have a buffered structure, writing the next transmit data while transmission is in progress and reading previously received data while reception is in progress are both possible. This enables continuous transfer.

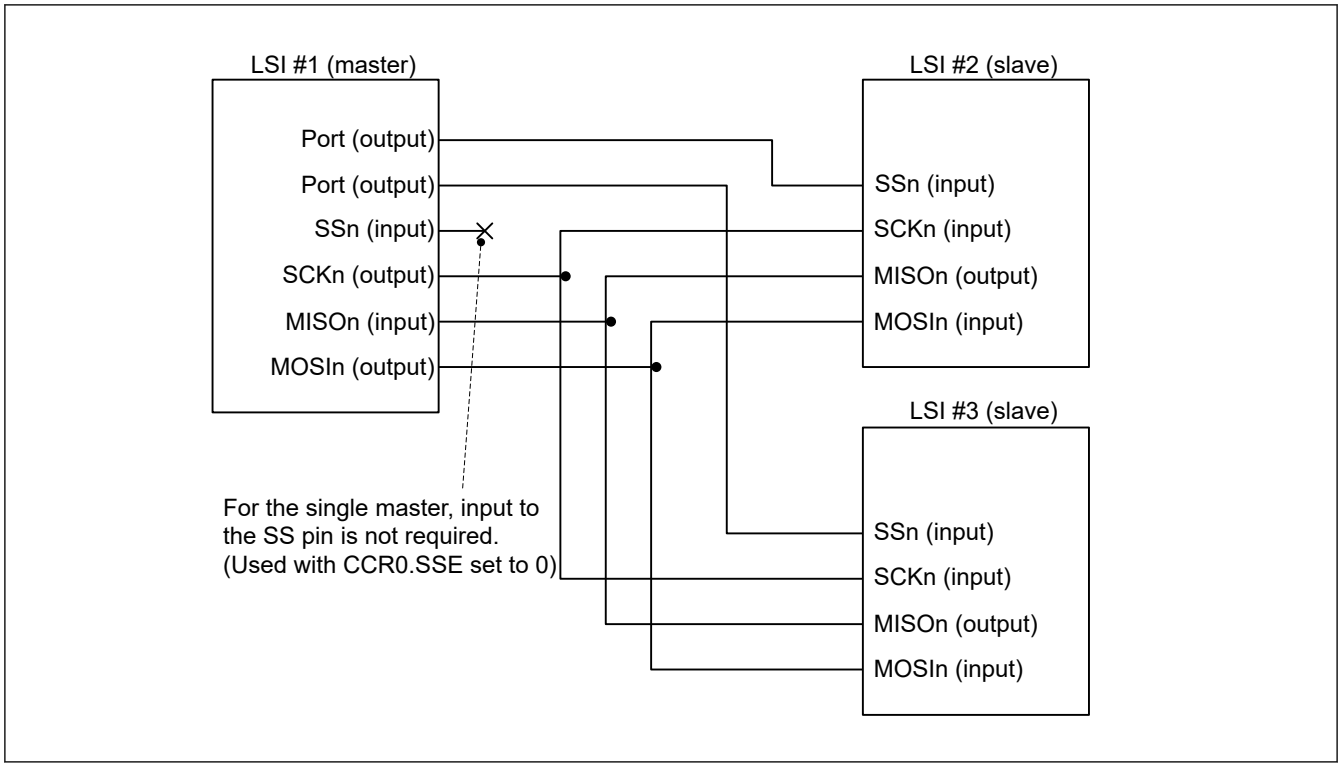


Figure 31.98 Example connections using simple SPI mode in single master mode with CCR0.SSE bit = 0

31.9.1 States of Pins in Master and Slave Modes

The direction (input or output) of pins for the simple SPI mode interface differs according to whether the device is a master (CCR3.CKE[1:0] = 00b or 01b) or slave (CCR3.CKE[1:0] = 10b or 11b).

Table 31.42 lists the relationship between the pin states, mode, and level on the SSn pin.

Table 31.42 States of pins by mode and input level on SSn pin

Mode	Input on SSn pin	State of MOSIn pin	State of MISOn pin	State of SCKn pin
Master mode*1	High level (transfer can proceed)	Output for data transmission*2	Input for received data	Clock output*3
	Low level (transfer cannot proceed)	High-impedance	Input for received data (but disabled)	High-impedance
Slave mode	High level (transfer cannot proceed)	Input for received data (but disabled)	High-impedance	Clock input (but disabled)
	Low level (transfer can proceed)	Input for received data	Output for data transmission	Clock input

Note 1. When there is only a single master (CCR0.SSE = 0), transfer is possible regardless of the input level on the SSn pin. This is equivalent to input of a high level on the SSn pin. The SSn pin is not used and is available for other purposes.
Note 2. The MOSIn pin output is in the high-impedance state when serial transmission is disabled (CCR0.TE bit = 0).
Note 3. The SCKn pin output is in the high-impedance state when serial transmission is disabled (CCR0.TE = 0 and CCR0.RE = 0) in a multi-master configuration (CCR0.SSE = 1).

31.9.2 SS Function in Master Mode

Setting the CKE[1:0] bits in the CCR3 to 00b or 01b selects master mode operation. The SSn pin is not used in single-master configurations (CCR0.SSE = 0), so transmission or reception can proceed regardless of the value of the SSn pin.

When the level on the SSn pin is high in a multi-master configuration (CCR0.SSE = 1), a master device outputs clock signals from the SCKn pin before starting transmission or reception to indicate that there are no other masters or another master is performing reception or transmission.

When the level on the SSn pin is low in a multi-master configuration (CCR0.SSE = 1), there are other masters, and a transmission or reception is in progress. The MOSIn output and SCKn pins are placed in the high-impedance state and starting transmission or reception is not possible. In addition, the value of the CSR.MFF bit is 1, indicating a mode fault error. In a multi-master configuration, start error processing by reading CSR.MFF flag. If a mode fault error occurs while transmission or reception is in progress, transmission or reception does not stop, but the MOSIn and SCKn outputs are in the high-impedance state after completion of the transfer. When the SSn pin input becomes high level, the SCKn pin outputs a clock signal and the MOSIn outputs data. Even if the SCKn pin and the MOSIn pin are in the high impedance state, internal transmission or reception operation continues, but it stops after transmission or reception of a single character is complete. In this case, any of SCIn_TXI, SCIn_RXI, and SCIn_TEI interrupts occurs.

Use a general port pin to produce the SS output signal from the master.

31.9.3 SS Function in Slave Mode

Setting the CCR3.CKE[1:0] bits to 10b or 11b selects slave operation. When the SSn pin is high, the MISOn output pin is in the high-impedance state and clock input through the SCKn pin is ignored. When the SSn pin is low, clock input through the SCKn pin is valid and transmission or reception can proceed.

If the input on the SSn pin changes from low to high during transmission or reception, the MISOn output pin is placed in the high-impedance state. Transmission / reception operation is immediately suspended. If the transmission is in progress, the CSR.TEND flag will not be set, a transmit end interrupt will not be output, and an abnormal stop status will occur. So, do not negate the SSn pin during slave transmission / reception. If an abnormal stop occurs, set CCR0.RE and CCR0.TE to 0 to stop transmission / reception. To resume transmission / reception, set CCR0.RE and CCR0.TE to 1 after at least TCLK × 3 cycles + PCLK × 3 cycles.

31.9.4 Relationship between Clock and Transmit/Receive Data

The CPOL and CPHA bits in the CCR3 register can be used to set up the clock for use in transmission and reception in four different ways. The relation between the clock signal and the transmission and reception of data is shown in Figure 31.99. The relation is the same for both master and slave operation. This is the same as when the level on the SSn pin is high.

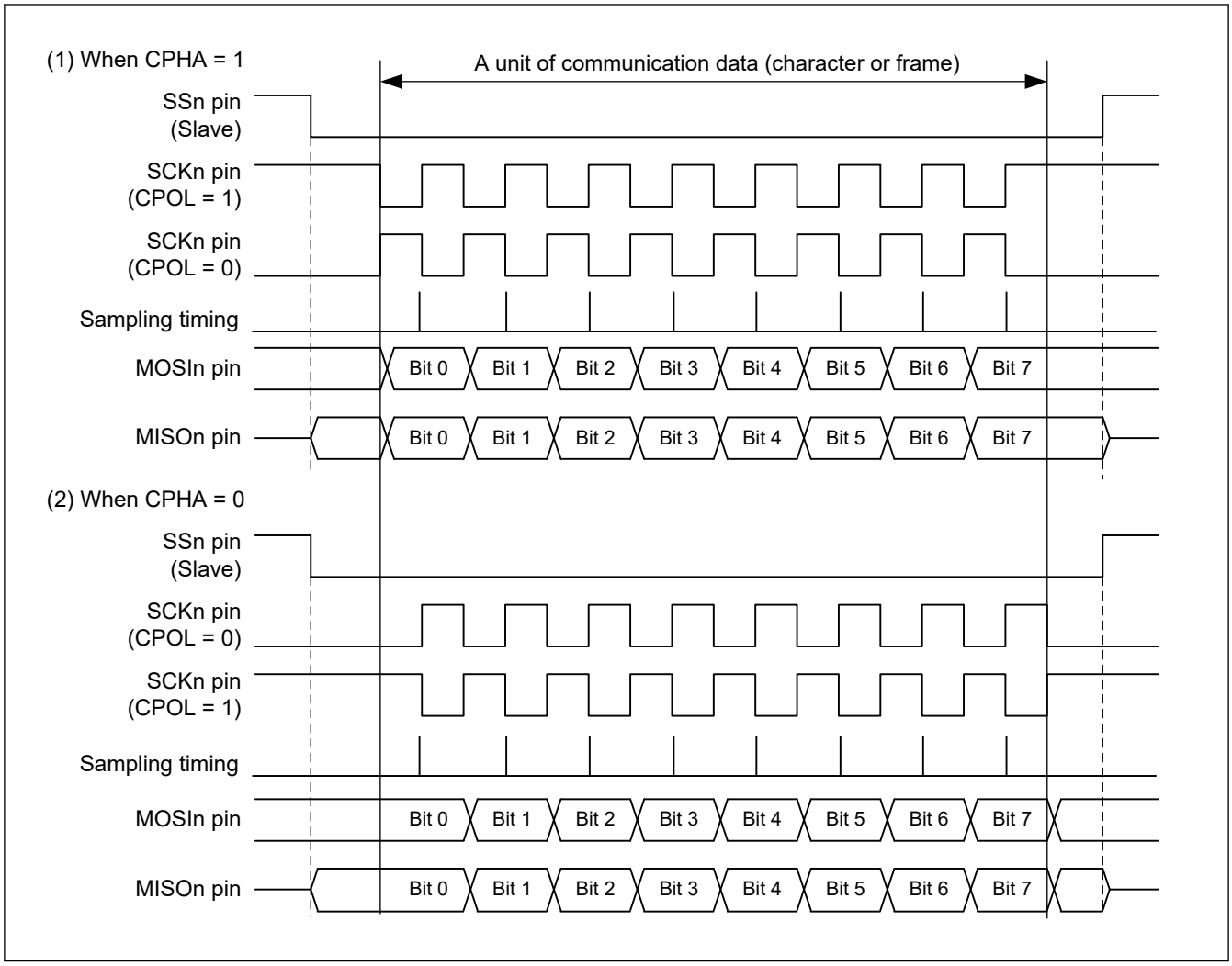


Figure 31.99 Relation between clock signal and transmit or receive data in simple SPI mode

31.9.5 SCI Initialization in Simple SPI Mode

Initialization in simple SPI mode is the same as in clock synchronous mode. See section 31.6.3. SCI Initialization in Clock Synchronous Mode for an example initialization flow. The CPOL and CPHA bits in the CCR3 register must be set to ensure that the clock signal is suitable for both master and slave devices.

Always initialize the CCR0 register before making any changes to the operating mode or transfer format.

Note: Only the RE bit is set to 0. The CSR.ORER, FER, PER, and RDR flags are not initialized.

Changing the value of the TE bit from 1 to 0 or from 0 to 1 when the TIE bit in the CCR0 register is 1 at the same time, leads to the generation of a transmit data empty interrupt (SCIn_TXI).

31.9.6 Transmission and Reception of Serial Data in Simple SPI Mode

In master operation, ensure that the SSn pin of the slave device on the other side of the transfer is at the low level before starting the transfer and at the high level on completion of the transfer. In multiple master operation with CCR0.SSE = 1 even in master mode, a mode fault error will occur if the SSn pin goes low. Therefore, make sure that no mode fault error has occurred before starting communication, and start communication, and make sure that no mode fault error has occurred even after communication ends. If a mode fault error has occurred, communication may be incomplete, so measures such as retransmission are required. Otherwise, the procedures are the same as in clock synchronous mode.

In slave mode, it operates according to the SSn pin input level. Other steps are the same as those of clock synchronous mode.

31.9.7 Reception Sampling Timing Adjustment Function in Simple SPI Mode with internal clock used

The reception sampling timing adjustment function in simple SPI mode is the same as the reception sampling timing adjustment function in clock synchronous mode. For the description of operation, see section 31.6.7. Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with internal clock used.

31.10 Bit Rate Modulation Function

Using the bit rate modulation function, the bit rate can be evenly corrected using the number specified in the MDDR register among 256 clock cycles of internal clocks which is selected by the CKS[1:0] bits in CCR2.

Figure 31.100 shows an example where the PCLK is selected in the CKS[1:0] bits in CCR2, the BRR bit is set to 0, and the MDDR is set to 160 in Asynchronous mode. In this example, the cycle of the base clock is evenly corrected (256/160) and the bit rate is also corrected (160/256).

Note: Enabling an internal clock causes bias, and expansion and contraction are generated in the pulse width of the internal base clock.

Do not use this function in clock synchronous mode, simple SPI mode, Smart Card Interface mode, Manchester mode and Simple LIN mode.

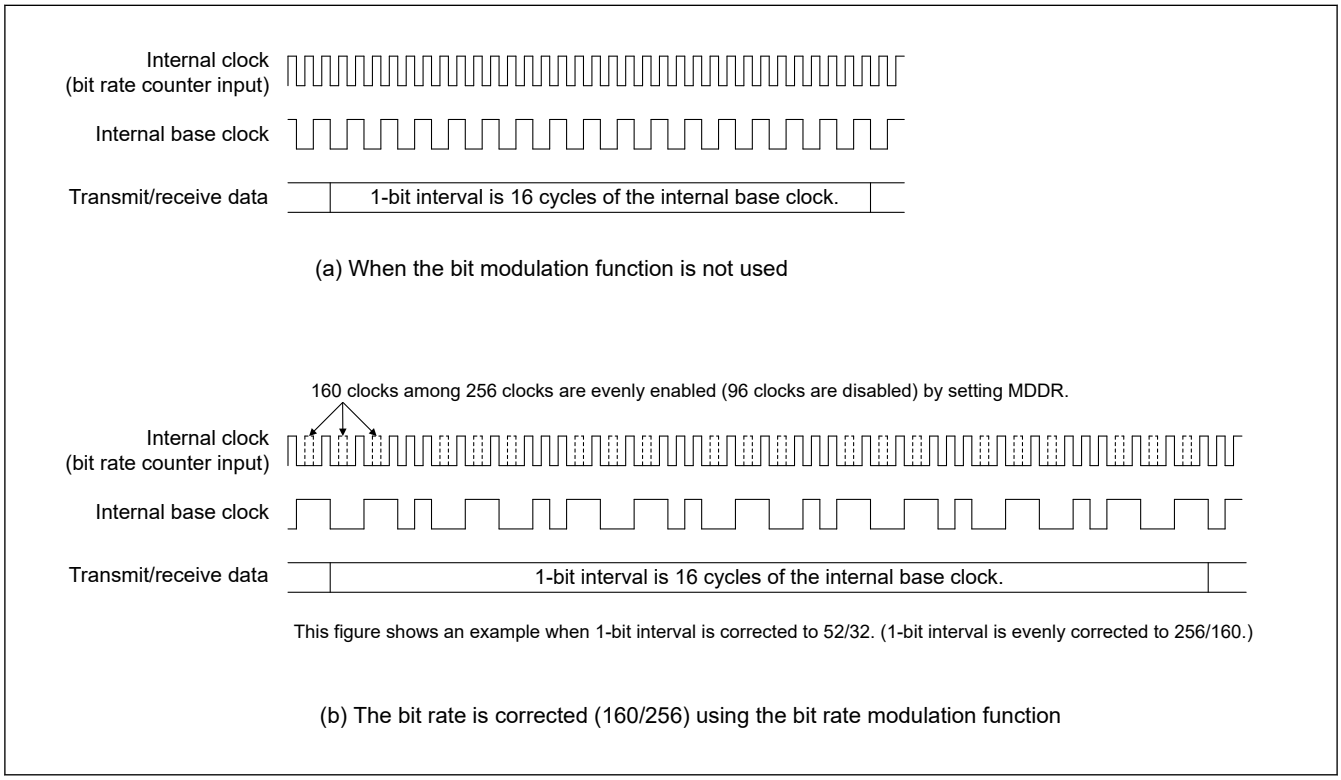


Figure 31.100 Example internal base clock when bit rate modulation function is used

31.11 Simple LIN mode

As an extended function of the SCI, the SCI supports the serial communication protocol (Figure 31.101) consisting of a Start Frame and an Information Frame as Simple LIN. Simple LIN mode is enabled by the CCR3.MOD[2:0] = 110b. Since the Simple LIN mode uses the same circuit as the Asynchronous mode for transmission / reception control other than Break Field, the basic communication settings required for the Asynchronous mode are also required for the Simple LIN mode.

(For the setting value when using simple LIN, see the explanation in section 31.2. Register Descriptions. In particular, note that CCR3.RXDESEL needs to be changed from the initial value and set to 1.)

The Start Frame consists of a Break Field, Control Field 0, and Control Field 1. The Information Frame can be configured with some Data Fields, a Checksum Field.

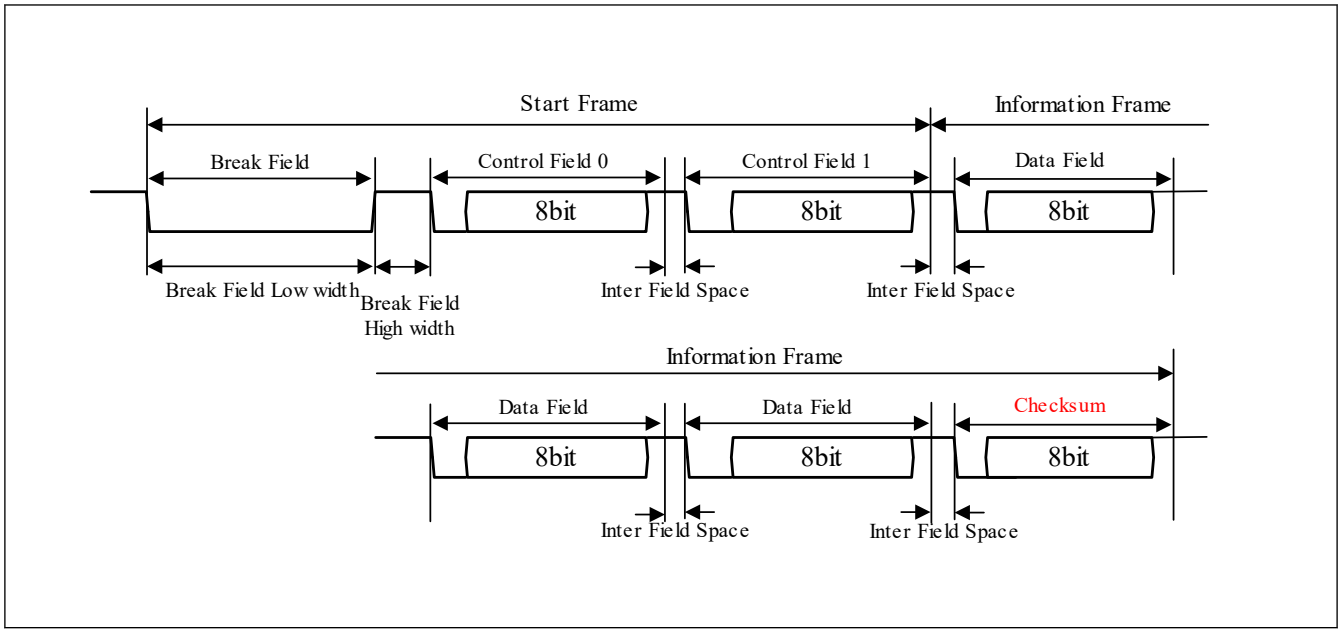


Figure 31.101 Simple LIN Protocol Example

The following describes operations when the Simple LIN is used. In this section, operations are described with the following conditions:

Communication pin (RXDn / TXDn) level inversion function: OFF (RINV = TINV = 0)

When using the simple LIN with the communication pin (RXDn / TXDn) level inversion function enabled, replace the RXDn and TXDn signal levels with their inverted levels.

31.11.1 Simple LIN Start Frame Transmission

Figure 31.102 shows an example of transmission of the Start Frame consisting of a Break Field, Control Field0, and Control Filed1. (Omit Break Field and Control Field0 according to the Start Frame configuration.)

Figure 31.103 shows a flowchart for Start Frame transmission.

The SCI operates as follows during Start Frame transmission.

1. Make the initial settings for the SCI according to the SCI initialization flow (Figure 31.68) in Asynchronous mode. In Simple LIN mode, do not set CCR0.TE and TIE to 1 at the same time to avoid SCIn_TXI output before the Break Field. Therefore, perform the following two steps sequentially to set the SCI initialization flow (Asynchronous mode) procedure [9].
   - Set the bits except CCR0.TIE. (CCR0.TIE = 0, CCR0.TE = 1, and CCR0.RE = 0)
   - Set CCR0.TIE to 1.
2. When 1 is written to TCST, the Break Field output timer starts counting and outputs a low level (Break Field) from the TXDn pin for the period set in XCR2.BFLW[15:0]. A timer count clock source can be selected by XCR0.TCSS[1:0]. Writing 0 to XCR1.TCST suspends output of the Break Field. After the suspension, set CCR0.TE = 0 and turn off the transmission.
3. When the Simple LIN module timer count value matches the set XCR2.BFLW[15:0] value, the timer stops counting and inverts the TXDn pin output level, and the XSR0.BFOF flag is set to 1*1. Furthermore, if XCR0.BFOIE has been set to 1 at this time, a SCIn_TXI interrupt is generated.
4. After the SCIn_TXI interrupt and confirming XSR0.BFOF = 1, write the transmitted data then the Control Field 0 data is transmitted using the SCI*2.
5. After the Control Field 0 data has been transmitted, write the Control Field 1 data to TDR. And it is transmitted.
6. After the Control Field 1 data has been transmitted, the Information Frame data is transmitted.

Note 1. After XSR0.BFOF is set to 1, if 1 is written to XCR1.TCST without clearing it, no SCIn_TXI interrupt is output at the end of Break Field transmission. Clear XSR0.BFOF before writing 1 to XCR1.TCST.

Note 2. LIN communication requires a Break Delimiter (IDLE period) of 1 bit or more from the end of Break Field output until the next data transmission starts. For this reason, the Break Delimiter length is counted upon completion of Break Field output. If transmit data is written while the Break Delimiter length is being counted, transmission does not start until the Break Delimiter length counting is completed. When transmit data is written after the Break Delimiter length has been counted, transmission starts at the same timing as normal data transmission.

Break Delimiter length count time after Break Field output:

1-bit to 2-bit length (CCR3.STP = 0)

2-bit to 3-bit length (CCR3.STP = 1)

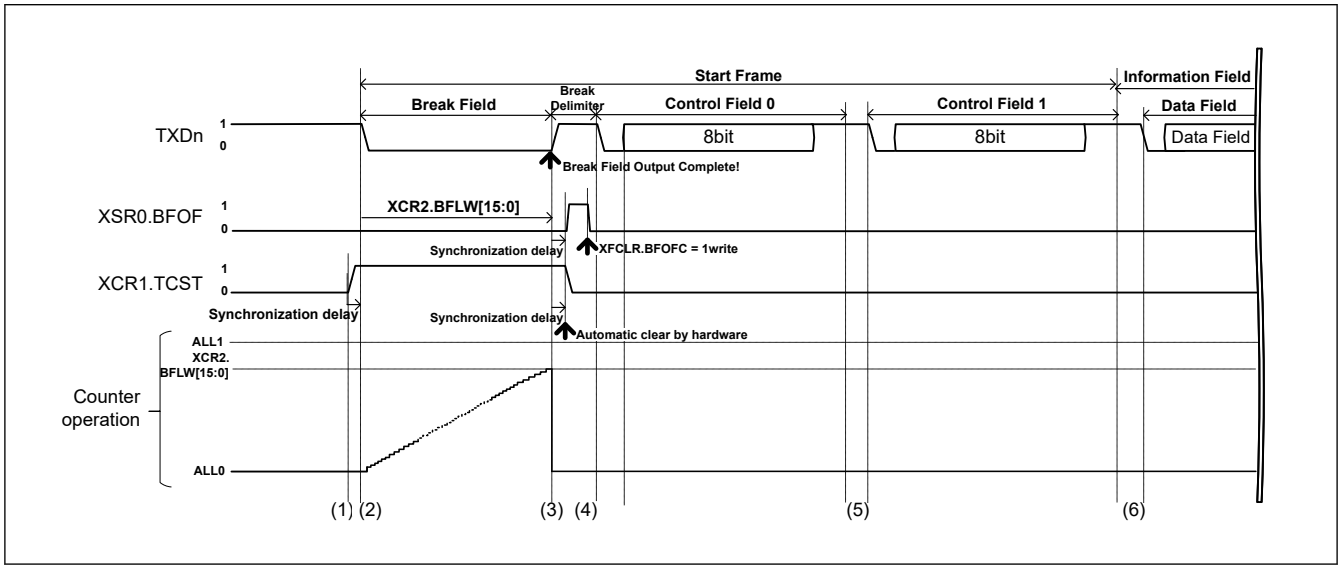


Figure 31.102 Start Frame Transmission Example

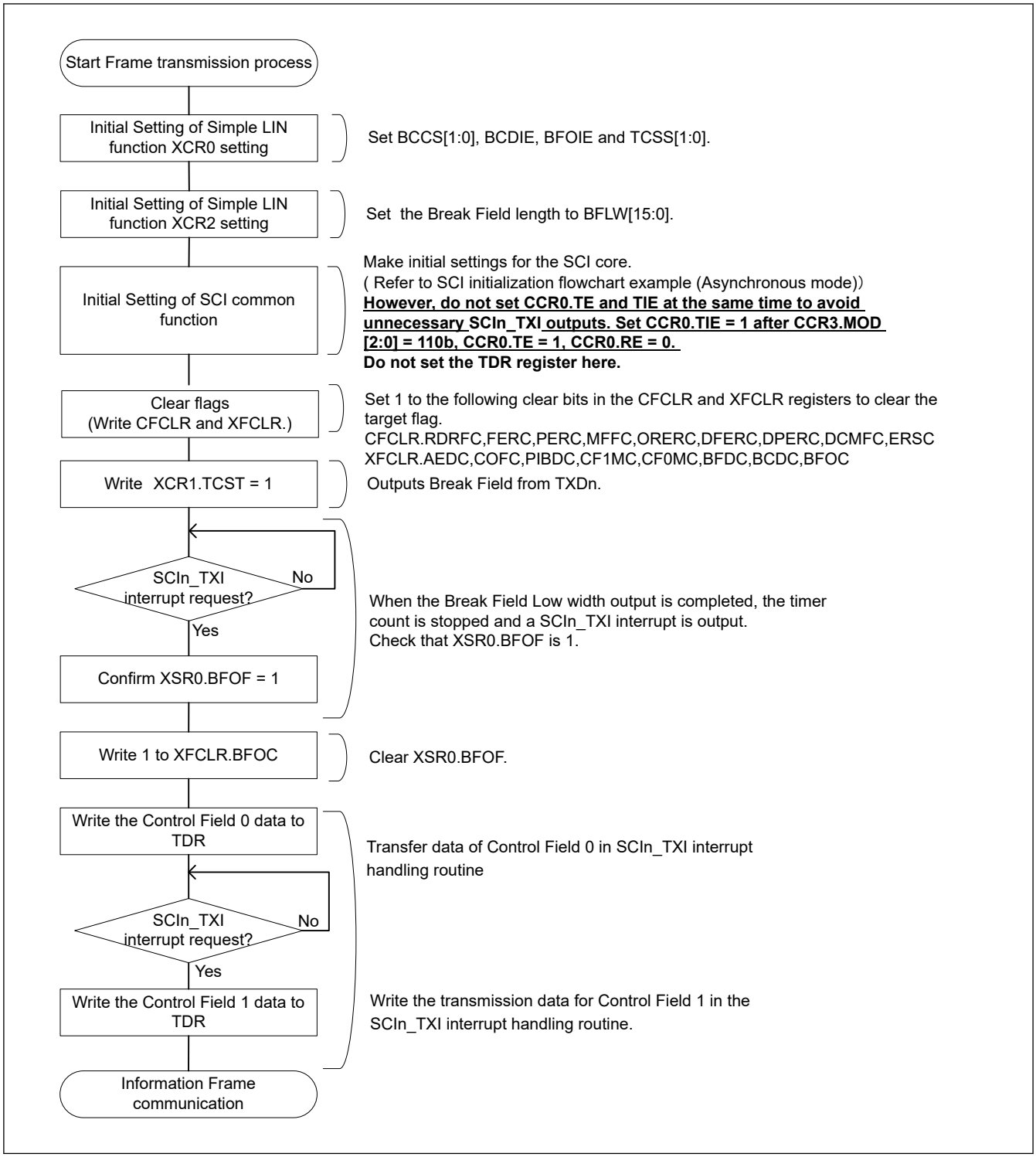


Figure 31.103 Start Frame Transmission Flowchart Example

31.11.2 Simple LIN Start Frame Reception

The SCI can detect Start Frames configured as shown in Figure 31.104.

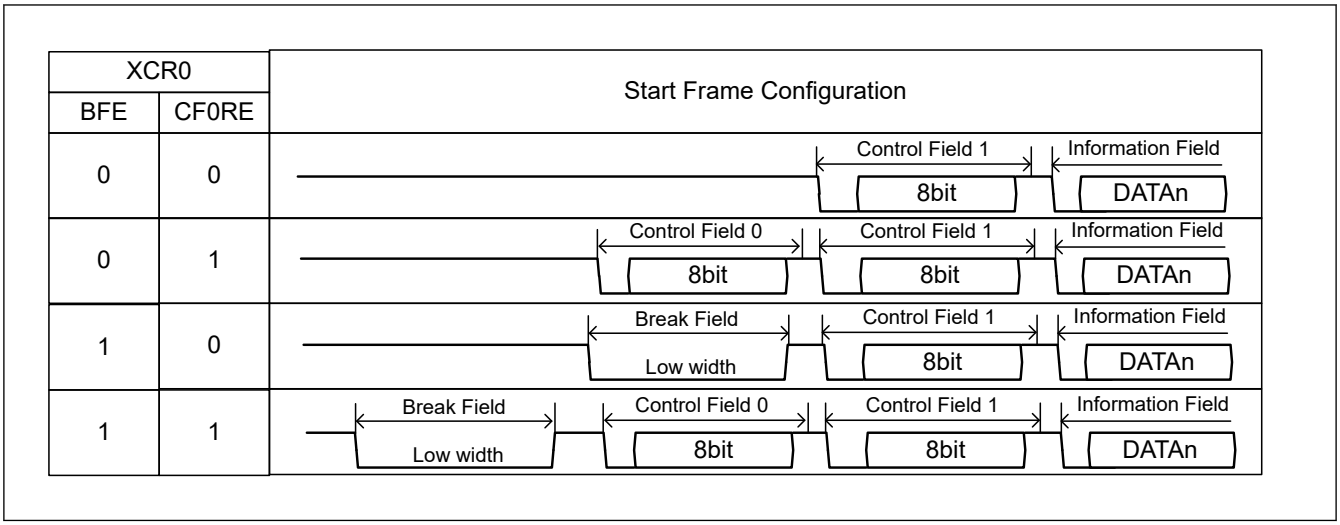


Figure 31.104 Start Frame Configuration

(1) Simple LIN normal reception of Start Frame (PIB not used)

Figure 31.105 shows an example of normal reception of the Start Frame consisting of a Break Field, Control Field0, and Control Field1. Figure 31.106 shows an example of reception to detect the Break Field during Control Field 1. Figure 31.107 shows a flowchart to receive the Start Frame, and Figure 31.108 shows a state transition diagram.

When receiving the Start Frame, the SCI operates as follows. Omit the processing of Break Field and Control Field0 according to the Start Frame configuration.

1. Writing 1 to XCR1.SDST makes it possible to detect the Start Frame. When XCR0.BFE = 1, RXDn input to the SCI core is disabled until the Break Field is detected (because XSR0.RXDSF is set to 1). Once the Break Field is detected, RXDn input can be received to the SCI core (XSR0.RXDSF = 0).
2. When a low level is input from the RXDn pin, the Break Field detection count starts. A timer count clock source can be selected by XCR0.TCSS[1:0].
3. When a low level that is equal to or longer than the period set in XCR2.BFLW[15:0] is input from the RXDn pin, it is determined as Break Field. At this time, XSR0.BFDF is set to 1. If XCR0.BFDIE has been set to 1 at this time, a SCIn_BFD interrupt is generated.
   The timer continues counting until the RXDn rising edge or counter overflow.
4. After the Break Field is detected, when the input level from the RXDn pin becomes high, the count value is captured to XSR1.TCNT[15:0] when BMEN = 0. At this time, XSR0.RXDSF is cleared to 0 and the SCI core starts receiving the RXDn input.
5. The SCI core starts receiving Control Field 0. Because the simple LIN continuously counts the edge interval, it determines a low level that is equal to or longer than the period set in XCR2.BFLW[15:0] as detection of the Break Field. When the Break Field is detected in the Control Field 0 phase, the SCI core waits for reception of Control Field 0 again (Figure 31.106).
6. When Control Field 0 has been received, an SCIn_RXI interrupt is generated and the Control Field 0 data is stored in XSR0.CF0RD[7:0]. When the received data matches the set XCR2.CF0D[7:0] value, XSR0.CF0MF is set to 1. If the received data differs from the set XCR2.CF0D[7:0] value, the SCI transitions to the state before the Break Field is detected.
7. The SCI core starts receiving Control Field 1. When BFE = 1, the Break Field detection function is continuously enabled while SDST = 1 as in the case of Control Field 0. When the Break Field is detected in the Control Field 1 phase, the SCI core waits for reception of Control Field 0 again.
8. When Control Field 1 has been received, an SCIn_RXI interrupt is generated and the Control Field 1 data is stored in XSR0.CF1RD[7:0]. When the received data matches the set XCR1.PCF1D[7:0] value or the set XCR1.SCF1D[7:0] value, XSR0.CF1MF is set to 1. If the received Control Field 1 data matches neither the set XCR1.PCF1D[7:0] value nor the set XCR1.SCF1D[7:0] value, the SCI transitions to the state before the Break Field is detected.
9. The SCI core performs Information Frame communication.
10. When communication is completed, write 0 to XCR1.SDST and 0 to CCR0.RE to stop reception.

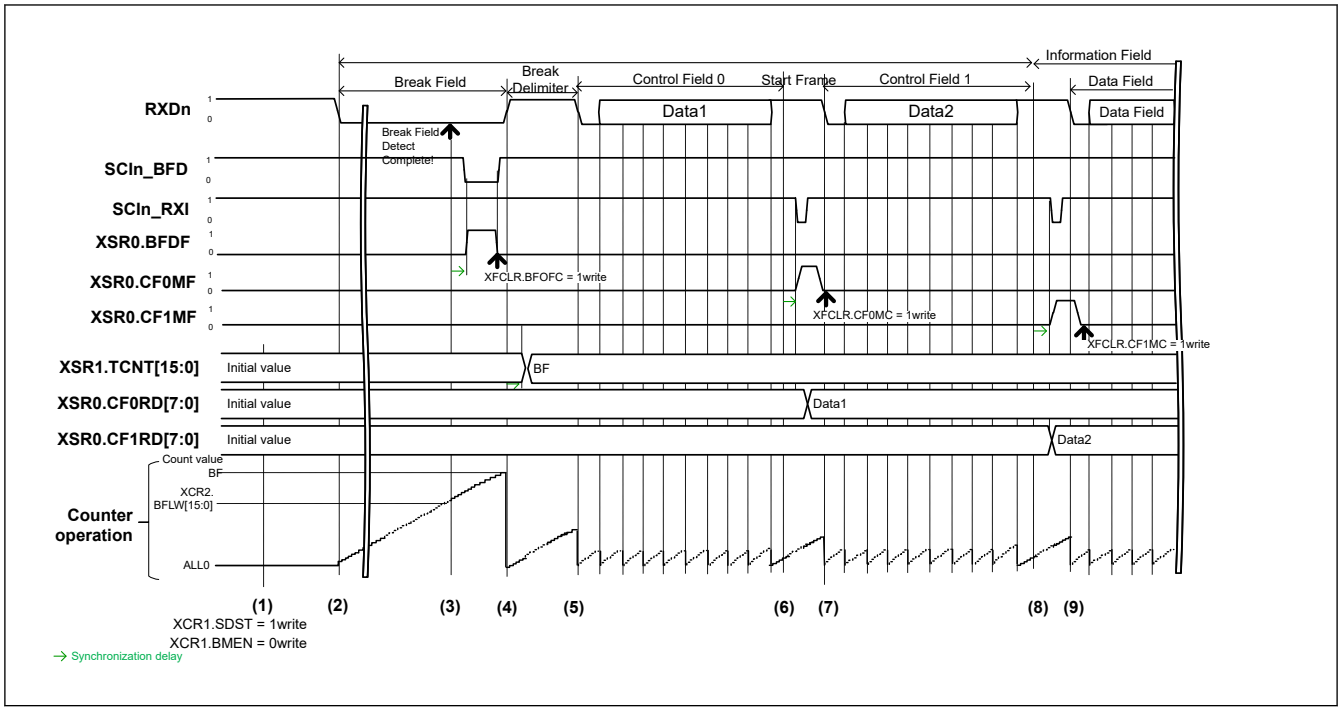


Figure 31.105 Normal Reception Example of Start Frame (PIB Not Used)

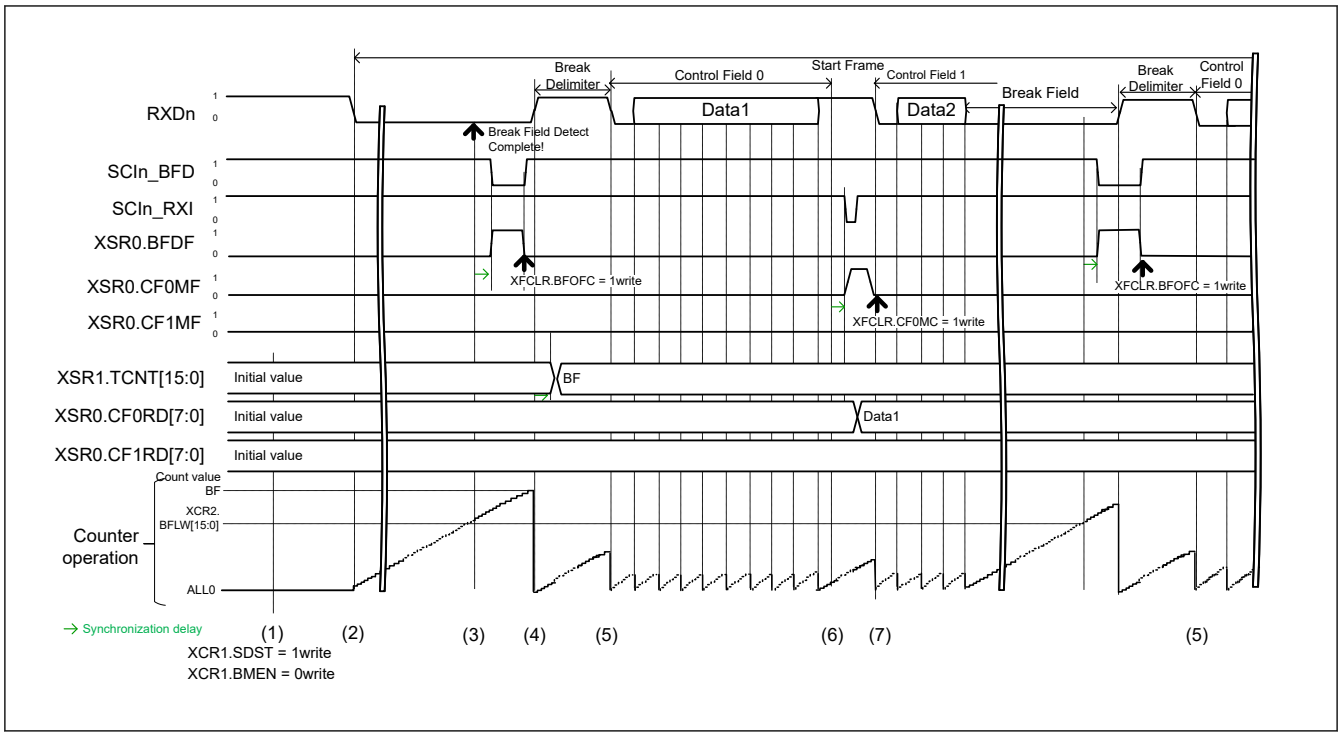


Figure 31.106 Start Frame Reception Example (PIB Not Used) Break Field Detected during Control Field 1

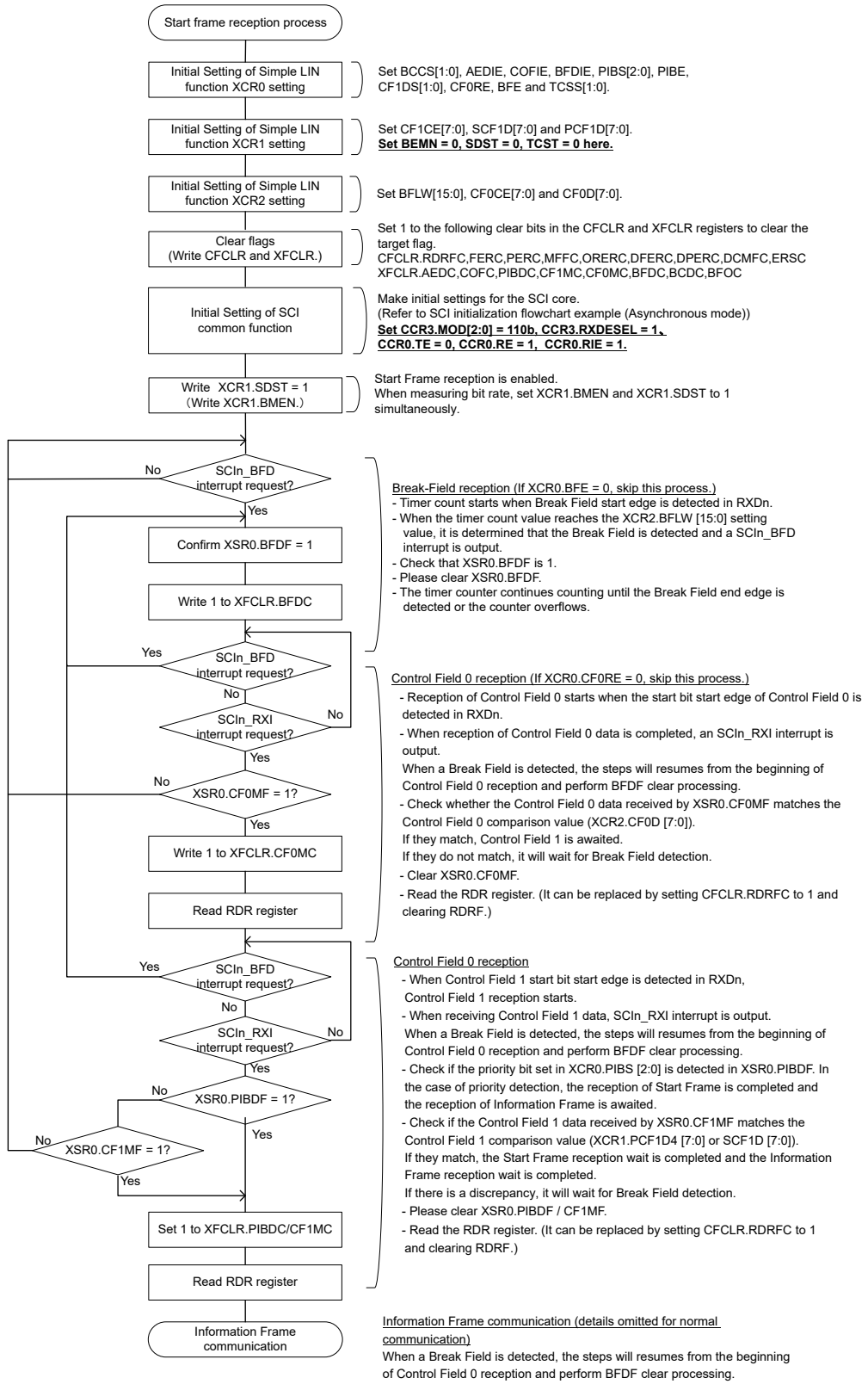


Figure 31.107 Example of Start Frame Reception Flowchart

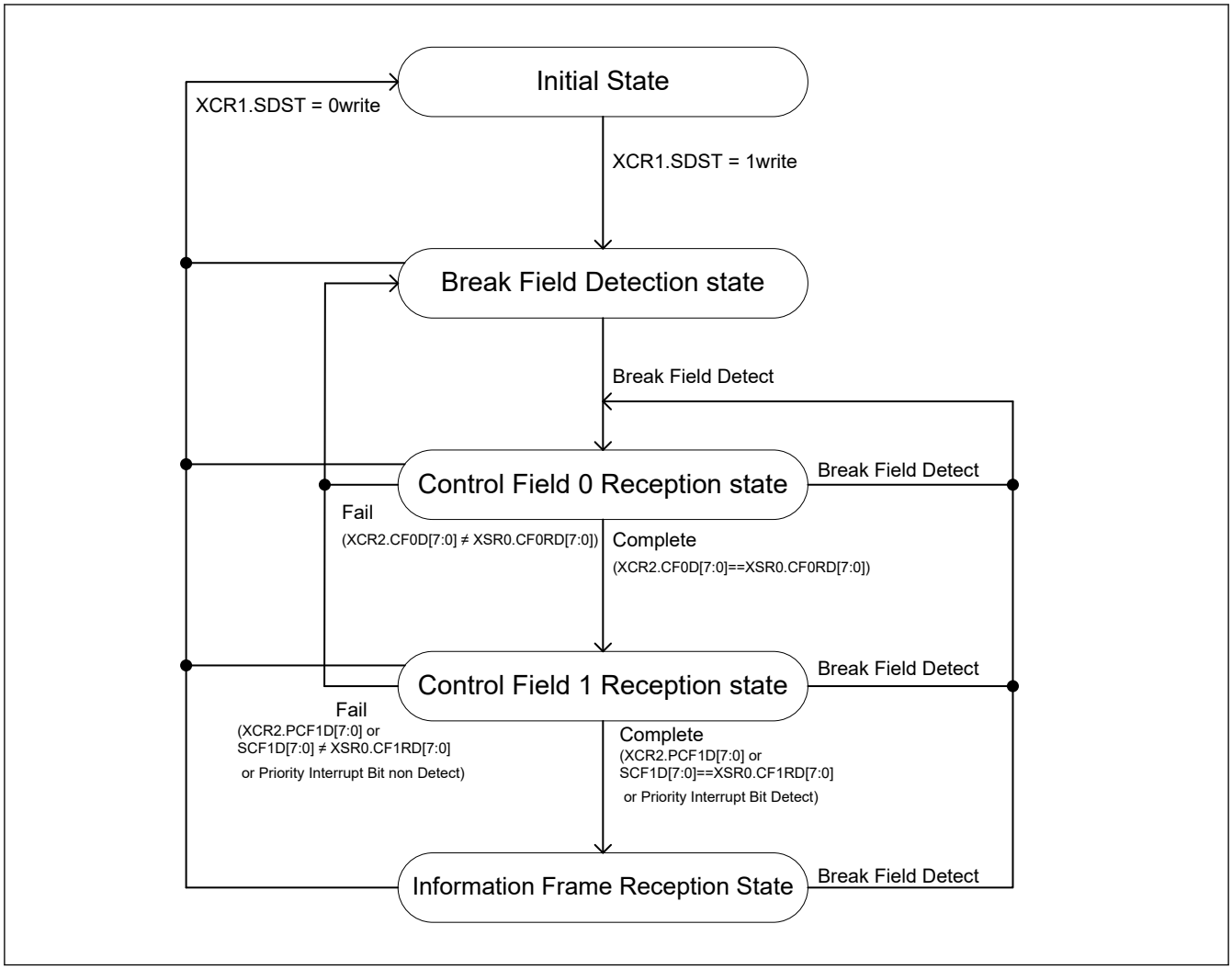


Figure 31.108 State Transition Diagram of Start Frame Reception

(2) Simple LIN Start Frame reception (using the priority interrupt bit)

Figure 31.109 shows an example of Start Frame reception using the priority interrupt bit. The priority interrupt bit is enabled by setting XCR0.PIBE to 1.

The SCI operates as follows during Start Frame reception using the priority interrupt bit.

Steps (1) to (7) are the same as steps (1) to (7) in the Start Frame reception example in Figure 31.105.

(8) When the value specified in the XCR0.PIBS[2:0] bits matches the set XCR1.PCF1D[7:0] value, XSR0.PIBDF is set to 1 and the SCI core performs communication of the Information Frame. If the data received in Control Field 1 matches neither the set XCR1.PCF1D[7:0] value nor the set XCR1.SCF1D[7:0] value and the priority interrupt bit is not detected, the SCI transitions to the state before the Break Field is detected.

(9) Communicate information frame at the SCI core.

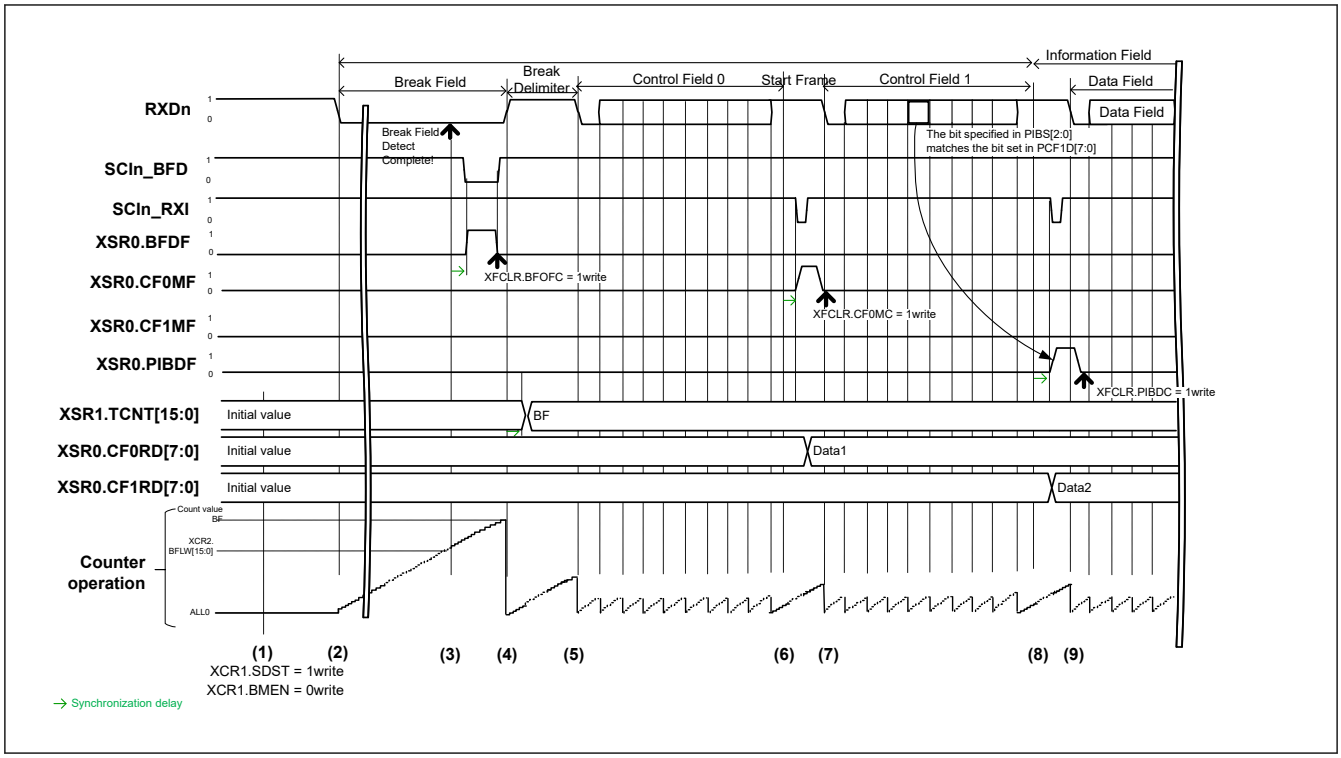


Figure 31.109 Start Frame Reception Example (Priority Interrupt Bit Used)

31.11.3 Simple LIN Bus Conflict Detection Function

In Simple LIN mode (CCR3MOD[2:0] = 110) when TE = 1, the bus conflict detection function works during Break Field output and during data transmission.

Figure 31.110 shows an operation example of the bus conflict detection function. The TXDn pin output and the RXDn pin input are sampled by the bus conflict detection clock set in XCR0.BCCS[1:0]. When a mismatch occurs three times in a row, XSR0.BCDF is set to 1, and if XCR0.BCDIE has been set to 1 at this time, an SCIn_ERI interrupt is generated.

When an SCIn_ERI interrupt is generated, stop transmission according to Figure 31.111. Check the bus state to decide whether to resume transmission.

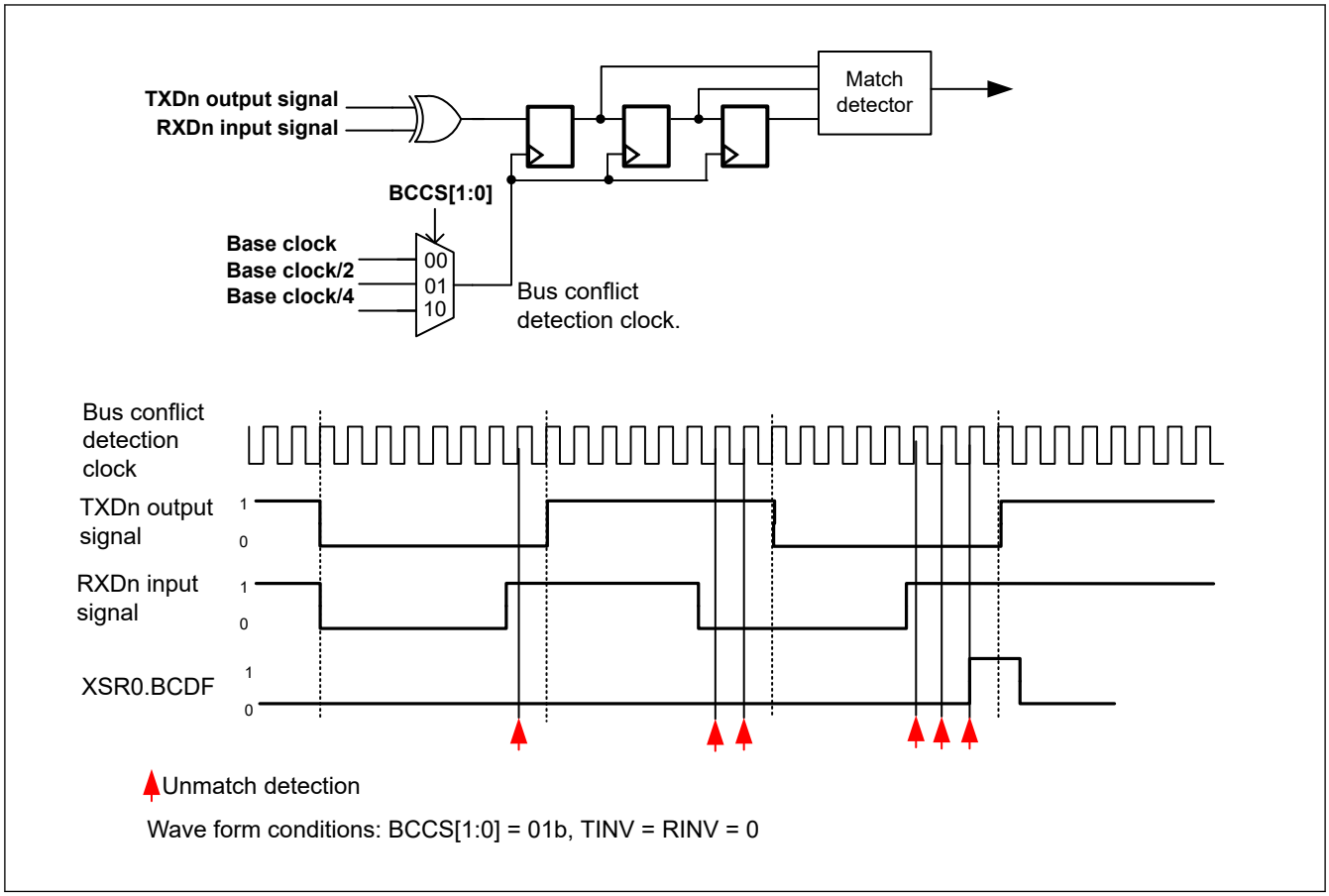


Figure 31.110 Operation Example of the Bus Conflict Detection Function

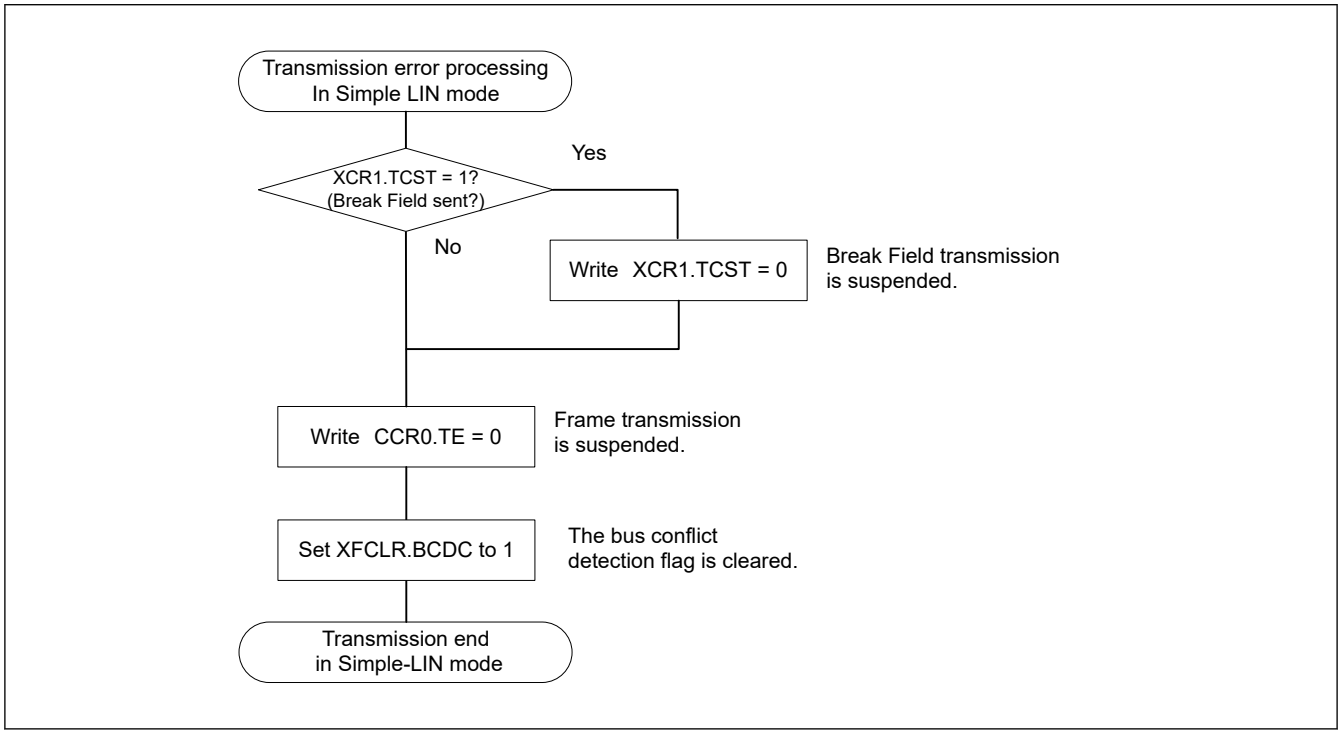


Figure 31.111 SCIn_ERI Interrupt handling flow at transmission in Simple LIN mode

31.11.4 Simple LIN Bit Rate Measurement Function

This function measures a bit rate between the effective edges of the input signal from the RXDn pin. Figure 31.112 shows an operation example of the bit rate measurement function.

1. Writing 1 to XCR1.SDST and XCR1.BMEN enables bit rate measurement. When this bit is set to 1, the valid edge interval of Control Field 0 and Control Field 1 data is measured. However, bit rate is not measured between the Break Field and the Break Delimiter. Set XCR1.BMEN and XCR1.SDST to 1 simultaneously, only when measuring bit rate.
2. Because bit rate is not measured in the Break Field, the effective edge detection flag is not set to 1 at the rising edge at the end of the Break Field, and the counter capture value is not stored in XSR1.TCNT[15:0].
3. The counter starts counting from the falling edge of the start bit in Control Field 0. The Break Delimiter count value is not captured in XSR1.TCNT[15:0].
4. The rising edge of the start bit is detected as an effective edge, and then the XSR0.AEDF flag is set to 1. If XCR0.AEDIE has been set to 1 at this time, an SCIn_AED interrupt is output. The start bit count value is stored in XSR1.TCNT[15:0]. The XSR1.TCNT[15:0] value is retained until the effective capture value is read.
5. Even if an effective edge is input from the RXDn input pin, the count value of this effective edge timing is not captured because the XRS1.TCNT[15:0] value has not been read and retention has not been released. In this case, an SCIn_AED interrupt is not output.
6. The XSR1.TCNT[15:0] value is read. Then the retention of XSR1.TCNT[15:0] is released and the XSR0.AEDF flag is cleared by hardware.
7. Because the retention of XSR1.TCNT[15:0] has been released, the count value is captured at the effective edge and is retained. At the same time, the XSR0.AEDF flag is set to 1, and if XCR0.AEDIE has been set to 1, an SCIn_AED interrupt is output. The bit rate can be adjusted by calculating it from the count value between effective edges by software and by changing the SCI settings.
8. To disable bit rate measurement, write 0 to XCR1.BMEN.
9. The XSR0.AEDF value and the XSR1.TCNT[15:0] value remain unchanged at the effective edge timing because the bit rate measurement function is disabled.

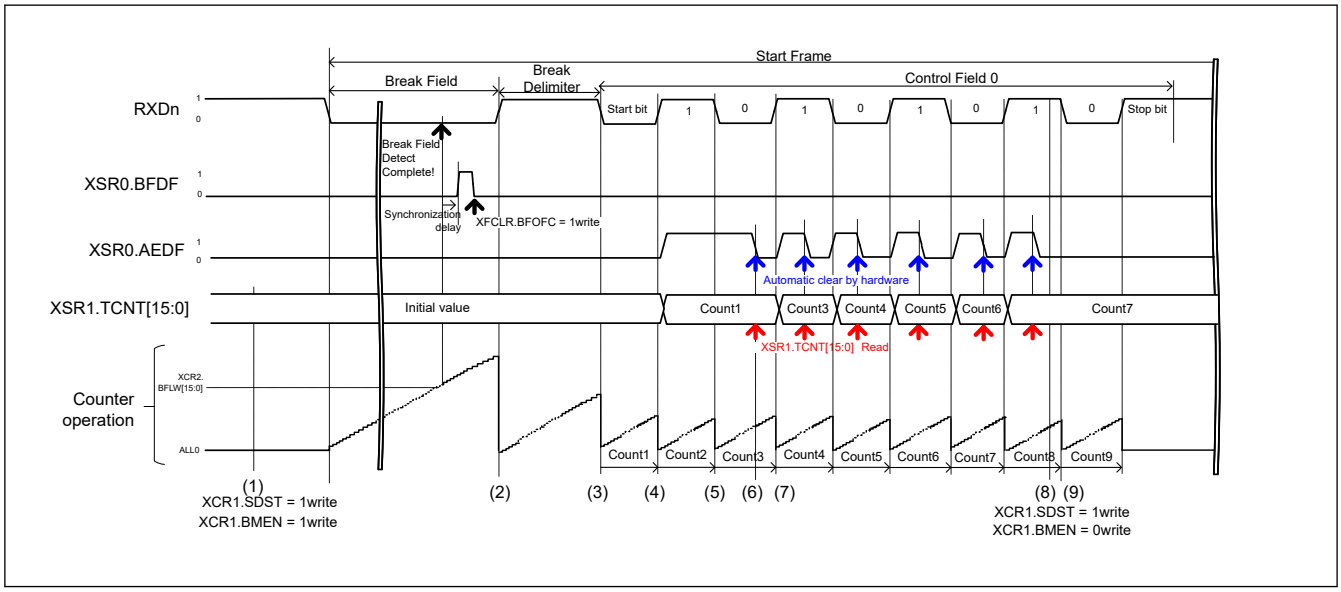


Figure 31.112 Operation Example of the Bit Rate Measurement Function

31.12 Interrupt Sources

31.12.1 Buffer Operation for SCIn_TXI and SCIn_RXI Interrupts

If the conditions for an SCIn_TXI and SCIn_RXI interrupt are satisfied while the interrupt status flag in the ICU is 1, the ICU does not output the interrupt request but saves it internally with a capacity for retention of one request per source.

31.12.2 Interrupts in Asynchronous, Manchester, Clock Synchronous, and Simple SPI Modes

(1) Non-FIFO selected

Table 31.43 lists interrupt sources in Asynchronous mode, Manchester mode, clock synchronous mode, and simple SPI mode.

A different interrupt vector can be assigned to each interrupt source. Individual interrupt sources can be enabled or disabled with the enable bits in the CCR0 register.

If the CCR0.TIE bit is 1, an SCIn_TXI interrupt request is generated when transmit data is transferred from the TDR register to the TSR register. An SCIn_TXI interrupt request can also be generated by using a single instruction to set the CCR0.TE and CCR0.TIE bits to 1 at the same time. An SCIn_TXI interrupt request can activate the DTC or DMAC to handle data transfer.

An SCIn_TXI interrupt request is not generated by setting the CCR0.TE bit to 1 when CCR0.TIE is 0 or by setting the CCR0.TIE bit to 1 when the CCR0.TE is 1.*1

When new data is not written by the time of transmission of the last bit of the current transmit data and CCR0.TEIE is 1, the CSR.TEND flag is set to 1 and an SCIn_TEI interrupt request is generated. Additionally, when CCR0.TE is 1, the CSR.TEND flag retains the value 1 until more transmit data is written to the TDR register, and setting CCR0.TEIE to 1 leads to the generation of an SCIn_TEI interrupt request.

Writing data to the TDR register leads to clearing of the CSR.TEND flag and, after a certain time, discarding of the SCIn_TEI interrupt request.

If the CCR0.RIE bit is 1, an SCIn_RXI interrupt request is generated when received data is stored in the RDR register. An SCIn_RXI interrupt request can activate the DTC or DMAC to handle data transfer.

Setting any of the CSR.ORER, FER, PER or MSR.MER*2, SYER*2, PFER*2, and SBER*2 flags to 1 when the CCR0.RIE bit is 1 leads to the generation of an SCIn_ERI interrupt request.

An SCIn_RXI interrupt request is not generated in this case. Clearing all these flags (ORER, FER, PER, MER*2, SYER*2, PFER*2 and SBER*2) leads to discarding of the SCIn_ERI interrupt request.

Note 1. To temporarily prohibit SCIn_TXI interrupts on transmission of the last of the data when a new round of transmission is to be started, after handling the transmission-completed interrupt, control activation of the interrupt by using the interrupt request enable bit in the ICU rather than using the CCR0.TIE bit. This approach can prevent the suppression of SCIn_TXI interrupt requests in the transfer of new data.

Note 2. MER, SYER, PFER, and SBER work as a factor of SCIn_ERI interrupt only in Manchester mode. SYER, PFER, and SBER also only work if its enable bits (SYEREN, PFEREN, SBEREN in MECR) are set to "1".

(2) FIFO selected

Table 31.44 lists interrupt sources in FIFO selected mode.

If the CCR0.TIE bit is 1, an SCIn_TXI interrupt request is generated when the stored amount of data in the transmit-FIFO (TDR) register becomes the threshold value indicated in FCR.TTRG or below. An SCIn_TXI interrupt request can also be generated by using a single instruction to set the CCR0.TIE and CCR0.TE bits to 1 simultaneously or by setting CCR0.TIE to 1 when CCR0.TE is 1.

An SCIn_TXI interrupt request is not generated by setting CCR0.TE to 1 when CCR0.TIE is 0 or by setting the CCR0.TIE bit to 1 while the setting of the CCR0.TE bit is 1.

If CCR0.TEIE is 1 and if the next data is not written to the transmit-FIFO (TDR) register by the time the last bit of the transmit data is sent, the CSR.TEND flag is set to 1 and the SCIn_TEI interrupt request is generated.

If CCR0.RIE is 1, the SCIn_RXI interrupt request is generated when the stored amount of data in the transmit-FIFO (TDR) register is equal to or greater than the threshold value indicated in FCR.RTRG. When RTRG is 0, an SCIn_RXI interrupt does not occur even when the amount of data in the receive FIFO is equal to 0.

If the CCR0.RIE bit is 1, when the CSR.ORER flag is set to 1 or data with a framing error or a parity error is stored in the transmit-FIFO (TDR) register, the SCIn_ERI interrupt request is generated. When the amount of data stored in the transmit-FIFO (TDR) register is at the threshold value or above, the SCIn_RXI interrupt request is also generated. The SCIn_ERI interrupt request can be canceled, in which case CSR.ORER, FER, and PER flags are all cleared.

Table 31.43 SCI interrupt sources with non-FIFO selected

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC or DMAC activation
SCIn_ERI (n = 0 to 4, 9)	Receive error	CSR.ORER, CSR.FER, CSR.PER, CSR.DFER, CSR.DPER, (MSR.MER, MSR.SYER, MSR.PFER, MSR.SBER)*1	CCR0.RIE	Not possible
SCIn_RXI (n = 0 to 4, 9)	Receive data full	CSR.RDRF	CCR0.RIE	Possible
	Address match	CSR.DCMF	CCR0.RIE	Possible
SCIn_TXI (n = 0 to 4, 9)	Transmit data empty	CSR.TDRE	CCR0.TIE	Possible
	TE = 0->1 detection			
SCIn_TEI (n = 0 to 4, 9)	Transmit end	CSR.TEND	CCR0.TEIE	Not possible

Note: Only SCI0 supports Manchester mode.
Note 1. MER, SYER, PFER, and SBER work as a factor of SCIn_ERI interrupt only in Manchester mode. SYER, PFER, and SBER also only work if its enable bits (SYEREN, PFEREN, SBEREN in MCR) are set to 1.

Table 31.44 SCI interrupt sources with FIFO selected

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC or DMAC activation
SCIn_ERI (n = 0 to 4, 9)	Receive error	CSR.ORER, CSR.FER, CSR.PER, CSR.DFER, CSR.DPER	CCR0.RIE	Not possible
		FRSR.DR (when FCR.DRES = 1)	CCR0.RIE	Not possible
SCIn_RXI (n = 0 to 4, 9)	Receive data full	CSR.RDRF	CCR0.RIE	Possible
	Receive data ready	FRSR.DR (when FCR.DRES = 0)	CCR0.RIE	Possible
	Address match	CSR.DCMF	CCR0.RIE	Possible
SCIn_TXI (n = 0 to 4, 9)	Transmit data empty	CSR.TDRE	CCR0.TIE	Possible
	TE=0->1 detection			
SCIn_TEI (n = 0 to 4, 9)	Transmit end	CSR.TEND	CCR0.TEIE	Not possible

Note: Only SCI0 supports Manchester mode.

31.12.3 Interrupts in Smart Card Interface Mode

Table 31.45 lists interrupt sources in smart card interface mode. A transmit end interrupt (SCIn_TEI) request and an address match (SCIn_AM) request cannot be used in this mode.

Table 31.45 SCI Interrupt sources in Smart Card Interface Mode

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC or DMAC activation
SCIn_ERI (n = 0 to 4, 9)	Receive error or error signal detection	CSR.ORER, CSR.PER, CSR.ERS	CCR0.RIE	Not possible
SCIn_RXI (n = 0 to 4, 9)	Receive data full	CSR.RDRF	CCR0.RIE	Possible
SCIn_TXI (n = 0 to 4, 9)	Transmit data empty	CSR.TEND	CCR0.TIE	Possible
	When set TE = 0->1			

Data transmission or reception using the DTC or DMAC is also possible in smart card interface mode, similar to normal SCI mode. In transmission, when the CCR0.TEND flag is set to 1, an SCIn_TXI interrupt request is generated. This SCIn_TXI interrupt request activates the DTC or DMAC, allowing transfer of transmit data if the SCIn_TXI request is previously specified as a source of DTC or DMAC activation. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During the retransmission, the TEND flag is kept at 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified

number of bytes, including retransmission after an error occurrence. However, the CSR.ERS flag is not automatically set to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the CCR0.RIE bit to 1 to enable an SCIn_ERI interrupt request to be generated at error occurrence.

When transmitting or receiving data using the DTC or DMAC, always enable the DTC or DMAC before making the SCI settings. For DTC or DMAC settings, see section 17, Data Transfer Controller (DTC), section 16, DMA Controller (DMAC).

In reception, an SCIn_RXI interrupt request is generated when receive data is set to the RDR register. This SCIn_RXI interrupt request activates the DTC or DMAC, allowing transfer of the receive data if the SCIn_RXI request is previously specified as a source of DTC or DMAC activation. If an error occurs, the error flag is set. Therefore, the DTC or DMAC is not activated and an SCIn_ERI interrupt request is issued to the CPU instead. The error flag must be cleared.

31.12.4 Interrupts in Simple IIC Mode

Table 31.46 lists the interrupt sources in simple IIC mode. The STI interrupt is allocated to the transmit end interrupt (SCIn_TEI) request. The receive error interrupt (SCIn_ERI) and the address match (SCIn_AM) request cannot be used.

The DTC or DMAC can also be used to handle transfer in simple IIC mode.

When the ICR.IICINTM bit is 1:

- An SCIn_RXI request is generated on the falling edge of the SCLn signal for the 8th bit. If SCIn_RXI is previously set up as an activation source for the DTC or DMAC, the SCIn_RXI request activates the DTC or DMAC to handle transfer of the received data.
- An SCIn_TXI request is generated on the falling edge of the SCLn signal for the 9th bit (acknowledge bit). If SCIn_TXI is previously set up as an activation source for the DTC or DMAC, the SCIn_TXI request activates the DTC or DMAC to handle transfer of the transmit data.

When the ICR.IICINTM bit is 0:

- An SCIn_RXI request (ACK detection) is generated if the input on the SDAn pin is low on the rising edge of the SCLn signal for the 9th bit (acknowledge bit)
- An SCIn_TXI request (NACK detection) is generated if the input on the SDAn pin is high on the rising edge of the SCLn signal for the 9th bit (acknowledge bit)
- If SCIn_RXI is previously set up as an activation source for the DTC or DMAC, the SCIn_RXI request activates the DTC or DMAC to handle transfer of the received data.

If the DTC or DMAC is used for data transfer in reception or transmission, always set up and enable the DTC or DMAC before setting up the SCI.

When the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits in ICR are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

Table 31.46 SCI interrupt sources in Simple IIC Mode

Name	Interrupt source		Interrupt flag	Interrupt enable	DTC or DMAC activation
	ICR.IICINTM = 1	ICR.IICINTM = 0			
SCIn_RXI (n = 0 to 4, 9)	Reception end	—	—	CCR0.RIE	Possible*1
	—	ACK detection	—		Possible
SCIn_TXI (n = 0 to 4, 9)	Transmission end		—	CCR0.TIE	Possible*1
	—	NACK detection	—		Possible
SCIn_TEI(STIn) (n = 0 to 4, 9)	Completion of generation of a start, restart, or stop condition		ICR.IICSTIF	CCR0.TEIE	Not possible

Note 1. If the DMAC or DTC are being used, you cannot confirm whether ACK or NACK.

31.12.5 Interrupts in Simple LIN mode

Table 31.47 lists interrupt sources in Simple LIN mode.

Table 31.47 SCI interrupt sources in Simple LIN mode

Name	Interrupt Sources	Interrupt Flag	Flag the needs to be confirmed	Interrupt Enable	DTC/DMAC Activation
SCIn_ERI (n = 0, 1)	Receive error	CSR.ORER, CSR.FER, CSR.PER	—	CCR0.RIE	Not Possible
		XSR0.BCDF		XCR0.BCDIE	
		XSR0.COF		CCR0.RIE, XCR0.COFIE	
SCIn_RXI(n = 0, 1)	Receive data full flag	CSR.RDRF	XSR0.CF0MF XSR0.CF1MF XSR0.PIBDF	CCR0.RIE	XSR0.SFSF = 0: Possible XSR0.SFSF = 1: Not Possible
SCIn_AED (n = 0, 1)	Active edge detection	XSR0.AEDF	—	XCR0.AEDIE	Possible
SCIn_TXI(n = 0, 1)	Transmit data empty interrupt	CSR.TDRE	—	CCR0.TIE	Possible
	When set TE = 0->1				
	Break Field output completion	XSR0.BFOF		CCR0.TIE, XCR0.BFOIE	
SCIn_TEI(n = 0, 1)	Transmit end	CSR.TEND	—	CCR0.TEIE	Not Possible
SCIn_BFD(n = 0, 1)	Break Field Detection	XSR0.BFDF	—	XCR0.BFDIE	Not Possible (Unnecessary)

In Simple LIN mode, in addition to reception errors (ORER, FER, PER), an SCIn_ERI interrupt request is output when a bus conflict is detected during transmission, or when a counter overflow of the Simple LIN module occurs. At this time, a SCIn_RXI interrupt request is not output. The SCIn_ERI interrupt request can be canceled by clearing all the flags.

When transmitting Start Frame, if CCR0.TIE = 1 and XCR0.BFOIE = 1, a SCIn_TXI interrupt request is output when Break Field transmission is completed. When Control Field 0 data is written to the TDR register, data transmission starts. Therefore, transmission using DTC or DMAC is possible.

Set CCR0.TEIE = 1 after writing the last transmit data to the TDR register and transmission starts.

During Start Frame reception (XSR0.SFSF = 1), reception using DTC or DMAC by SCIn_RXI interrupt is not possible. Check the CSR register and XSR0 register, check the reception status (See Figure 31.83), and then clear the flag. Also read the RDR register (if you do not need to check the received data value, clear the RDRF flag without reading the RDR register). When reception of Control Field 1 is completed (XSR0.CF1MF = 1), Start Frame detection is disabled (XSR0.SFSF = 0) and reception using DTC or DMAC is possible. Be sure to read the RDR register.

When Start Frame / Break Field detection is enabled (XCR1.SDST = 1), if a Break Field longer than the period set in XCR2.BFLW [15:0] is received, the BFDF flag is set and a SCIn_BFD interrupt request is output. Then SCI becomes the Start Frame reception state. Clear the BFDF flag.

When Start Frame / Break Field detection is enabled (XCR1.SDST = 1) and the bit rate measurement function is enabled (XCR1.BMEN = 1), an SCIn_AED interrupt factor is output when an active edge is detected. Read the timer count capture value (XSR1.TCNT [15:0]).

31.13 Event Linking

By using interrupt request signals as event signals, the SCIn can provide linked operation through the ELC for modules selected in advance.

Event signals can be output regardless of the values of the associated interrupt request enable bits.

(1) Error event output (receive error or error signal detected) (SCIn_ERI, n = 0 to 4, 9)

- Indicates abnormal termination because of a parity error during reception in Asynchronous mode
- Indicates abnormal termination because of a framing error during reception in Asynchronous mode
- Indicates abnormal termination because of an overrun error during reception
- Indicates abnormal termination due to a Manchester error during reception (Only in Manchester mode).

- Indicates that a preface error occurred upon reception and abnormal termination occurred (only in Manchester mode and MCR.PFEREN = 1).
- Indicates that a start bit error occurred during reception and abnormal termination occurred (only in Manchester mode and MCR. SBEREN = 1).
- Indicates that a reception sync error occurred during reception and abnormal termination occurred (only in Manchester mode and only when MCR.SYEREN = 1).
- Indicates detection of the error signal during transmission in smart card interface mode
- The CSR.FER and PER flags are 0, and receive data less than the receive FIFO data trigger number is set in a reception FIFO buffer, and it indicates that 15 etus elapse when FIFO is selected and the FCR.DRES bit is 1
- In Simple LIN mode, indicates that the 16-bit counter in the Simple LIN module has overflowed.
- In Simple LIN mode, a bus collision is detected during transmission (CCR0.TE = 1).

(2) Receive data full event output (SCIn_RXI, n = 0 to 4, 9)

- Indicates that ACK is detected if the ICR.IICINTM bit is 0 in simple IIC mode
- Indicates that the 8th-bit SCLn falling edge is detected if the ICR.IICINTM bit is 1 in simple IIC mode
- When the ICR.IICINTM bit is 1 during master transmission in simple IIC mode, set the ELC so that receive data full events are not used

Non-FIFO selected

- Indicates that received data is set in the Receive Data Register (RDR).

FIFO selected

- Using this event output is prohibited.

(3) Transmit data empty event output (SCIn_TXI, n = 0 to 4, 9)

- Indicates that the CCR0.TE bit is changed from 0 to 1
- Indicates that transmission is complete in smart card interface mode
- Indicates that NACK is detected if the ICR.IICINTM bit is 0 in simple IIC mode
- Indicates that the 9th-bit SCLn falling edge is detected if the ICR.IICINTM bit is 1 in simple IIC mode
- In Simple LIN mode, indicates that Break Field output is complete.

Non-FIFO selected

- Indicates that transmit data is transferred from the Transmit Data Register (TDR) to the Transmit Shift Register (TSR).

FIFO selected

- Using this event output is prohibited.

(4) Transmit end event output (SCIn_TEI, n = 0 to 4, 9)

- Indicates the completion of transmission
- Indicates that the starting condition, resumption condition, or termination condition is generated in simple IIC mode
- In Smart Card mode, the transmit end event is not output.

Note: When FIFO is selected, using this event output is prohibited

(5) Address match event output (SCIn_AM, n = 0 to 4, 9)

- Indicates a match of the comparison data (CCR4.CMPD) with one frame of receive data when CCR0.DCME is set to 1 in Asynchronous mode, including multi-processor mode.

(6) Active edge detection event output (SCIn_AED, n= 0, 1)

- In Simple LIN mode, when CCR1.BMEN is 1, it indicates that a valid edge has been detected in the RXD input signal.

31.14 Noise Cancellation Function

Figure 31.113 shows the configuration of the noise filter used for noise cancellation. The noise filter consists of a 2-stage flip-flop circuit and a match detection circuit. When the input signals of the noise filter and the output signals of the 2-stage flip-flop circuits completely match, the matched level is conveyed as an internal signal. Unless otherwise matched, the previous value is retained. When the same level is retained for 3 cycles or longer on the sampling clock of the noise filter, it is considered as a valid receive signal. A change in pulse for 3 cycles or shorter is considered as noise, not as a receive signal. (When CCR1.NFM = 0)

When CCR2.ABCSE2 = 1 in asynchronous mode, the above noise cancellation cannot be used because one bit has only 4 base clocks. Therefore, only in this case, set the noise cancellation mode to transmit the majority value internally by a majority vote of 3 sampling values instead of matching 3 consecutive sampling values. (CCR1.NFM = 1)

In Asynchronous mode, Manchester and Simple LIN modes, the noise cancellation function can be applied to the receive signal input to the RXDn pin. The sampling period of the noise filter can be selected from the base clock period and the divided clock of the baud rate generator clock source by CCR1.NFCS[2:0].

- When CCR1.NFCS[2:0] = 000b, CCR2.ABCS = 0 and CCR2.ABCSE = 0, the cycle is 1/16 of a 1-bit period.
- When CCR1.NFCS[2:0] = 000b, CCR2.ABCS = 1 and CCR2.ABCSE = 0, the cycle is 1/8 of a 1-bit period.
- When CCR1.NFCS[2:0] = 000b, CCR2.ABCSE = 1, the cycle is 1/6 of a 1-bit period.
- When CCR1.NFCS[2:0] = 000b, CCR2.ABCSE2 = 1, the cycle is 1/4 of a period 1 transfer bit.
- For the CCR1.NFCS value, select “000b” or “001b” if ABCSE = 1 or ABCSE2 = 1. At least, the sampling period of Noise Filter should not be greater than the base clock due to the CCR2.BRR setting value.

In simple IIC mode, this function can be used for each input on SDAn and SCLn. The sampling clock is selected from divided clock of baud rate generator settings by CCR1.NFCS[2:0].

If the base clock is stopped once with the noise filter enabled and then the base clock input is restarted again, the noise filter operation resumes from the state where the clock was stopped. When CCR0.TE and CCR0.RE are set to 0 during base clock input, all of the noise filter flip-flop values are initialized to 1. Accordingly, if the input data is 1 when reception operation resumes, the function determines that a level match is detected and the result is conveyed as an internal signal. If the input data is 0, the Noise Filter output holds its initial value until all three sampled pin levels match (CCR1.NFM = 0) or two or more match (CCR1.NFM = 1).

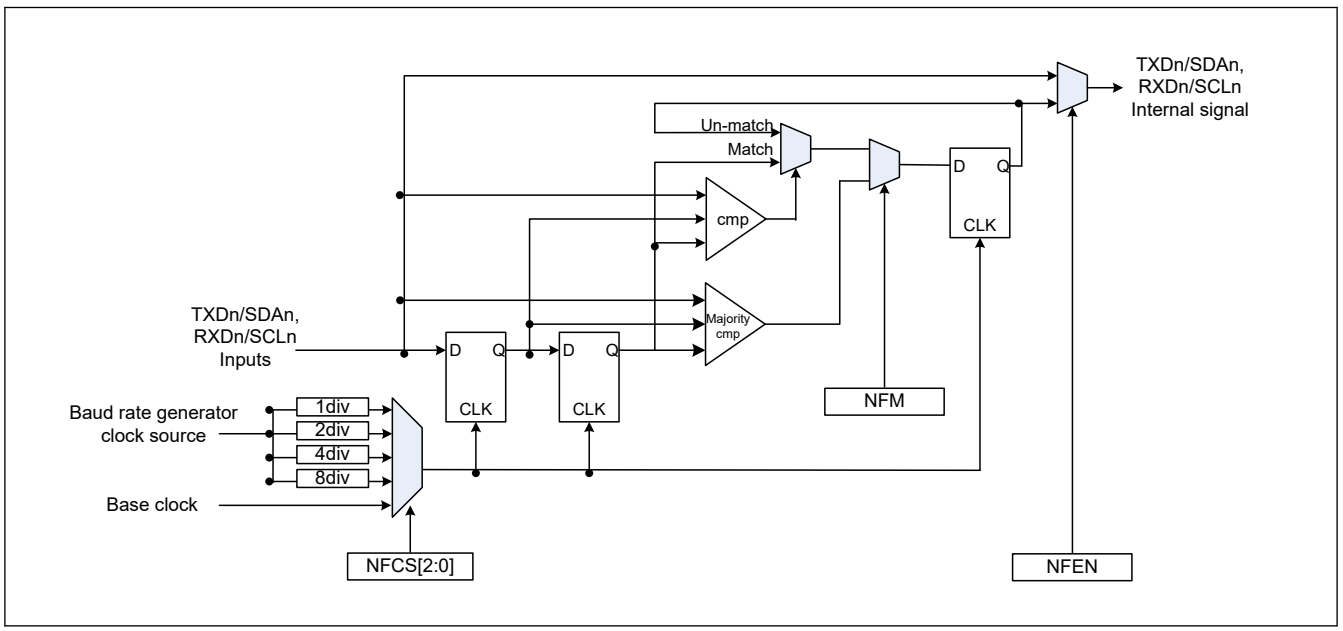


Figure 31.113 Digital noise filter circuit block diagram

31.15 RS-485 Driver Control Function

Setting the DEN bit in the SCI common control register3 (CCR3) to 1 enables the RS-485 driver control function and generates a DEn (Driver Enable) signal that enables the external transceiver transmission mode. The DEn signal outputs a valid level for the period with driver assertion time and driver negate time added before and after data transmission. The DEn signal valid level is set by the DEPOL bit in the driver control register (DCR).

The driver assertion time is the time from when the DEn signal is valid until the start bit starts. Set by DEAST [4:0] of driver control register (DCR).

The driver negate time is the time from the end of the last stop bit of the transmitted message to the invalidation of the DEn signal. Set with DENGT [4:0] of the driver control register (DCR).

DEAST and DENGT are expressed in base clock period (1/4, 1/6, 1/8, or 1/16 bit period, see Table 31.6). For details, see section 31.2.13. DCR : Driver Control Register.

When this function is used (CCR3.DEN = 1), the CSR.TEND set timing and SCIn_TEI interrupt output timing are at the end of the driver negation time.

When transmission is completed and the next transmission data is not written before the DEn signal is negated, the DEn signal is negated once. If the timing for writing the next transmit data is not in time, assert the DEn signal after negating it again, insert the driver assertion time, and transmit the next data. If you want to perform the next transmission with the DEn signal asserted, write the next transmission data to the TDR quickly enough in consideration of the synchronization delay time of the register.

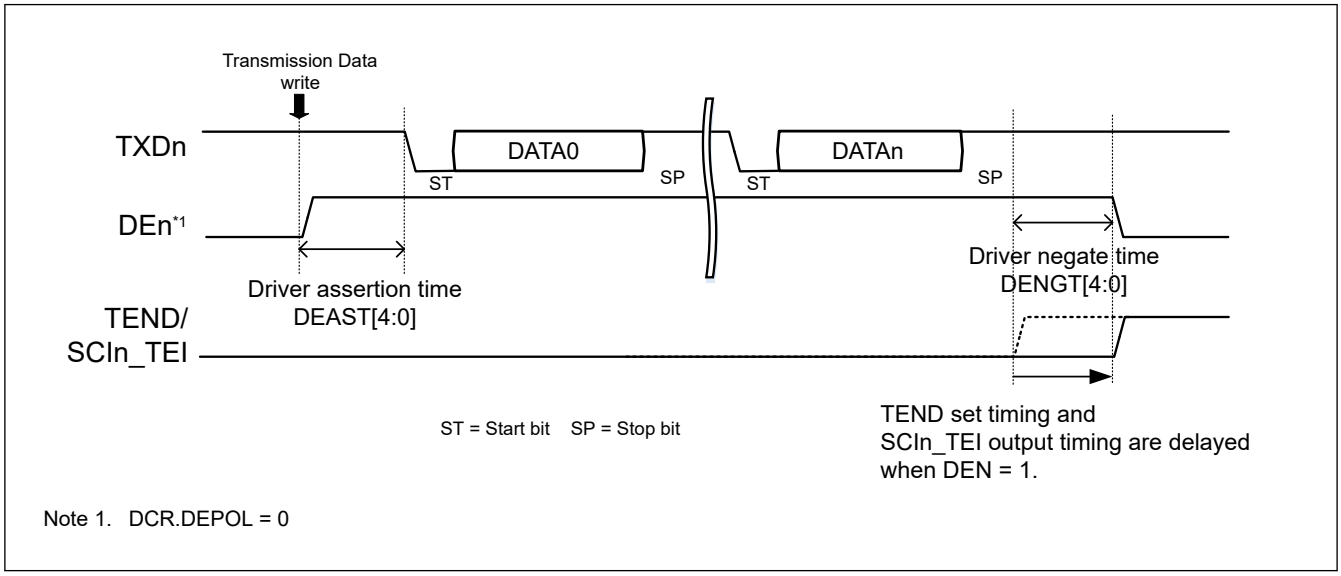


Figure 31.114 The image waveform for RS-485 driver control DE signal output

31.16 Loopback Function

The loopback function can be used in Asynchronous mode with the internal clock, and Manchester mode with the internal clock, and Clock synchronous mode with the internal clock.

When 1 is written to the SPLP bit in the CCR1 register, SCI blocks the external input (RXDn) path and connects the output path of the transmit data register and the input path of the receive data register.

When this function is used with TINV bit = 1, inversion of transmission data becomes reception data. However, this function can be used with TINV = 1 only when operating in clock synchronous mode internal clock.

Table 31.48 shows the relationship between the TINV and SPLP bit settings and the received data.

Table 31.48 TINV and SPLP bit settings and received data

CCR1.TINV	CCR1.SPLP	Receive Data	Communication mode		
			Asynchronous	Manchester	Clock synchronous
			internal clock	internal clock	internal clock
—	0	Receive Data from RXDn pin	Possible	Possible	Possible
0	1	Transmit Data	Possible	Possible	Possible
1	1	Inverted transmit data	Impossible	Impossible	Possible

Note: —: don't care

Figure 31.115 shows the configuration of the shift register input / output path in loopback mode.

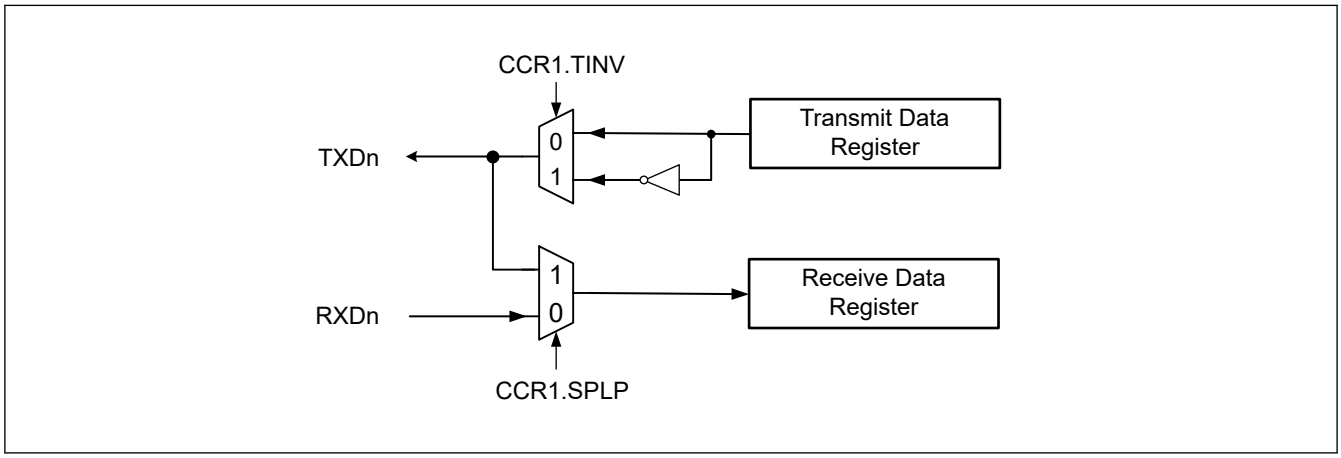


Figure 31.115 Shift register input output configuration image in loopback mode

31.17 Half-Duplex communication Function

Do not use the half-duplex communication function in Simple IIC, Simple SPI and Smart Card Interface modes.

In other communication modes, if the CCR1.SHARPS bit is set to 1, half-duplex communication using the TXDn pin is possible. When half-duplex communication is used, transmission and reception must be performed exclusively. Transmission and reception settings (CCR0.TE = 1 and CCR0.RE = 1) is prohibited.

However, if half-duplex communication is performed as the master reception in clock synchronous mode, perform transmission / reception settings (CCR0.TE = 1 and CCR0.RE = 1) and perform dummy transmission. By dummy transmission (arbitrary transmission data is written to TDR), SCKn is output and reception is enabled. The dummy transmission data is discarded inside the SCI and is not actually transmitted.

During half-duplex communication, only the TXDn pin is used as the communication pin. Output when CCR0.TE = 1, input when CCR0.TE = 0.

31.18 Synchronizer Bypass Function

The SCI has a bus clock (PCLK) and the operation clock (TCLK). And these have each operating circuit. Therefore, there is a synchronization circuit for signal transfer between different clocks, and synchronization delay time is required for signal propagation between different clocks.

However, the synchronization circuit can be bypassed by the CCR3.BPEN bit only when the same clock is input to the bus clock and the operation clock. In this case, eliminates synchronization delay time and improves responsiveness. Figure 31.116 shows the image waveform of the bypass function.

The SCI also has a synchronization circuit between the communication clock (SCKn) and the operation clock (TCLK), but this synchronization circuit cannot be bypassed.

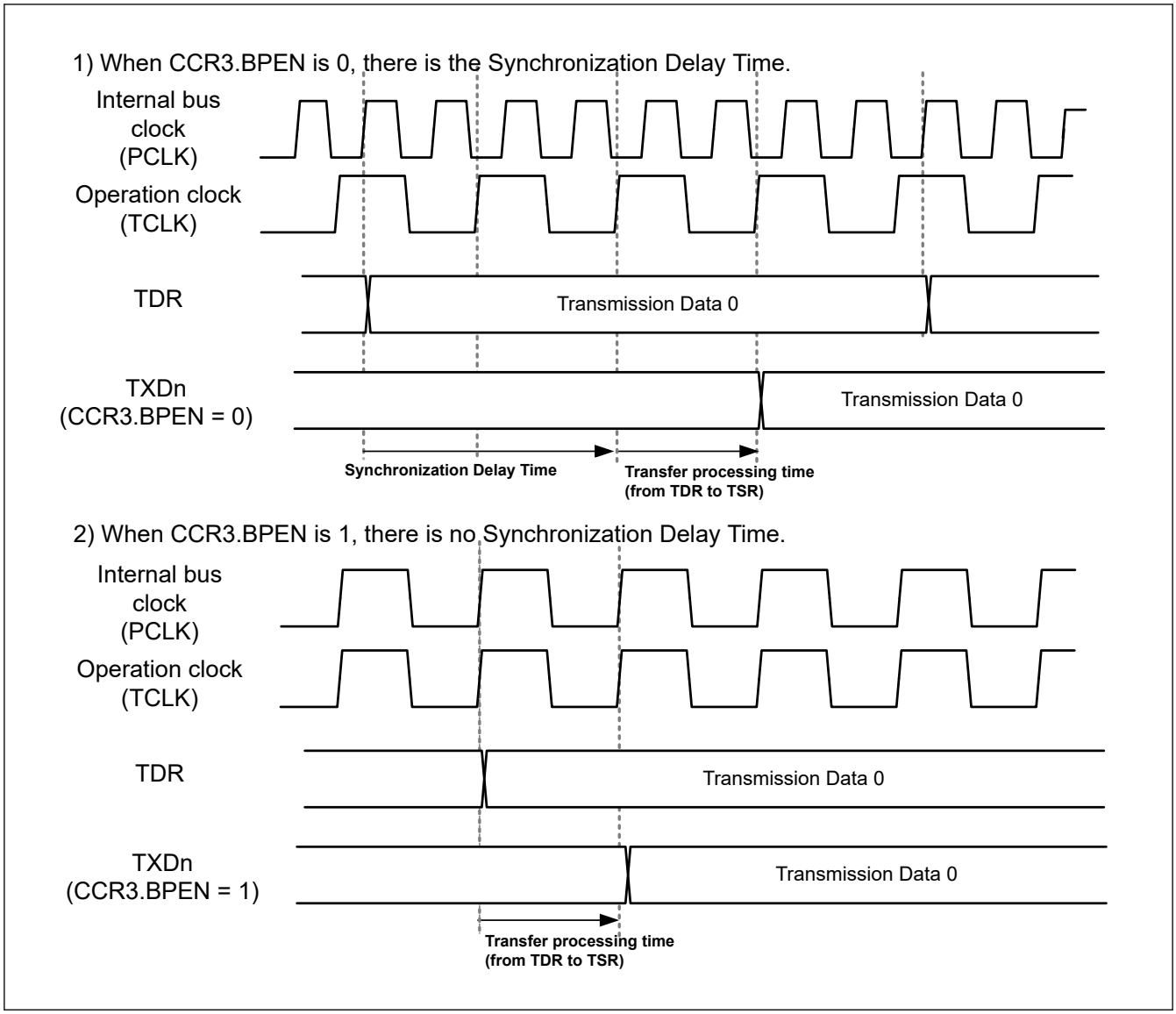


Figure 31.116 Image waveform of Synchronizer bypass function

31.19 Usage Notes

31.19.1 Settings for the Module-Stop Function

The Module Stop Control Register B (MSTPCRB) can enable or disable SCI operation. The SCI is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

31.19.2 SCI Operation during Low Power State

(1) Transmission

Before using the power consumption reduction function to reduce SCI's power consumption, do the following to confirm transmission end (CSR.TEND = 1):

- Set the output pin state after transmission operation is stopped by CCR1.SPB2DT, SPB2IO.
- Stop the transmission (CCR0.TIE = 0, TE = 0, TEIE = 0)

When transitions to these states are made during transmission, the data being transmitted become indeterminate.

To transmit data in the same operation mode after cancellation of the low power consumption state, set the TE bit to 1, read CSR, and write data to TDR sequentially to start data transmission. To transmit data with a different operation mode, initialize the SCI first.

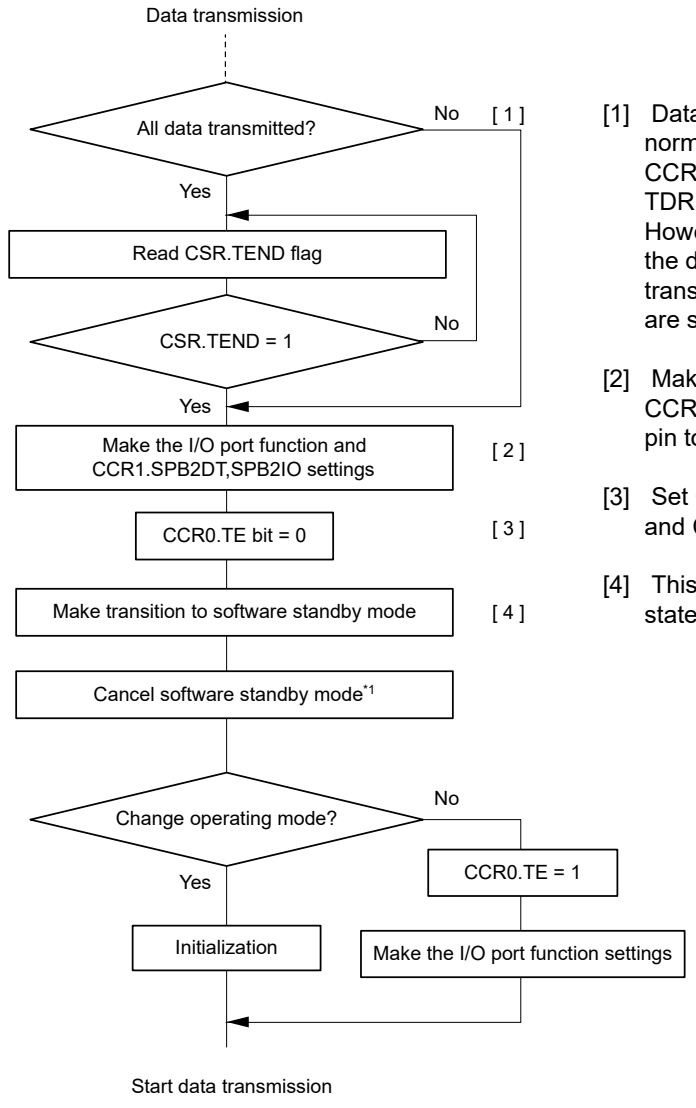
To start transmission using the DMAC or DTC after cancellation from software standby mode, set the CCR0.TE and CCR0.TIE bit to at the same time. Then SCIn_TXI interrupt flag is generated, which causes the DMAC or DTC to write the transmit data, which starts transmission.

Figure 31.117 shows a sample flowchart for transition to software standby mode during transmission. Figure 31.118 and Figure 31.119 show the port pin states during transition to software standby mode.

(2) Reception

Before specifying the module stop state or making a transition to software standby mode, stop the receive operations (CCR0.RE = 0). If transition is made during data reception, the data being received will be invalid.

Figure 31.120 shows a sample flowchart for reception to software standby mode during reception.



[1] Data being transmitted is lost halfway. Data can be normally transmitted from the CPU by setting the CCR0.TE to “1”, reading CSR, and writing data to TDR after canceling software stand by mode. However, if the DMAC or DTC has been activated, the data remaining in the DMAC or DTC will be transmitted when both the TE and TIE bits in CCR0 are set to “1”.

[2] Make the I/O port function and CCR1.SPB2DT,SPB2IO settings to switch the TxDn pin to operate as a general I/O port.

[3] Set CCR0.TE bit to “0”. Set also 0 to CCR0.TIE and CCR0.TEIE if you set these bits were 1.

[4] This includes the setting for the module stopped state.

Note 1. The registers of channels other than SCI0 are initialized in Software Standby mode. So they need to be set again

Figure 31.117 Example of Flowchart for Transition to Software Standby Mode during Transmission

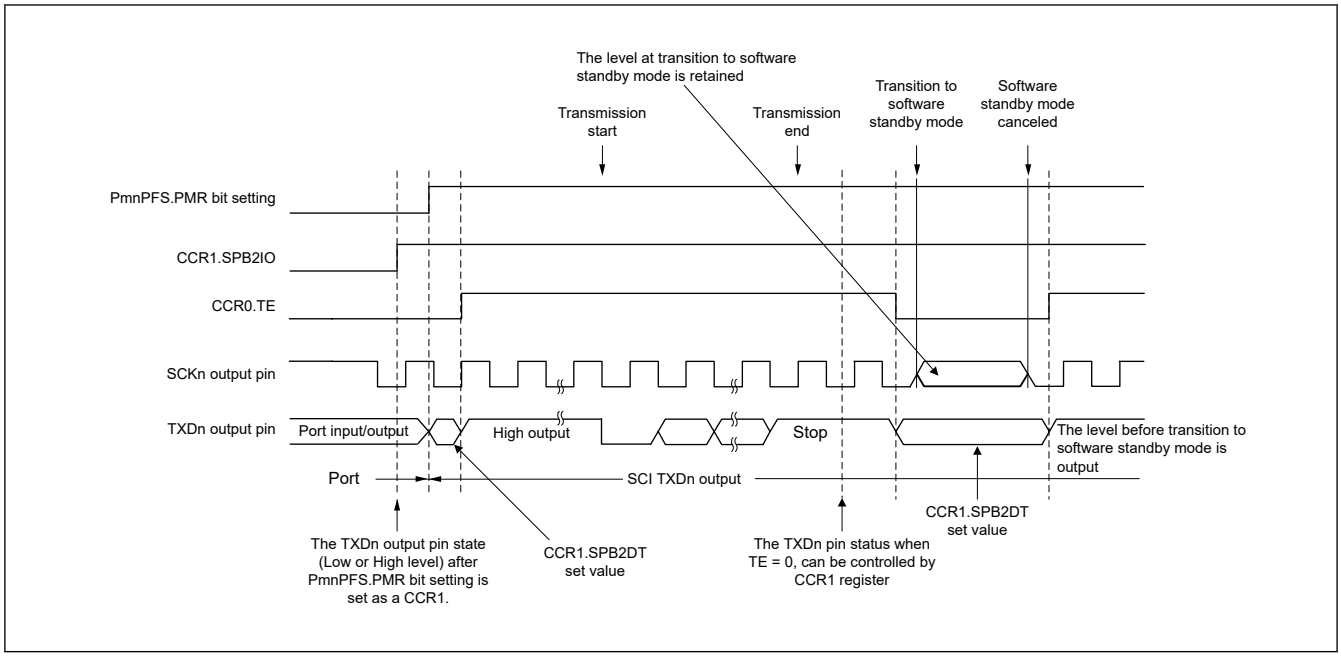


Figure 31.118 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)

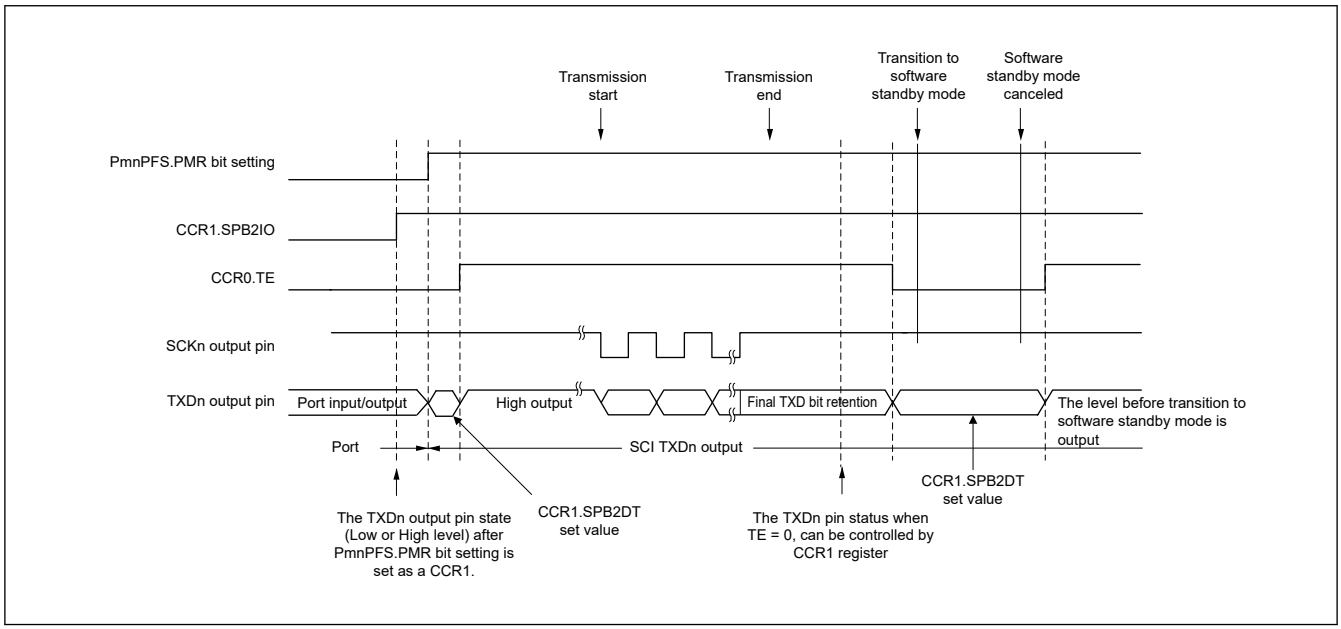
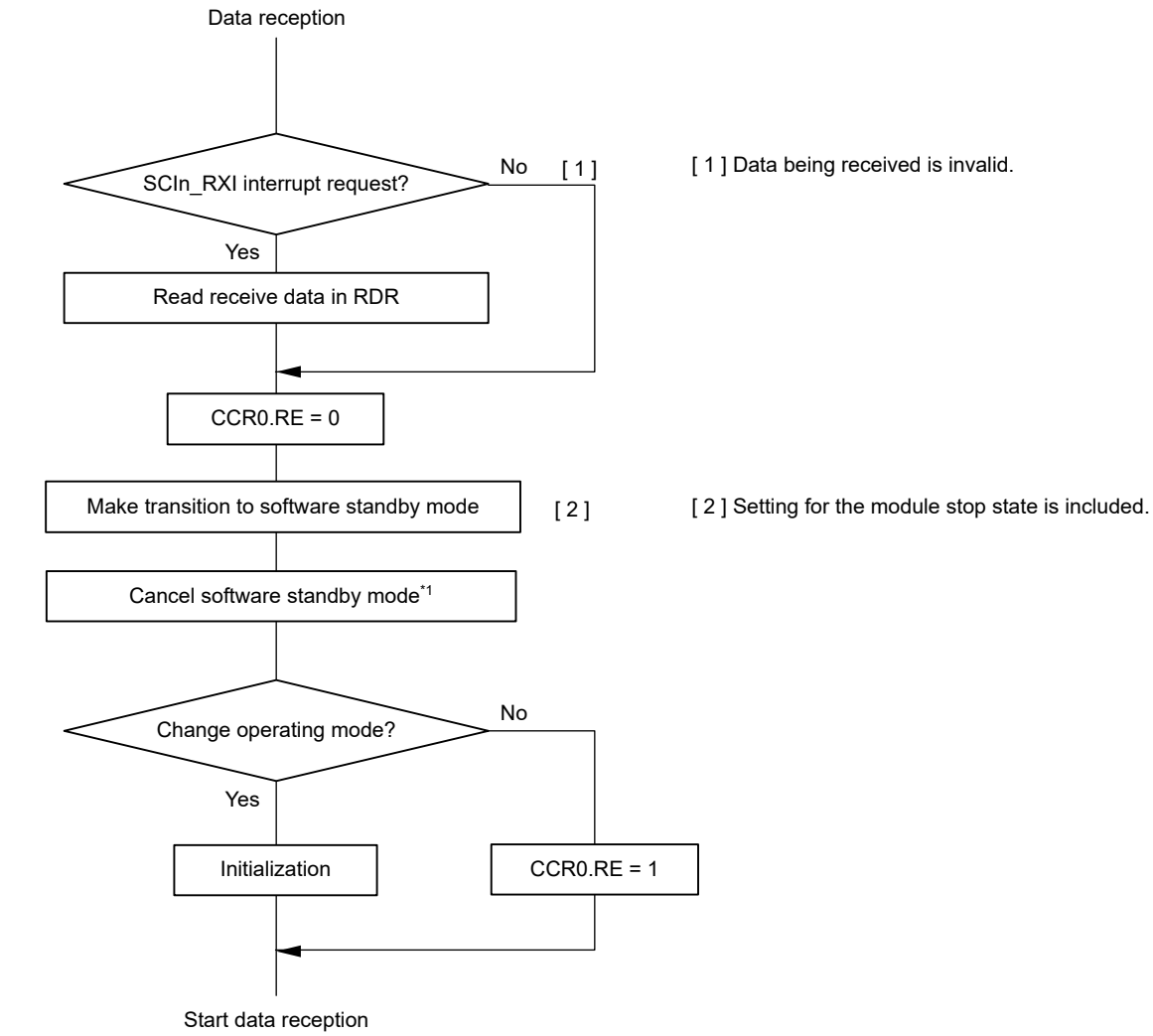


Figure 31.119 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)



Note 1. The registers of channels other than SCI0 are initialized in Software Standby mode. So they need to be set again

Figure 31.120 Example of Flowchart for Reception to Software Standby Mode during Reception

31.19.3 Break Detection and Processing

(1) Non-FIFO selected

When a framing error is detected, a break can be detected by reading CSR.RXDMON bit value. In a break, the input from the RXDn pin becomes all 0s, and the CSR.FER flag is set to 1 to indicate a framing error, and the CSR.PER flag might also be set to 1 to indicate a parity error. The SCI continues the receive operation even after a break is received. Therefore, even if the FER flag is 0, indicating that no framing error occurred, it is set to 1 again. When the CCR3.RXDESEL bit is 1, the SCI sets the CSR.FER flag to 1 and stops receiving operations until a start bit of the next data frame is detected. If the CSR.FER flag is set to 0, the CSR.FER flag retains 0 during the break.

When the RXDn pin is set to 1 and the break ends, detecting the beginning of the start bit on the first falling edge of the RXDn pin allows the SCI to start the receiving operation.

(2) FIFO selected

After a framing error is detected and when the SCI detects that continuous receive data is 0 for 1 frame, reception stops. When a framing error is detected, a break can be detected by reading the CSR.RXDMON flag value. After the RXDn signal is in high and the break is finished, data reception to the receive-FIFO (RDR) register resumes.

31.19.4 Mark State and Production of Breaks

When the CCR0.TE bit is 0, disabling serial transmission, the state of the TXDn pin can be set using the CCR1.SPB2IO and CCR1.SPB2DT bits. With this approach, a TXDn pin can be placed in the mark state to transmit a break.

Before setting the CCR0.TE bit to 1, enabling serial transmission, set the SPB2IO and SPB2DT bits to put the communication line in the mark state (the state of 1), and change the TXDn pin using I/O port function. To output a break on data transmission, after setting the TXDn pin to output 0 by setting the SPB2IO and SPB2DT bits, change the TXDn pin using the I/O port function and set the CCR0.TE bit to 0. When the CCR0.TE bit is set to 0, the transmitter is initialized regardless of the current state of transmission.

31.19.5 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode)

Transmission can be start by writing transmit-data to TDR even if CSR.ORER is 1. However, reception cannot be started. Note also that the receive error flags cannot be set to 0 even if the CCR0.RE is set to 0 (serial reception is disabled).

31.19.6 Writing Data to TDR

(1) Non-FIFO selected

Data can be written to TDR anytime when CCR0.TE is 1. However, if new data is written to TDR when transmit data is remaining in TDR, the previous data in TDR is lost because it has not been transferred to TSR yet. If you use DTC or DMAC, be sure to write transmit data to TDR in the SCIn_TXI interrupt request handling routine.

(2) FIFO selected

Data can be written to transmit-FIFO(TDR) when CCR0.TE is 1. Check the number of writable data with the FTSR.T [5:0] bit.

31.19.7 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode)

When the external clock source is used as a synchronization clock, the following restrictions apply.

(1) Start of transmission

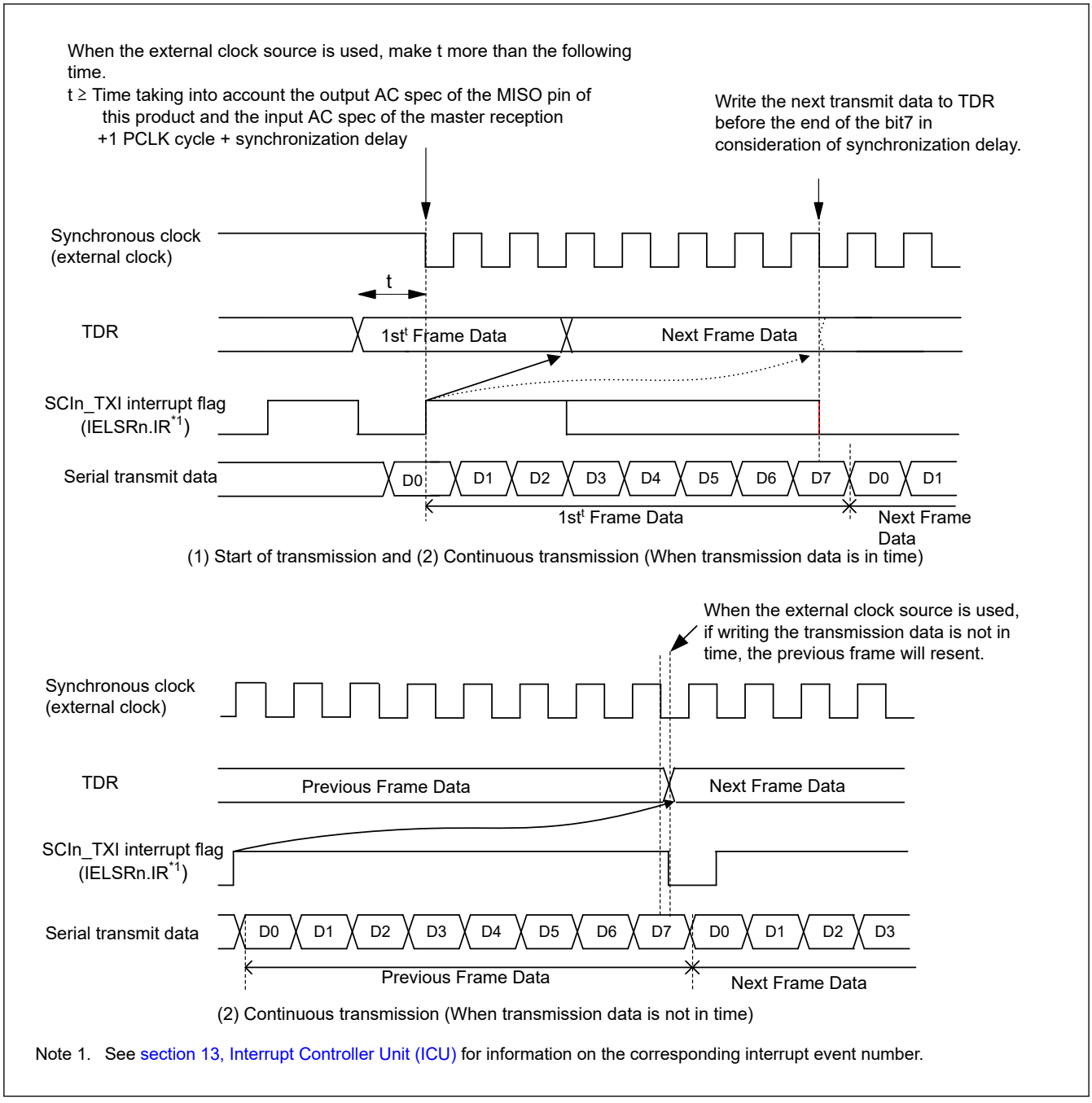
Update TDR by the CPU, DMAC, or DTC and wait at least the following time until the start of the external clock input: (See Figure 31.121)

Take the following time into account: the output AC spec of the MISO pin of this product and the input AC spec of the master reception + 1 PCLK cycle + synchronization delay.

(2) Continuous transmission

Write the next transmit data to TDR before the falling edge[*1] of the transmit clock for bit 7. Write the transmit data to TDR in consideration of synchronization delay. If the transmit data cannot be written in time, the previous frame data is resent. (See Figure 31.121)

Note 1. When CCR3.CPOL = 1 and CCR3.CPHA = 0, or CCR3.CPOL = 0 and CCR3.CPHA = 1. In the case of CCR3.CPOL = 0 and CCR3.CPHA = 0, or CCR3.CPOL = 1 and CCR3.CPHA = 1, it's the rising edge.



Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 31.121 Restrictions on Use of External Clock in Clock Synchronous Transmission

31.19.8 Restrictions on Using DMAC or DTC

When using the DMAC or DTC to read RDR, be sure to set the receive data full interrupt (SCIn_RXI) as the activation source of the relevant SCI.

During the operation in transmission / reception using the DMAC or DTC, it should not set transfer information of DMAC or DTC.

31.19.9 Notes on Starting Transfer

At the point where transfer starts when the interrupt status flag (IELSRn.IR flag) in the ICU is 1, follow the procedure in this section to clear interrupt requests before permitting operations (by setting the CCR0.TE or CCR0.RE bit to 1). For details on the interrupt status flag, see section 13, Interrupt Controller Unit (ICU).

1. Confirm that transfer has stopped (the CCR0.TE or CCR0.RE bit is 0)

2. Set the associated interrupt enable bit (CCR0.TIE or CCR0.RIE bit) to 0
3. Read the associated interrupt enable bit (CCR0.TIE or CCR0.RIE bit) to check that it actually becomes 0
4. Set the interrupt status flag, IELSRn.IR, in the ICU to 0

31.19.10 Limitations on Simple SPI Mode

(1) Master mode

- Use a resistor to pull up or pull down the clock line matching the initial settings for the transfer clock set in the CCR3.CPHA and CPOL bits when the CCR0.SSE bit is 1.

  This prevents the clock line from being placed in the high-impedance state when the CCR0.TE bit is set to 0 or unexpected edges from being generated on the clock line when the CCR0.TE bit changes from 0 to 1. When the CCR0.SSE bit is 0 in single master mode, pulling up or pulling down the clock line is not required because the clock line is not placed in the high-impedance state even when the SCR.TE bit is set to 0.
- For the clock delay setting (CCR3.CPHA bit is 0), the receive data full interrupt (SCIn_RXI) is generated before the final clock edge on the SCKn pin as indicated in Figure 31.122. If the TE and RE bits in the CCR0 register become 0 before the final edge of the clock signal on the SCKn pin, the SCKn pin is placed in the high-impedance state, so the width of the last clock pulse of the transfer clock is shortened. Additionally, an SCIn_RXI interrupt might lead to the input signal on the SSn pin of a connected slave going to the high level before the final edge of the clock signal on the SCKn pin, leading to incorrect operation of the slave.
- In a multi-master configuration, the SCKn pin output goes to high-impedance while the input on the SSn pin is at the low level if a mode fault error occurs while a character is being transferred, stopping supply of the clock signal to the connected slave. Reset the connected slave to avoid misaligned bits when transfer is restarted.

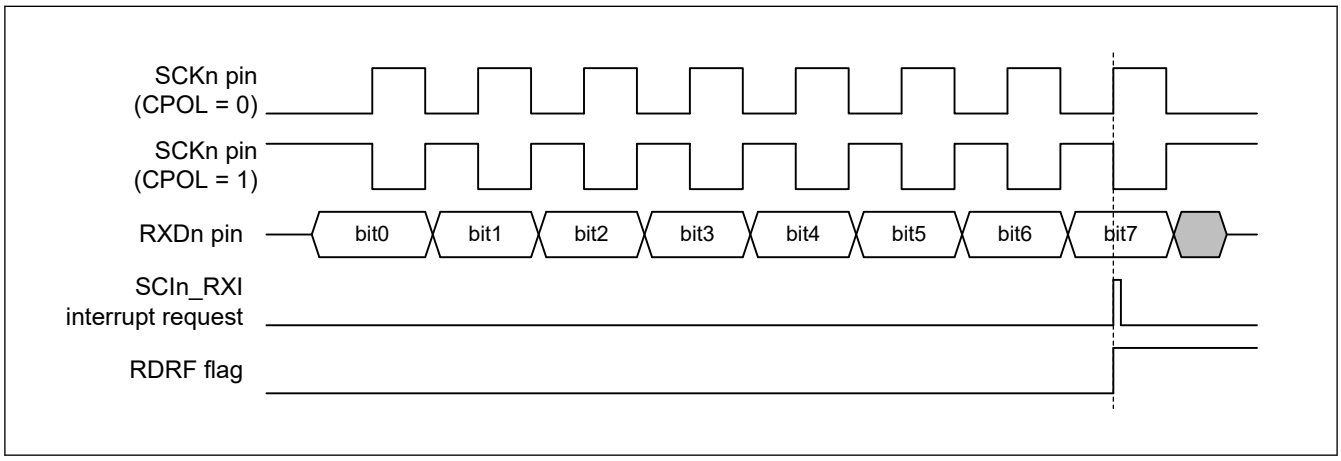


Figure 31.122 Timing of SCIn_RXI interrupt in simple SPI mode with clock delay

(2) Slave mode

- It takes 1PCLK + synchronization delay time + data output delay time (AC spec) from writing the transmit data to the TDR register until the data is output to the RXDn pin. Take these into account when starting external clock input.
- Provide an external clock signal to the master the same as the data length for transfer
- Secure the SSn input setup time (AC spec) from the SSn low-level input to the start of external clock input.
- Control the input on the SSn pin before the start and after the end of data transfer
- When the input level on the SSn pin is to be changed from low to high while a character is being transferred, set the TE and RE bits in the CCR0 register to 0 and, after restoring the settings, restart transfer of the first byte

31.19.11 Notes on Transmit Enable bit (CCR0.TE)

In initial register value, when CCR0.TE = 0, the state of the TXDn pin is high impedance. The TXDn line should not be high impedance by the following one of ways.

1. The pull-up or pull-down resistance is connected to the TXDn line.

2. Set CCR1 and decided level of TXDn terminal during TE is 0.

31.19.12 Notes on Simple LIN mode

In Simple LIN mode (CCR3.MOD[2:0] = 110), the following functions cannot be used.

- Multi-processor communication function
- Bit Rate Modulation function
- Loopback function
- FIFO buffer

31.19.13 Notes on RS-485 Driver Control function

RS-485 Driver control function is valid only in Asynchronous mode.

When RS-485 Driver control function is active (CCR3.DEN = 1), the CSR.TEND set timing / SCIn_TEI output timing changes as follows. Wait for the SCIn_TEI interrupt and set the CCR0.TE bit in SCI to 0.

When RS-485 Driver control function is inactive: When STOP bit output is completed.

When RS-485 Driver control function is active: At the end of DEn negation time.

31.19.14 Notes on Loopback function

The Loopback function is valid in Asynchronous mode with internal clock, in Manchester mode with internal clock and Clock synchronous mode with internal clock.

31.19.15 Notes regarding register access when operation clock (TCLK) is slower than bus clock (PCLK)

If the operating clock (TCLK) is slower than the bus clock (PCLK), the time until this information is transmitted internally after writing to the CCR0.TE and CCR0.RE registers is slower than the bus access time. In particular, when trying to change the setting register after writing 0 and interrupting communication, do not change the setting register before the signal inside the SCI is in the communication stopped state. To prevent this, after setting CCR0.TE and CCR0.RE to 0, check the CESR.TIST and CESR.RIST bits until they are 0 before setting the next register.

31.19.16 Notes on interrupting operation

If 0 is written to CCR0.RE during data reception and the reception operation is interrupted, there is a possibility of an invalid state, do not use the received data (RDR register stored value) and the flag value of each status register. To interrupt the reception operation, stop the interrupt or event link reception side and then write 0 to the CCR0.RE bit.

31.19.17 Notes on CCR3.BPEN bit setting

Set the BPEN bit only once when setting the CCR3 register in the SCI initialization flow.

This bit cannot be changed after the initialization.

When this bit setting is changed, start from the SCI initialization flow again.

32. I2C Bus Interface (IIC)

32.1 Overview

The I2C bus interface (IIC) has 2 channels. The IIC module conforms with and provides a subset of the NXP I2C (Inter-Integrated Circuit) bus interface functions.

Table 32.1 lists the IIC specifications, Figure 32.1 shows a block diagram, and Figure 32.2 shows an example of I/O pin connections to external circuits, with an I2C bus configuration. Table 32.2 lists the I/O pins.

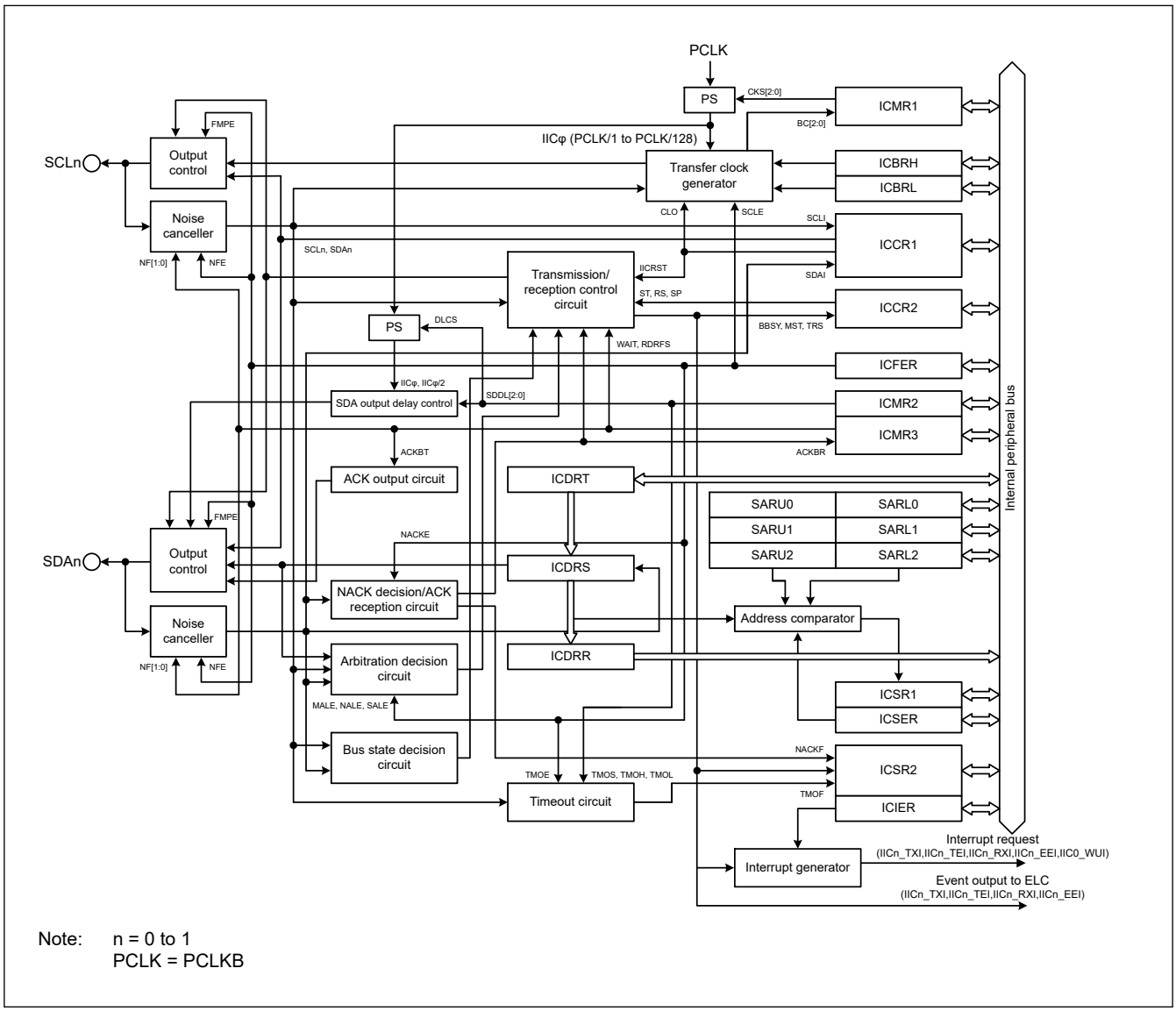
Table 32.1 IIC specifications (1 of 2)

Parameter	Specifications
Communications format	• I2C-bus format or SMBus format • Master or slave mode selectable • Automatic securing of the setup times, hold times, and bus-free times for the transfer rate
Transfer rate	• Fast-mode Plus supported, up to 1 Mbps
SCL clock	For master operation, the duty cycle of the SCL clock is selectable in the range from 4% to 96%
Issuing and detecting conditions	• Start, restart, and stop conditions are automatically generated • Start conditions (including restart conditions) and stop conditions are detectable
Slave address	• Configurable for up to three different slave addresses • 7- and 10-bit address formats supported, including simultaneous use • General call addresses, device ID addresses, and SMBus host addresses detectable
Acknowledgment	• For transmission, automatic loading of the acknowledge bit Transfer of the next transmit data can be automatically suspended on detection of a not-acknowledge bit. • For reception, automatic transmission of the acknowledge bit If a wait between the 8th and 9th clock cycles is selected, the software can control the value in the acknowledge field in response to the received value.
Wait function	During reception, the following wait periods are available by holding the SCL clock low: • Waiting between the eighth and ninth clock cycles • Waiting between the ninth clock cycle and the 1st clock cycle of the next transfer
SDA output delay function	Output timing of transmitted data, including the acknowledge bit, can be delayed
Arbitration	• For multi-master operation: – SCL clock synchronization is possible when conflict occurs with the SCL signal from another master – When issuing the start condition creates conflict on the bus, loss of arbitration is detected by testing for a mismatch between the internal signal for the SDA line and the level on the SDA line – In master operation, loss of arbitration is detected by testing for non-matching between the signal on the SDA line and the internal signal for the SDA line • Loss of arbitration because the start condition occurs while the bus is busy is detectable, to prevent the issuing of double start conditions • Loss of arbitration is detectable on transfer of a not-acknowledge bit because the internal signal for the SDA line and the level on the SDA line do not match • Loss of arbitration because a mismatch of internal and line levels for data is detectable in slave transmission
Timeout function	Internal detection of long-interval stops of the SCL clock
Noise cancellation	• Digital noise filters for both the SCL and SDA signals • Programmable window for noise cancellation by the filters
Interrupt sources	• Transfer error or event occurrence (arbitration-lost, NACK, timeout, start or restart condition, or stop condition) • Receive data full, including matching with a slave address • Transmit data empty, including matching with a slave address • Transmit end
Module-stop function	Module-stop state can be set for each channel to reduce power consumption.
IIC operating modes	• Master transmit • Master receive • Slave transmit • Slave receive

Table 32.1 IIC specifications (2 of 2)

Parameter	Specifications
Event link function (output)	• Transfer error or event occurrence (arbitration-lost, NACK, timeout, start or restart condition, or stop condition) • Receive data full, including matching with a slave address • Transmit data empty, including matching with a slave address • Transmit end
Wakeup function*1	CPU can return from Software Standby mode using a wakeup event
TrustZone Filter	Security and Privilege attribution can be set for each channels

Note 1. This function is only available for IIC channel IIC0. IIC1 is not supported.



Note: n = 0 to 1
PCLK = PCLKB

Figure 32.1 IIC block diagram

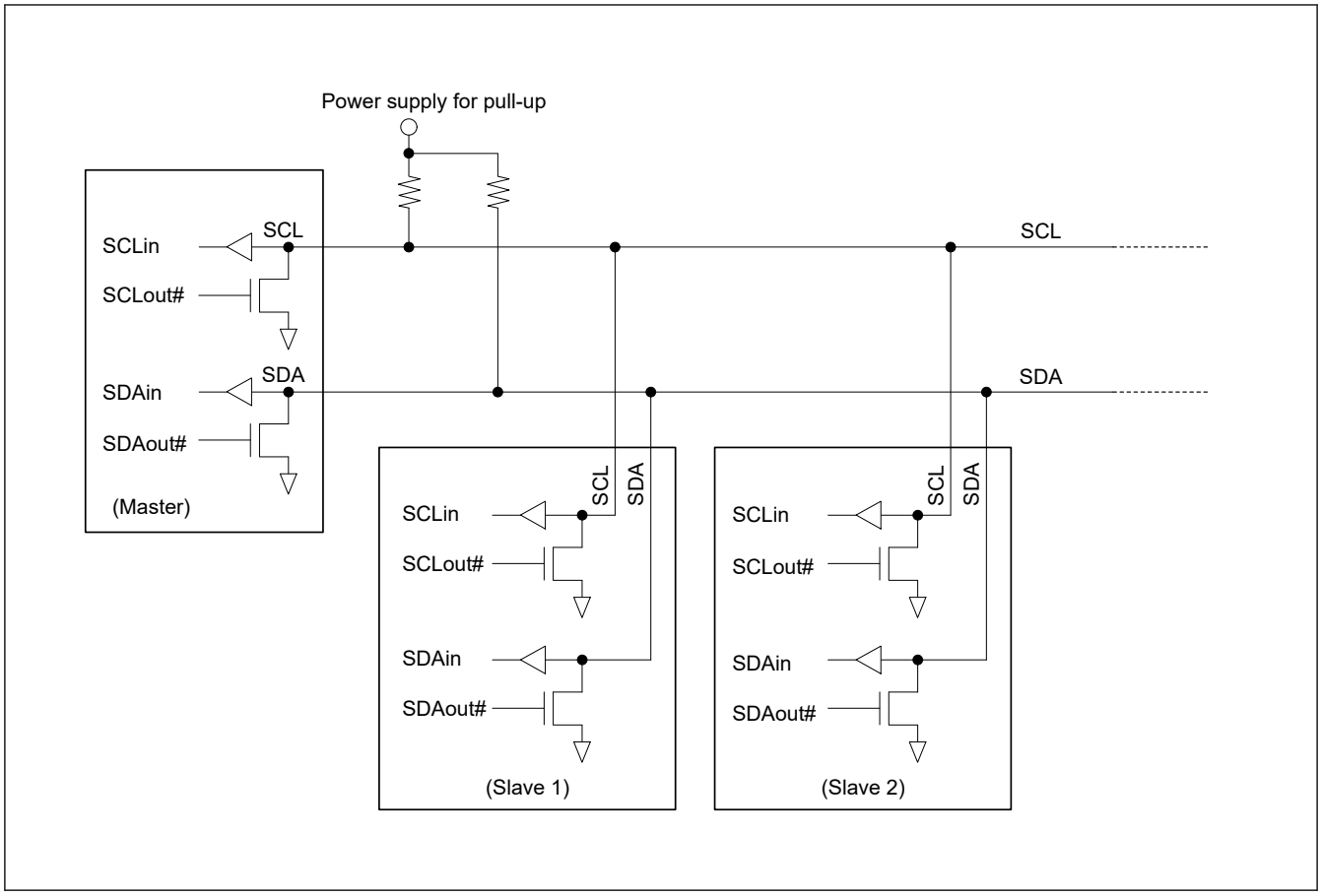


Figure 32.2 I/O pin connection to an external circuit (I^2C bus configuration example)

The input level of the signals for IIC is CMOS when I^2C bus is selected (ICMR3.SMBS = 0), or TTL when SMBus is selected (ICMR3.SMBS = 1).

Table 32.2 IIC I/O pins

Channel	Pin name	I/O	Function
IICn	SCLn	I/O	IICn serial clock I/O pin
	SDAn	I/O	IICn serial data I/O pin

Note: n = 0 to 1

32.2 Register Descriptions

32.2.1 ICCR1 : I^2C Bus Control Register 1

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 1)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Function	R/W
0	SDAI	SDA Line Monitor 0: SDAn line is low 1: SDAn line is high	R

Bit	Symbol	Function	R/W
1	SCLI	SCL Line Monitor 0: SCLn line is low 1: SCLn line is high	R
2	SDAO	SDA Output Control/Monitor 0: Read: IIC drives SDAn pin low Write: IIC drives SDAn pin low 1: Read: IIC releases SDAn pin Write: IIC releases SDAn pin	R/W
3	SCLO	SCL Output Control/Monitor Use an external pull-up resistor to drive the signal high. 0: Read: IIC drives SCLn pin low Write: IIC drives SCLn pin low 1: Read: IIC releases SCLn pin Write: IIC releases SCLn pin	R/W
4	SOWP	SCLO/SDAO Write Protect This bit is read as 1. 0: Write enable SCLO and SDAO bits 1: Write protect SCLO and SDAO bits	W
5	CLO	Extra SCL Clock Cycle Output This bit clears automatically after 1 clock cycle is output. 0: Do not output extra SCL clock cycle (default) 1: Output extra SCL clock cycle	R/W
6	IICRST	I^2C Bus Interface Internal Reset This setting clears the bit counter and the SCLn/SDAn output latch. 0: Release IIC reset or internal reset 1: Initiate IIC reset or internal reset	R/W
7	ICE	I^2C Bus Interface Enable Used in combination with the IICRST bit to select either IIC or internal reset. 0: Disable (SCLn and SDAn pins in inactive state) 1: Enable (SCLn and SDAn pins in active state)	R/W

Note: S-TYPE3, P-TYPE3

SDAO bit (SDA Output Control/Monitor) and SCLO bit (SCL Output Control/Monitor)

The SDAO and SCLO bits directly control the SDAn and SCLn signals output from the IIC. When writing to these bits, also write 0 to the SOWP bit. Setting these bits results in input to the IIC by the input buffer. When slave mode is selected, a start condition might be detected and the bus might be released, depending on the bit settings.

Do not rewrite these bits during a start condition, stop condition, restart condition, transmission, or reception. Operation after rewriting under these conditions is not guaranteed. When reading these bits, the state of signals output from the IIC can be read.

CLO bit (Extra SCL Clock Cycle Output)

The CLO bit allows output of an extra SCL clock cycle for debugging or error processing. Normally, set this bit to 0. Setting the bit to 1 in a normal communication state causes a communication error. For details on this function, see section 32.12.2. Extra SCL Clock Cycle Output Function.

IICRST bit (I^2C Bus Interface Internal Reset)

The IICRST bit initiates an internal state reset of the IIC. Setting this bit to 1 initiates an IIC reset or internal reset. Whether an IIC reset or internal reset is initiated is determined by the settings of this bit in combination with the ICE bit. Table 32.3 lists the IIC resets.

The IIC reset initializes all registers except ICCR1.ICE and ICCR1.IICRST bits and internal states of the IIC. In addition to the internal states of the IIC, the internal reset initializes the following:

- Bit counter (ICMR1.BC[2:0] bits)
- I^2C Bus Shift Register (ICDRS)
- I^2C Bus Status Registers (ICSR1 and ICSR2)
- SDAO and SCLO Output Control/Monitor (ICCR1.SDAO and ICCR1.SCLO bits)

- I²C Bus Control Register 2 (except ICCR2.BBSY bit)

For the reset conditions for each register, see section 32.15. State of Registers When Issuing Each Condition.

An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the IIC without initializing the port settings and the control and setting registers of the IIC. If the IIC hangs in a low-level output state, resetting the internal states cancels the low-level output state and releases the bus with the SCLn pin and SDAn pin at high impedance.

Note: If an internal reset is initiated using the IICRST bit for a bus hang-up that occurs during communication with the master device in slave mode, the slave and master devices might enter different states, because the bit counter information differs. For this reason, do not initiate an internal reset in slave mode. Initiate recovery processing from the master device. If an internal reset is required because the IIC hangs with the SCLn line in a low-level output state in slave mode, initiate an internal reset, then issue a restart condition from the master device, or issue a stop condition and resume communication from the start condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start or restart condition from the master device, synchronization is lost because the master and slave devices operate asynchronously.

Table 32.3 IIC resets

IICRST	ICE	State	Specifications
1	0	IIC reset	Resets all registers except ICCR1.IICRST and ICCR1.ICE bits, and the internal states of the IIC
	1	Internal reset	Reset the following: • ICMR1.BC[2:0] bits • ICSR1, ICSR2, ICDRS registers • SDAO and SCLO Output Control/Monitor (ICCR1.SDAO and ICCR1.SCLO bits) • I²C Bus Control Register 2 (except ICCR2.BBSY bit) • Internal states of the IIC

ICE bit (I²C Bus Interface Enable)

The ICE bit selects the active or inactive state of the SCLn and SDAn pins. It can also be combined with the IICRST bit to initiate two types of resets. See Table 32.3 for the reset descriptions.

Set the ICE bit to 1 when using the IIC. The SCLn and SDAn pins are placed in the active state when the ICE bit is set to 1. Set the ICE bit to 0 when the IIC is not used. The SCLn and SDAn pins are placed in the inactive state when the ICE bit is set to 0. Do not assign the SCLn or SDAn pin to the IIC when setting up the pin function control. Slave address comparison is performed if the pins are assigned to the IIC.

32.2.2 ICCR2 : I²C Bus Control Register 2

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 1)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	ST	Start Condition Issuance Request 0: Do not issue a start condition request 1: Issue a start condition request	R/W
2	RS	Restart Condition Issuance Request 0: Do not issue a restart condition request 1: Issue a restart condition request	R/W

Bit	Symbol	Function	R/W
3	SP	Stop Condition Issuance Request 0: Do not issue a stop condition request 1: Issue a stop condition request	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	TRS	Transmit/Receive Mode 0: Receive mode 1: Transmit mode	R/W*1
6	MST	Master/Slave Mode 0: Slave mode 1: Master mode	R/W*1
7	BBSY	Bus Busy Detection Flag 0: I2C bus released (bus free state) 1: I2C bus occupied (bus busy state)	R

Note: S-TYPE3, P-TYPE3
Note 1. The MST and TRS bits can be written to when the ICMR1.MTWP bit is set to 1.

ST bit (Start Condition Issuance Request)

The ST bit requests transition to master mode and triggers a start condition. When this bit is set to 1, a start condition is issued when the BBSY flag is set to 0 (bus free state). For details on this function, see section 32.11. Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the ST bit.

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition is issued (a start condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Only set the ST bit to 1 (start condition request) when the BBSY flag is set to 0 (bus free state). Arbitration might be lost if the ST bit is set to 1 (start condition request) when the BBSY flag is 1 (bus busy state).

RS bit (Restart Condition Issuance Request)

The RS bit requests that a restart condition be issued in master mode. When this bit is set to 1 to request a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode). For details on this function, see section 32.11. Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the RS bit with the BBSY flag in ICCR2 set to 1.

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition is issued (a start condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Do not set the RS bit to 1 while issuing a stop condition.
Note: If 1 (restart condition request) is written to the RS bit in slave mode, the restart condition is not issued, but the RS bit remains set to 1. If the operating mode changes to master mode without the bit being cleared, a restart condition might be issued.

SP bit (Stop Condition Issuance Request)

The SP bit requests that a stop condition be issued in master mode. When this bit is set to 1, a stop condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode). For details on this function, see section 32.11. Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the SP bit with both the BBSY flag and the MST bit in ICCR2 set to 1.

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition is issued (a stop condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Writing to the SP bit is not possible while the BBSY flag is 0 (bus free state).
Note: Do not set the SP bit to 1 while a restart condition is being issued.

TRS bit (Transmit/Receive Mode)

The TRS bit indicates transmit or receive mode. The IIC is in receive mode when the TRS bit is 0 and in transmit mode when the bit is 1. The combination of this bit and the MST bit indicates the IIC operating mode.

The value of the TRS bit automatically changes to 1 for transmit mode or 0 for receive mode when a start condition is issued or detected and the R/W# bit is set. Although writing to the TRS bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not required during normal usage.

[Setting conditions]

- When a start condition is issued normally because of a start condition request (when a start condition is detected with the ST bit set to 1)
- When a restart condition is issued normally because of a restart condition request (when a restart condition is detected with the RS bit set to 1)
- When the R/W# bit appended to the slave address is set to 0 in master mode
- When the address received in slave mode matches the address enabled in ICSER, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the MTWP bit in ICMR1 set to 1.

[Clearing conditions]

- When a stop condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When the R/W# bit appended to the slave address is set to 1 in master mode
- In slave mode, on a match between the received address and the address enabled in ICSER when the value of the received R/W# bit is 0, including when the received address is the general call address
- In slave mode, when a restart condition is detected (a restart condition is detected with ICCR2.BBSY = 1 and ICCR2.MST = 0)
- When 0 is written to the TRS bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

MST bit (Master/Slave Mode)

The MST bit indicates master or slave mode. The IIC is in slave mode when the MST bit is 0 and is in master mode when the bit is 1. The combination of this bit and the TRS bit indicates the IIC operating mode.

The value of the MST bit automatically changes to 1 for master mode or 0 for slave mode when a start condition is issued or a stop condition is issued or detected. Although writing to the MST bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not required during normal usage.

[Setting conditions]

- When a start condition is issued normally because of a start condition request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the MTWP bit in ICMR1 set to 1.

[Clearing conditions]

- When a stop condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 0 is written to the MST bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

BBSY flag (Bus Busy Detection Flag)

The BBSY flag indicates whether the I^2C bus is occupied (bus busy state) or released (bus free state). The flag is set to 1 when the SDAn line changes from high to low when the SCLn line is high, assuming that a start condition was issued. The flag is set to 0 if a start condition is not detected for the bus free time (ICBRL setting), assuming that a stop condition was issued.

[Setting condition]

- When a start condition is detected.

[Clearing conditions]

- When a start condition is not detected for the bus free time (ICBRL setting) after detecting a stop condition
- When 1 is written to the IICRST bit in ICCR1 with the ICE bit in ICCR1 set to 0 (IIC reset).

32.2.3 ICMR1 : I^2C Bus Mode Register 1

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 1)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MTWP	CKS[2:0]			BCWP	BC[2:0]		
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
2:0	BC[2:0]	Bit Counter 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits	R/W*1
3	BCWP	BC Write Protect This bit is read as 1. 0: Write enable BC[2:0] bits 1: Write protect BC[2:0] bits	W*1
6:4	CKS[2:0]	Internal Reference Clock Select Select the internal reference clock source (IICφ) for the IIC. IICφ = (PCLKB / $2^{CKS[2:0]}$) clock	R/W
7	MTWP	MST/TRS Write Protect 0: Write protect MST and TRS bits in ICCR2 1: Write enable MST and TRS bits in ICCR2	R/W

Note: S-TYPE3, P-TYPE3

Note 1. Rewrite the BC[2:0] bits and set the BCWP bit to 0 at the same time.

BC[2:0] bits (Bit Counter)

The BC[2:0] bits function as a counter indicating the number of bits remaining to be transferred on detection of a rising edge on the SCLn line. Although BC[2:0] are read/write bits, it is not required to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one, for an additional acknowledge bit, between transferred frames when the SCLn line is at a low level. The value in the BC[2:0] bits returns to 000b at the end of a data transfer, including the acknowledge bit, or when a start or restart condition is detected.

32.2.4 ICMR2 : I2C Bus Mode Register 2

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 1)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Function	R/W
0	TMOS	Timeout Detection Time Select 0: Select long mode 1: Select short mode	R/W
1	TMOL	Timeout L Count Control 0: Disable count while SCLn line is low 1: Enable count while SCLn line is low	R/W
2	TMOH	Timeout H Count Control 0: Disable count while SCLn line is high 1: Enable count while SCLn line is high	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	SDDL[2:0]	SDA Output Delay Counter 0 0 0: No output delay 0 0 1: 1 IICφ cycle (When ICMR2.DLCS = 0 (IICφ)) 1 or 2 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 0 1 0: 2 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 3 or 4 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 0 1 1: 3 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 5 or 6 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 0 0: 4 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 7 or 8 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 0 1: 5 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 9 or 10 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 1 0: 6 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 11 or 12 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 1 1: 7 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 13 or 14 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2))	R/W
7	DLCS	SDA Output Delay Clock Source Select 0: Select internal reference clock (IICφ) as the clock source for SDA output delay counter 1: Select internal reference clock divided by 2 (IICφ/2) as the clock source for SDA output delay counter*1	R/W

Note: S-TYPE3, P-TYPE3
Note 1. The setting DLCS = 1 (IICφ/2) is only valid when SCL is low. When SCL is high, the DLCS = 1 setting becomes invalid and the clock source becomes the internal reference clock (IICφ).

TMOS bit (Timeout Detection Time Select)

The TMOS bit selects long or short mode for the timeout detection time when the timeout function is enabled (ICFER.TMOE = 1). When this bit is set to 0, long mode is selected. When it is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16 bit-counter. In short mode, the counter functions as a

14- bit counter. While the SCLn line is in the state that enables this counter as specified in the TMOH and TMOL bits, the counter counts up in synchronization with the internal reference clock (IICφ) as a count source. For details on this function, see section 32.12.1. Timeout Function.

TMOL bit (Timeout L Count Control)

The TMOL bit enables or disables up-counting on the internal counter of the timeout function while the SCLn line is held low and the timeout function is enabled (ICFER.TMOE = 1).

TMOH bit (Timeout H Count Control)

The TMOH bit enables or disables up-counting on the internal counter of the timeout function while the SCLn line is held high and the timeout function is enabled (ICFER.TMOE = 1).

SDDL[2:0] bits (SDA Output Delay Counter)

The SDDL[2:0] bits can be used to delay the SDA output. This counter works with the clock source selected in the DLCS bit. This setting can be used for all types of SDA output, including transmission of the acknowledge bit.

Set the SDA output delay to meet the I2C bus standard for the data enable time/acknowledge enable time,*1 or the SMBus standard, within [data hold time (300 ns or more + the SCL clock low-level period) - the data setup time (250 ns)]. If a value outside the standard is set, communication between devices might malfunction or falsely indicate a start or stop condition, depending on the bus state.

For details on this function, see section 32.5. SDA Output Delay Function.

Note 1. Data enable time/acknowledge enable time.
3,450 ns for up to 100 kbps: Standard mode (Sm)
900 ns for up to 400 kbps: Fast mode (Fm)
450 ns for up to 1 Mbps: Fast-mode Plus (Fm+)

32.2.5 ICMR3 : I2C Bus Mode Register 3

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 1)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	NF[1:0]	Noise Filter Stage Select 0 0: Filter out noise of up to 1 IICφ cycle (single-stage filter) 0 1: Filter out noise of up to 2 IICφ cycles (2-stage filter) 1 0: Filter out noise of up to 3 IICφ cycles (3-stage filter) 1 1: Filter out noise of up to 4 IICφ cycles (4-stage filter)	R/W
2	ACKBR	Receive Acknowledge 0: 0 received as the acknowledge bit (ACK reception) 1: 1 received as the acknowledge bit (NACK reception)	R
3	ACKBT	Transmit Acknowledge 0: Send 0 as the acknowledge bit (ACK transmission) 1: Send 1 as the acknowledge bit (NACK transmission)	R/W*1
4	ACKWP	ACKBT Write Protect 0: Write protect ACKBT bit 1: Write enable ACKBT bit	R/W

Bit	Symbol	Function	R/W
5	RDRFS	RDRF Flag Set Timing Select Low-hold is released by writing to ACKBT. 0: Set the RDRF flag on the rising edge of the 9th SCL clock cycle. The SCLn line is not held low on the falling edge of the 8th clock cycle. 1: Set the RDRF flag on the rising edge of the 8th SCL clock cycle. The SCLn line is held low on the falling edge of the 8th clock cycle.	R/W*2
6	WAIT	Low-hold is released by reading ICDRR. 0: No wait (The SCLn line is not held low during the period between the 9th clock cycle and the 1st clock cycle.) 1: Wait (The SCLn line is held low during the period between the 9th clock cycle and the 1st clock cycle.)	R/W*2
7	SMBS	SMBus/I2C Bus Select 0: Select I2C Bus 1: Select SMBus	R/W

Note: S-TYPE3, P-TYPE3
Note 1. Write to the ACKBT bit only while the ACKWP bit is already 1. If the application writes 1 to the ACKWP and ACKBT bits at the same time, the ACKBT bit is not set to 1.
Note 2. The WAIT and RDRFS bits are only valid in receive mode (invalid in transmit mode).

NF[1:0] bits (Noise Filter Stage Select)

The NF[1:0] bits select the number of stages in the digital noise filter. For details on this function, see section 32.6. Digital Noise Filter Circuits

Note: Set the noise range to be filtered within a range less than the SCLn line high- or low-level period. If the noise range is set to a value of [SCL clock width: high- or low-level period, whichever is shorter] - [1.5 internal reference clock (IICφ) cycles + analog noise filter: 120 ns (reference values)] or more, the SCL clock is regarded as noise, which might prevent the IIC from operating normally.

ACKBR bit (Receive Acknowledge)

The ACKBR bit stores the acknowledge bit information received from the receive device in transmit mode.

[Setting condition]

- When 1 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1.

[Clearing conditions]

- When 0 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (IIC reset).

ACKBT bit (Transmit Acknowledge)

The ACKBT bit sets the acknowledge bit to be sent in receive mode

[Setting condition]

- When 1 is written to this bit with the ACKWP bit set to 1.

[Clearing conditions]

- When 0 is written to this bit with the ACKWP bit set to 1
- When stop condition request is detected with the SP bit in ICCR2 set to 1
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (IIC reset).

ACKWP bit (ACKBT Write Protect)

The ACKWP bit controls write enabling of the ACKBT bit.

RDRFS bit (RDRF Flag Set Timing Select)

The RDRFS bit selects the RDRF flag set timing in receive mode and also selects whether to hold the SCLn line low on the falling edge of the 8th SCL clock cycle.

When the RDRFS bit is 0, the SCLn line is not held low on the falling edge of the 8th SCL clock cycle, and the RDRF flag is set to 1 on the rising edge of the 9th SCL clock cycle.

When the RDRFS bit is 1, the RDRF flag is set to 1 on the rising edge of the 8th SCL clock cycle, and the SCLn line is held low on the falling edge of the 8th SCL clock cycle. The low-hold of the SCLn line is released by a write to the ACKBT bit.

After data is received with this setting, the SCLn line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT = 0) or NACK (ACKBT = 1), based on the receive data.

WAIT bit (WAIT)

The WAIT bit controls whether to force a low-hold between the ninth SCL clock cycle and the first SCL clock cycle, until the receive data buffer (ICDRR) is completely read each time single-byte data is received in receive mode.

When the WAIT bit is 0, the receive operation continues without a low-hold between the ninth and the first SCL clock cycle. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.

When the WAIT bit is 1, the SCLn line is held low from the falling edge of the ninth clock cycle until the ICDRR value is read each time single-byte data is received. This enables receive operation in byte units.

Note: When the value of the WAIT bit is to be read, always read ICDRR first.

SMBS bit (SMBus/I2C Bus Select)

Setting the SMBS bit to 1 selects the SMBus and enables the HOAE bit in ICSER.

32.2.6 ICFER : I2C Bus Function Enable Register

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 1)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x05

Bit position:	7	6	5	4	3	2	1	0
Bit field:	FMPE	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset:	0	1	1	1	0	0	1	0

Bit	Symbol	Function	R/W
0	TMOE	Timeout Function Enable 0: Disable 1: Enable	R/W
1	MALE	Master Arbitration-Lost Detection Enable 0: Disable the arbitration-lost detection function and disable automatic clearing of the MST and TRS bits in ICCR2 when arbitration is lost 1: Enable the arbitration-lost detection function and enable automatic clearing of the MST and TRS bits in ICCR2 when arbitration is lost	R/W
2	NALE	NACK Transmission Arbitration-Lost Detection Enable 0: Disable 1: Enable	R/W
3	SALE	Slave Arbitration-Lost Detection Enable 0: Disable 1: Enable	R/W
4	NACKE	NACK Reception Transfer Suspension Enable 0: Do not suspend transfer operation during NACK reception (disable transfer suspension) 1: Suspend transfer operation during NACK reception (enable transfer suspension)	R/W
5	NFE	Digital Noise Filter Circuit Enable 0: Do not use the digital noise filter circuit 1: Use the digital noise filter circuit	R/W
6	SCLE	SCL Synchronous Circuit Enable 0: Do not use the SCL synchronous circuit 1: Use the SCL synchronous circuit	R/W

Bit	Symbol	Function	R/W
7	FMPE*1	Fast-Mode Plus Enable 0: Do not use the Fm+ slope control circuit for the SCLn and SDAn pins 1: Use the Fm+ slope control circuit for the SCLn and SDAn pins.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. The Fast-mode Plus enable bit (FMPE) is supported by IIC0 (SCL0_A, SDA0_A) and IIC1(SCL1_A, SDA1_A). Bit [7] is the reserved bit in the not supported channel.

TMOE bit (Timeout Function Enable)

The TMOE bit enables or disables the timeout function. For details on this function, see section 32.12.1. Timeout Function.

MALE bit (Master Arbitration-Lost Detection Enable)

The MALE bit specifies whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

NALE bit (NACK Transmission Arbitration-Lost Detection Enable)

The NALE bit specifies whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode, for example when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with a different number of receive bytes.

SALE bit (Slave Arbitration-Lost Detection Enable)

The SALE bit specifies whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode, for example when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs because of noise.

NACKE bit (NACK Reception Transfer Suspension Enable)

The NACKE bit specifies whether to continue or discontinue the transfer operation when NACK is received in transmit mode. Normally, set this bit to 1.

When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended. When the NACKE bit is 0, the next transfer operation continues regardless of the received acknowledge content.

For details, see section 32.9.2. NACK Reception Transfer Suspension Function.

SCLE bit (SCL Synchronous Circuit Enable)

The SCLE bit specifies whether to synchronize the SCL clock with the SCL input clock. Normally, set this bit to 1.

When the SCLE bit is set to 0 (no SCL synchronous circuit used), the IIC does not synchronize the SCL clock with the SCL input clock. With this setting, the IIC outputs the SCL clock at the transfer rate set in ICBRH and ICBRL, regardless of the SCLn line state. For this reason, if the bus load of the I^2C bus line is much larger than the specification value, or if the SCL clock output overlaps in multiple masters, a short-cycle SCL clock that does not meet the specification might be output. When no SCL synchronous circuit is used, it also affects the issuing of the start, restart, and stop conditions, and the continuous output of extra SCL clock cycles.

Do not set this bit to 0 except when checking the output of the set transfer rate.

FMPE bit (Fast-Mode Plus Enable)

The FMPE bit specifies whether to use a slope control circuit for Fast-mode Plus (Fm+).

When this bit is set to 1, a slope control circuit conforming to the I^2C bus Fast-mode Plus (Fm+) standard (tof) is selected. When this bit is set to 0, a slope control circuit conforming to the I^2C bus Standard-mode (Sm) and Fast-mode (Fm) standards (tof) is selected.

Set this bit to 1 when using transmission rates up to 1 Mbps (Fast-mode Plus (Fm+) standard). Set it to 0 when using other transmission rates (up to 100 kbps (Sm) or up to 400 kbps (Fm)) or for SMBus (10 to 100 kbps).

32.2.7 ICSER : I²C Bus Status Enable Register

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 1)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E
Value after reset:	0	0	0	0	1	0	0	1

Bit	Symbol	Function	R/W
0	SAR0E	Slave Address Register 0 Enable 0: Disable slave address in SARL0 and SARU0 1: Enable slave address in SARL0 and SARU0	R/W
1	SAR1E	Slave Address Register 1 Enable 0: Disable slave address in SARL1 and SARU1 1: Enable slave address in SARL1 and SARU1	R/W
2	SAR2E	Slave Address Register 2 Enable 0: Disable slave address in SARL2 and SARU2 1: Enable slave address in SARL2 and SARU2	R/W
3	GCAE	General Call Address Enable 0: Disable general call address detection 1: Enable general call address detection	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	DIDE	Device-ID Address Detection Enable 0: Disable device-ID address detection 1: Enable device-ID address detection	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	HOAE	Host Address Enable 0: Disable host address detection 1: Enable host address detection	R/W

Note: S-TYPE3, P-TYPE3

SARyE bit (Slave Address Register y Enable) (y = 0 to 2)

The SARyE bit enables or disables the received slave address and the slave address set in SARLy and SARUy.

When this bit is set to 1, the slave address set in SARLy and SARUy is enabled and is compared with the received slave address. When this bit is set to 0, the slave address set in SARLy and SARUy is disabled and is ignored even if it matches the received slave address.

GCAE bit (General Call Address Enable)

The GCAE bit specifies whether to ignore the general call address (0000 000b + 0 [W]: All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the IIC recognizes the received slave address as the general call address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2) and performs the data receive operation. When this bit is set to 0, the received slave address is ignored even if it matches the general call address.

DIDE bit (Device-ID Address Detection Enable)

The DIDE bit specifies whether to recognize and execute the device-ID address when a device ID (1111 100b) is received in the first frame after a start or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the IIC recognizes that the device-ID address was received. When the next R/W# bit is 0 (W), the IIC recognizes the second and the subsequent frames as slave addresses and continues the receive operation. When this bit is set to 0, the IIC ignores the received first frame even if it matches the device-ID address, and it recognizes the first frame as a normal slave address.

For details on this function, see section 32.7.3. Device-ID Address Detection.

HOAE bit (Host Address Enable)

The HOAE bit specifies whether to ignore the received host address (0001 000b) when the SMBS bit in ICMR3 is 1.

When this bit is set to 1 while the SMBS bit in ICMR3 is 1, if the received slave address matches the host address, the IIC recognizes the received slave address as the host address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2) and performs the receive operation.

When the SMBS bit in ICMR3 or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

32.2.8 ICIER : I²C Bus Interrupt Enable Register

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 1)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x07

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMOIE	Timeout Interrupt Request Enable 0: Disable timeout interrupt (TMOI) request 1: Enable timeout interrupt (TMOI) request	R/W
1	ALIE	Arbitration-Lost Interrupt Request Enable 0: Disable arbitration-lost interrupt (ALI) request 1: Enable arbitration-lost interrupt (ALI) request	R/W
2	STIE	Start Condition Detection Interrupt Request Enable 0: Disable start condition detection interrupt (STI) request 1: Enable start condition detection interrupt (STI) request	R/W
3	SPIE	Stop Condition Detection Interrupt Request Enable 0: Disable stop condition detection interrupt (SPI) request 1: Enable stop condition detection interrupt (SPI) request	R/W
4	NAKIE	NACK Reception Interrupt Request Enable 0: Disable NACK reception interrupt (NAKI) request 1: Enable NACK reception interrupt (NAKI) request	R/W
5	RIE	Receive Data Full Interrupt Request Enable 0: Disable receive data full interrupt (IICn_RXI) request 1: Enable receive data full interrupt (IICn_RXI) request	R/W
6	TEIE	Transmit End Interrupt Request Enable 0: Disable transmit end interrupt (IICn_TEI) request 1: Enable transmit end interrupt (IICn_TEI) request	R/W
7	TIE	Transmit Data Empty Interrupt Request Enable 0: Disable transmit data empty interrupt (IICn_TXI) request 1: Enable transmit data empty interrupt (IICn_TXI) request	R/W

Note: S-TYPE3, P-TYPE3

TMOIE bit (Timeout Interrupt Request Enable)

The TMOIE bit enables or disables timeout interrupt (TMOI) requests when the TMOF flag in ICSR2 is 1. To cancel a TMOI interrupt request, set the TMOF flag or the TMOIE bit to 0.

ALIE bit (Arbitration-Lost Interrupt Request Enable)

The ALIE bit enables or disables arbitration-lost interrupt (ALI) requests when the AL flag in ICSR2 is 1. To cancel an ALI interrupt request, set the AL flag or the ALIE bit to 0.

STIE bit (Start Condition Detection Interrupt Request Enable)

The STIE bit enables or disables start condition detection interrupt (STI) requests when the START flag in ICSR2 is 1. To cancel an STI interrupt request, set the START flag or the STIE bit to 0.

SPIE bit (Stop Condition Detection Interrupt Request Enable)

The SPIE bit enables or disables stop condition detection interrupt (SPI) requests when the STOP flag in ICSR2 is 1. To cancel an SPI interrupt request, set the STOP flag or the SPIE bit to 0.

NAKIE bit (NACK Reception Interrupt Request Enable)

The NAKIE bit enables or disables NACK reception interrupt (NAKI) requests when the NACKF flag in ICSR2 is 1. To cancel an NAKI interrupt request, set the NACKF flag or the NAKIE bit to 0.

RIE bit (Receive Data Full Interrupt Request Enable)

The RIE bit enables or disables receive data full interrupt (IICn_RXI) requests when the RDRF flag in ICSR2 is 1.

TEIE bit (Transmit End Interrupt Request Enable)

The TEIE bit enables or disables transmit end interrupt (IICn_TEI) requests when the TEND flag in ICSR2 is 1. To cancel an IICn_TEI interrupt request, set the TEND flag or the TEIE bit to 0.

TIE bit (Transmit Data Empty Interrupt Request Enable)

The TIE bit enables or disables transmit data empty interrupt (IICn_TXI) requests when the TDRE flag in ICSR2 is 1.

32.2.9 ICSR1 : I2C Bus Status Register 1

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 1)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AAS0	Slave Address 0 Detection Flag 0: Slave address 0 not detected 1: Slave address 0 detected	R/(W)*1
1	AAS1	Slave Address 1 Detection Flag 0: Slave address 1 not detected 1: Slave address 1 detected	R/(W)*1
2	AAS2	Slave Address 2 Detection Flag 0: Slave address 2 not detected 1: Slave address 2 detected	R/(W)*1
3	GCA	General Call Address Detection Flag 0: General call address not detected 1: General call address detected	R/(W)*1
4	—	This bit is read as 0. The write value should be 0.	R/W
5	DID	Device-ID Address Detection Flag This bit is set to 1 when the first frame received immediately after a start condition is detected matches a value of (device ID (1111 100b) + 0[W]). 0: Device-ID command not detected 1: Device-ID command detected	R/(W)*1
6	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	HOA	Host Address Detection Flag This bit is set to 1 when the received slave address matches the host address (0001 000b). 0: Host address not detected 1: Host address detected	R/(W)*1

Note: S-TYPE3, P-TYPE3
Note 1. Only 0 can be written to clear the flag.

AASy flag (Slave Address y Detection flag) (y = 0 to 2)

The AASy flag indicates whether slave address y was detected.

[Setting conditions]

For 7-bit address format (SARUy.FS = 0):

- When the received slave address matches the SVA[6:0] value in SARLy, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

For 10-bit address format: (SARUy.FS = 1):

- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy), and the subsequent address matches the SARLy value, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the AASy flag after reading AASy = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

For 7-bit address format (SARUy.FS = 0):

- When the received slave address does not match the SVA[6:0] value in SARLy, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.

For 10-bit address format (SARUy.FS = 1):

- When the received slave address does not match a value of (11110b + SVA[1:0] in SARUy), with the SARyE bit in ICSER set to 1 (slave address y detection enabled)
  The AASy flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy), and the subsequent address does not match the SARLy value, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.

GCA flag (General Call Address Detection Flag)

The GCA flag indicates whether the general call address was detected.

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 [W]), with the GCAE bit in ICSER set to 1 (general call address detection enabled).
  The GCA flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA flag after reading GCA = 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000b + 0 [W]), with the GCAE bit in ICSER set to 1 (general call address detection enabled)
  The GCA flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.

- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

DID flag (Device-ID Address Detection Flag)

The DID flag indicates whether the device-ID address was detected.

[Setting condition]

- When the first frame received immediately after a start or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]), with the DIDE bit in ICSER set to 1 (device-ID address detection enabled).
  The DID flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID flag after reading DID = 1
- When a stop condition is detected
- When the first frame received immediately after a start or restart condition is detected does not match a value of (device ID (1111 100b)), with the DIDE bit in ICSER set to 1 (device-ID address detection enabled)
  The DID flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When the first frame received immediately after a start or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]), and the second frame does not match any slave address from 0 to 2, with the DIDE bit in ICSER set to 1 (device-ID address detection enabled)
  The DID flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

HOA flag (Host Address Detection Flag)

The HOA flag indicates whether the host address was detected.

[Setting condition]

- When the received slave address matches the host address (0001 000b), with the HOAE bit in ICSER set to 1 (host address detection enabled).
  The HOA flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the HOA flag after reading HOA = 1
- When a stop condition is detected
- When the received slave address does not match the host address (0001 000b), with the HOAE bit in ICSER set to 1 (host address detection enabled)
  The HOA flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

32.2.10 ICSR2 : I²C Bus Status Register 2

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 1)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x09

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMOF	Timeout Detection Flag 0: Timeout not detected 1: Timeout detected	R/(W)*1

Bit	Symbol	Function	R/W
1	AL	Arbitration-Lost Flag 0: Arbitration not lost 1: Arbitration lost	R/(W)*1
2	START	Start Condition Detection Flag 0: Start condition not detected 1: Start condition detected	R/(W)*1
3	STOP	Stop Condition Detection Flag 0: Stop condition not detected 1: Stop condition detected	R/(W)*1
4	NACKF	NACK Detection Flag 0: NACK not detected 1: NACK detected	R/(W)*1
5	RDRF	Receive Data Full Flag 0: ICDRR contains no receive data 1: ICDRR contains receive data	R/(W)*1
6	TEND	Transmit End Flag 0: Data being transmitted 1: Data transmit complete	R/(W)*1
7	TDRE	Transmit Data Empty Flag 0: ICDRT contains transmit data 1: ICDRT contains no transmit data	R

Note: S-TYPE3, P-TYPE3
Note 1. Only 0 can be written, to clear the flag.

TMOF flag (Timeout Detection Flag)

The TMOF flag is set to 1 when the IIC detects a timeout because the SCLn line state remains unchanged for the set period.

[Setting condition]

- When the SCLn line state remains unchanged for the period specified in the ICMR2.TMOH, TMOL, and TMOS bits while the ICFER.TMOE bit is 1 (timeout function enabled) in master or in slave mode and the received slave address matches.

[Clearing conditions]

- When 0 is written to the TMOF flag after reading TMOF = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

AL flag (Arbitration-Lost Flag)

The AL flag indicates that bus mastership was lost in arbitration because of a bus conflict or some other reason when a start condition was issued or an address and data was transmitted. The IIC monitors the level on the SDAn line during transmission and, if the level on the line does not match the value of the bit being output, is set the value of the AL flag to 1 to indicate that the bus is occupied by another device.

The IIC can also set the AL flag to indicate the detection of arbitration loss during NACK transmission or during data transmission.

[Setting conditions]

When master arbitration-lost detection is enabled (ICFER.MALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock except for the ACK period during data transmission in master transmit mode
- When a start condition is detected while the ST bit in ICCR2 is 1 (start condition requested) or the internal SDA output state does not match the SDAn line level
- When the ST bit in ICCR2 is 1 (start condition requested), with the BBSY flag in ICCR2 set to 1.

When NACK arbitration-lost detection is enabled (ICFER.NALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock in the ACK period during NACK transmission in receive mode.

When slave arbitration-lost detection is enabled (ICFER.SALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock, except for the ACK period during data transmission in slave transmit mode.

[Clearing conditions]

- When 0 is written to the AL flag after reading AL = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Table 32.4 Relationship between arbitration-lost generation sources and arbitration-lost enable functions

ICFER			ICSR2		
MALE	NALE	SALE	AL	Error	Arbitration-lost generation source
1	x	x	1	Start condition issuance error	When internal SDA output state does not match SDAn line level when a start condition is detected, while the ST bit in ICCR2 is 1
					When ST in ICCR2 is set to 1 while BBSY in ICCR2 is 1
			1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode
x	1	x	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master or slave receive mode
x	x	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

x: Don’t care

START flag (Start Condition Detection Flag)

The START flag indicates whether a start or restart condition was detected.

[Setting condition]

- When a start or restart condition is detected.

[Clearing conditions]

- When 0 is written to the START flag after reading START = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

STOP flag (Stop Condition Detection Flag)

The STOP flag indicates whether a stop condition was detected.

[Setting condition]

- When a stop condition is detected.

[Clearing conditions]

- When 0 is written to the STOP flag after reading STOP = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

NACKF flag (NACK Detection Flag)

The NACKF flag indicates whether a NACK was detected.

[Setting condition]

- When acknowledge is not received (NACK received) from the receive device in transmit mode, with the NACKE bit in ICFER set to 1 (transfer suspension enabled).

[Clearing conditions]

- When 0 is written to the NACKF flag after reading NACKF = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: When the NACKF flag is set to 1, the IIC suspends data transmission and reception. Writing to ICDRT in transmit mode or reading from ICDRR in receive mode with the NACKF flag set to 1 does not enable data transmit or receive operation. To restart data transmission or reception, set the NACKF flag to 0.

RDRF flag (Receive Data Full Flag)

The RDRF flag indicates whether the ICDRR contains receive data.

[Setting conditions]

- When receive data is transferred from ICDRS to ICDRR
  The RDRF flag is set to 1 on the rising edge of the eighth or ninth SCL clock cycle (selected in the RDRFS bit in ICMR3).
- When the received slave address matches after a start or restart condition is detected with the TRS bit in ICCR2 set to 0.

[Clearing conditions]

- When 0 is written to the RDRF flag after reading RDRF = 1
- When data is read from ICDRR
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

TEND flag (Transmit End Flag)

The TEND flag indicates completion of transmission.

[Setting condition]

- On the rising edge of the ninth SCL clock cycle while the TDRE flag is 1.

[Clearing conditions]

- When 0 is written to the TEND flag after reading TEND = 1
- When data is written to ICDRT
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

TDRE flag (Transmit Data Empty Flag)

The TDRE flag indicates whether the ICDRT contains transmit data.

[Setting conditions]

- When data is transferred from ICDRT to ICDRS and ICDRT becomes empty
- When the TRS bit in ICCR2 is set to 1
- When the received slave address matches while the TRS bit is 1.

[Clearing conditions]

- When data is written to ICDRT
- When the TRS bit in ICCR2 is set to 0
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: When the NACKF flag is set to 1 while the NACKE bit in ICFER is 1, the IIC suspends data transmission and reception. In this case, if the TDRE flag is 0 (next transmit data written), data is transferred to the ICDRS register and the ICDRT register becomes empty on the rising edge of the 9th clock cycle, but the TDRE flag does not set to 1.

32.2.11 ICWUR : I²C Bus Wakeup Unit Register

Base address: IIC0WU = 0x4025_E014
IIC0WU_NS = 0x5025_E014

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	WUE	WUIE	WUF	WUACK	—	—	—	WUAFA
Value after reset:	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	WUAFA	Wakeup Analog Filter Additional Selection 0: Do not add the wakeup analog filter 1: Add the wakeup analog filter	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	WUACK	ACK Bit for Wakeup Mode Choice of four response modes in combination with ICCR1.IICRST and WUACK. See Table 32.5.	R/W
5	WUF	Wakeup Event Occurrence Flag 0: Slave address not matching during wakeup 1: Slave address matching during wakeup	R/W
6	WUIE	Wakeup Interrupt Request Enable 0: Disable wakeup interrupt request (IIC0_WUI) 1: Enable wakeup interrupt request (IIC0_WUI)	R/W
7	WUE	Wakeup Function Enable 0: Disable wakeup function 1: Enable wakeup function	R/W

Note: S-TYPE3, P-TYPE3

Table 32.5 Wakeup mode

IICRST	WUACK	Operation mode	Description
0	0	Normal wakeup mode 1	ACK response on 9th SCL, and SCL low-hold after 9th SCL.
0	1	Normal wakeup mode 2	No ACK response immediately and SCL low-hold between 8th and 9th SCL. SCL low-hold release and ACK response on 9th SCL.
1	0	Command recovery mode	ACK response on 9th SCL and no SCL low-hold.
1	1	EEP response mode	NACK response on 9th SCL and no SCL low-hold.

WUF flag (Wakeup Event Occurrence Flag)

The WUF flag indicates whether the slave address is matching during wakeup.

[Setting condition]

- When PCLKB is supplied after the slave address of I²C bus is matched with the address whose enable bits of ICSER (except Device ID address) are set to 1 (enable), the bit is set.

[Clearing conditions]

- When 0 is written to the WUF flag after reading WUF = 1 (when WUSYF flag is set to 1).
- When ICCR1.ICE = 0 and IICRST = 1.

32.2.12 ICWUR2 : I2C Bus Wakeup Unit Register 2

Base address: IIC0WU = 0x4025_E014
IIC0WU_NS = 0x5025_E014

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	WUSYF	WUASYF	WUSEN
Value after reset:	1	1	1	1	1	1	0	1

Bit	Symbol	Function	R/W
0	WUSEN	Wakeup Function Synchronous Enable 0: IIC asynchronous circuit enable 1: IIC synchronous circuit enable	R/W
1	WUASYF	Wakeup Function Asynchronous Operation Status Flag 0: IIC synchronous circuit enable condition 1: IIC asynchronous circuit enable condition	R
2	WUSYF	Wakeup Function Synchronous Operation Status Flag 0: IIC asynchronous circuit enable condition 1: IIC synchronous circuit enable condition	R
7:3	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE3, P-TYPE3

WUSEN bit (Wakeup Function Synchronous Enable)

The WUSEN bit is used in combination with the WUASYF flag (or WUSYF flag) to switch between the PCLKB synchronous and asynchronous operation, when the wakeup effective function is enabled (ICWUR.WUE = 1).

The PCLKB operation switches from synchronous to asynchronous operation:

When the ICCR2.BBSY flag is 0, if 0 is written to the WUSEN bit while the WUASYF flag is 0, the reception occurs independently of the operation of PCLKB (with PCLKB stopped) after it switches to the PCLKB asynchronous operation, on wakeup event detection.

The PCLKB operation switches from asynchronous to synchronous operation:

- When 1 is written to the WUSEN bit, with the WUASYF flag at 1, when a wakeup event is detected. After writing 1, the WUASYF flag immediately becomes 0.
- When the stop condition is detected with a wakeup event undetected.

[Setting condition]

- When 1 is written to the WUSEN bit.
- ICCR1.ICE = 0 and IICRST = 1 (IIC reset)
- ICWUR.WUE = 0

[Clearing conditions]

- When 0 is written to the WUSEN bit.

WUASYF flag (Wakeup Function Asynchronous Operation Status Flag)

The WUASYF flag can place the IIC in PCLKB asynchronous operation when the wakeup effective function is enabled (ICWUR.WUE = 1).

[Setting condition]

- When the ICCR2.BBSY flag is 0, and the WUSEN bit is set to 0 with the ICWUR.WUE bit set to 1.

[Clearing conditions]

- When 1 is written to the WUSEN bit after detecting the wake-up event with ICWUR.WUE bit set to 1.

- When a stop condition is detected with WUSEN bit set to 1 before detecting the wake-up event with WUASY flag set to 1 with ICWUR.WUE bit set to 1.
- When you write 1 in the WUSEN bit with the WUASYF flag detected 1 and the wake-up event in the state of ICWUR.WUE = 1.
- ICCR1.ICE = 0 and ICCRST = 1 (ICC reset)
- ICWUR.WUE = 0.

WUSYF flag (Wakeup Function Synchronous Operation Status Flag)

It is shown that IIC is in the PCLKB synchronous operation at wake-up effective function (ICWUR.WUE = 1). This flag is a value in which the WUASYF flag is always reserved.

[Setting conditions]

- When 1 is written to the WUSEN bit after detecting the wake-up event with ICWUR.WUE bit set to 1 with WUSYF flag set to 0 with ICWUR.WUE bit set to 1.
- When a stop condition is detected with WUSEN bit set to 1 before detecting the wake-up event with WUSYF flag set to 0 with ICWUR.WUE bit set to 1.
- ICCR1.ICE = 0 and ICCRST = 1 (ICC reset)
- ICWUR.WUE = 0.

[Clearing condition]

- When the ICCR2.BBSY flag is 0 with the ICWUR.WUE bit set to 1 after writing 0 to the WUSEN bit.

32.2.13 SARLy : Slave Address Register Ly (y = 0 to 2)

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 1)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x0A + 0x02 × y

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SVA[6:0]							SVA0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SVA0	10-bit Address LSB Slave address setting.	R/W
7:1	SVA[6:0]	7-bit Address/10-bit Address Lower Bits Slave address setting.	R/W

Note: S-TYPE3, P-TYPE3

SVA0 bit (10-bit Address LSB)

When the 10-bit address format is selected (SARUy.FS = 1), the SVA0 bit functions as the LSB of a 10-bit address and is combined with the SVA[6:0] bits to form the lower 8 bits of a 10-bit address.

This bit is valid when the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1. When the SARUy.FS or SARyE bit is 0, the setting in this bit is ignored.

SVA[6:0] bits (7-bit Address/10-bit Address Lower Bits)

When the 7-bit address format is selected (SARUy.FS = 0), the SVA[6:0] bits function as a 7-bit address. When the 10-bit address format is selected (SARUy.FS = 1), these bits combine with the SVA0 bit to form the lower 8 bits of a 10-bit address.

When the SARyE bit in ICSER is 0, the setting in these bits is ignored.

32.2.14 SARUy : Slave Address Register Uy (y = 0 to 2)

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 1)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x0B + 0x02 × y

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SVA[1:0]		FS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FS	7-bit/10-bit Address Format Select 0: Select 7-bit address format 1: Select 10-bit address format	R/W
2:1	SVA[1:0]	10-bit Address Upper Bits Slave address setting.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

FS bit (7-bit/10-bit Address Format Select)

The FS bit selects 7- or 10-bit format for slave address y (in SARLy and SARUy).

When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SVA[6:0] setting in SARLy is valid, and the SVA[1:0] and SVA0 settings in SARLy are ignored.

When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, the 10-bit address format is selected for slave address y and the SVA[1:0] and SARLy settings are valid.

When the SARyE bit in ICSER is 0 (SARLy and SARUy disabled), the SARUy.FS setting is invalid.

SVA[1:0] bits (10-bit Address Upper Bits)

When the 10-bit address format is selected (FS = 1), the SVA[1:0] bits function as the upper 2 bits of a 10-bit address.

These bits are valid when the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1. When the SARUy.FS or SARyE bit is 0, the setting in these bits is ignored.

32.2.15 ICBRL : I2C Bus Bit Rate Low-Level Register

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 1)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	BRL[4:0]				
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
4:0	BRL[4:0]	Bit Rate Low-Level Period Low-level period of SCL clock.	R/W
7:5	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE3, P-TYPE3

BRL[4:0] bits (Bit Rate Low-Level Period)

The BRL[4:0] bits set the low-level period of the SCL clock. ICBRL counts the low-level period with the internal reference clock source (IICφ) specified by the CKS[2:0] bits in ICMR1. ICBRL also generates the data setup time for automatic SCL

low-hold operation, see section 32.9. Automatic Low-Hold Function for SCL. When the IIC is used in slave mode, the BRL[4:0] bits must be set to a value longer than the data setup time*1.

If the digital noise filter is enabled (NFE bit in ICFER is 1), set the BRL[4:0] bits to a value at least one greater than the number of stages in the noise filter. For details on the number of stages, see the description of the NF[1:0] bits in section 32.2.5. ICMR3 : I2C Bus Mode Register 3.

Note 1. Data setup time (t_{SU}: DAT)

250 ns for up to 100 kbps: Standard-mode (Sm)

100 ns for up to 400 kbps: Fast-mode (Fm)

50 ns for up to 1 Mbps: Fast-mode Plus (Fm+)

32.2.16 ICBRH : I2C Bus Bit Rate High-Level Register

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 1)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x11

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	BRH[4:0]				
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
4:0	BRH[4:0]	Bit Rate High-Level Period High-level period of SCL clock.	R/W
7:5	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE3, P-TYPE3

BRH[4:0] bits (Bit Rate High-Level Period)

The BRH[4:0] bits set the high-level period of SCL clock. BRH[4:0] bits are valid in master mode. If the IIC is used only in slave mode, do not set the BRH[4:0] bits.

ICBRH counts the high-level period with the internal reference clock source (IICφ) specified in the CKS[2:0] bits in ICMR1.

If the digital noise filter is enabled (the NFE bit in ICFER is 1), set these bits to a value at least one greater than the number of stages in the noise filter. For the number of stages in the noise filter, see the description of the NF[1:0] bits in section 32.2.5. ICMR3 : I2C Bus Mode Register 3.

The IIC transfer rate and the SCL clock duty are calculated using the following expressions (1) to (5):

1. ICFER.SCLE = 0
   Transfer rate = 1/[{(BRH + 1) + (BRL + 1)}/IICφ*1 + tr*2 + tf*2]
   Duty cycle = [tr + {(BRH + 1)/IICφ}]/[tr + tf + {(BRH + 1) + (BRL + 1)}/IICφ]
2. ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] = 000b (IICφ = PCLKB)
   Transfer rate = 1/[{(BRH + 3) + (BRL+ 3)}/IICφ + tr + tf]
   Duty cycle = [tr + {(BRH + 3)/IICφ}]/[tr + tf + {(BRH + 3) + (BRL + 3)}/IICφ]
3. ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] = 000b (IICφ = PCLKB)
   Transfer rate = 1/[{(BRH + 3 + nf*3) + (BRL + 3 + nf)}/IICφ + tr + tf]
   Duty cycle = [tr + {(BRH + 3 + nf)/IICφ}]/[tr + tf + {(BRH + 3 + nf) + (BRL + 3 + nf)}/IICφ]
4. ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] ≠ 000b
   Transfer rate = 1/[{(BRH + 2) + (BRL + 2)}/IICφ + tr + tf]
   Duty cycle = [tr + {(BRH + 2)/IICφ}]/[tr + tf + {(BRH + 2) + (BRL + 2)}/IICφ]
5. ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] ≠ 000b
   Transfer rate = 1/[{(BRH + 2 + nf) + (BRL + 2 + nf)}/IICφ + tr + tf]
   Duty cycle = [tr + {(BRH + 2 + nf)/IICφ}]/[tr + tf + {(BRH + 2 + nf) + (BRL + 2 + nf)}/IICφ]

Note 1. IICφ = PCLKB × division ratio
Note 2. The SCLn line rise time (tr) and SCLn line fall time (tf) depend on the total bus line capacitance (Cb) and the pull-up resistor (Rp). For details, see the I2C bus standard from NXP Semiconductors.
Note 3. nf = Number of digital noise filters selected in the ICMR3.NF bit.

Table 32.6 Example of ICBRH/ICBRL Settings for Transfer Rate IIC when SCLE = 0

Transfer rate (kbps)	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	PCLKB (MHz)	NF[1:0]	Computation expression
100	100b	14 (0xEE)	17 (0xF1)	60	—	(1)
400	010b	8 (0xE8)	19 (0xF3)	60	—	(1)
1000	000b	15 (0xEF)	29 (0xFD)	60	—	(1)

Table 32.7 Example of ICBRH/ICBRL Settings for Transfer Rate when SCLE = 1 and NFE = 0

Transfer rate (kbps)	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	PCLKB (MHz)	NF[1:0]	Computation expression
100	100b	13 (0xED)	16 (0xF0)	60	—	(4)
400	010b	7 (0xE7)	18 (0xF2)	60	—	(4)
1000	000b	13 (0xED)	27 (0xFB)	60	—	(2)

Table 32.8 Example of ICBRH/ICBRL Settings for Transfer Rate when SCLE = 1 and NFE = 1

Transfer rate (kbps)	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	PCLKB (MHz)	NF[1:0]	Computation expression
100	100b	11 (0xEB)	14 (0xEE)	60	01b	(5)
400	010b	5 (0xE5)	16 (0xF0)	60	01b	(5)
1000	000b	11 (0xEB)	25 (0xF9)	60	01b	(3)

32.2.17 ICDRT : I2C Bus Transmit Data Register

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 1)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x12

Bit position:	7	6	5	4	3	2	1	0
Bit field:								
Value after reset:	1	1	1	1	1	1	1	1

When ICDRT detects a space in the I2C Bus Shift Register (ICDRS), it transfers the transmit data that was written to ICDRT to ICDRS and starts transmitting data in transmit mode. The double-buffer structure of ICDRT and ICDRS allows continuous transmit operation if the next transmit data is written to ICDRT while the ICDRS data is being transmitted.

ICDRT can always be read and written to. Write transmit data to ICDRT once when a transmit data empty interrupt (IICn_TXI) request is generated.

32.2.18 ICDRR : I2C Bus Receive Data Register

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 1)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x13

Bit position:	7	6	5	4	3	2	1	0
Bit field:								
Value after reset:	0	0	0	0	0	0	0	0

When 1 byte of data is received, the received data is transferred from the I2C Bus Shift Register (ICDRS) to ICDRR to enable the next data to be received. The double-buffer structure of ICDRS and ICDRR allows continuous receive operation if the received data is read from ICDRR while ICDRS is receiving data. ICDRR cannot be written to. Read data from ICDRR once when a receive data full interrupt (IICn_RXI) request is generated.

If ICDRR receives the next receive data before the current data is read from ICDRR (while the RDRF flag in ICSR2 is 1), the IIC automatically holds the SCL clock low 1 cycle before the RDRF flag is set to 1 next.

32.2.19 ICDRS : I2C Bus Shift Register

Base address: n/a

Offset address: n/a

Bit position:	7	6	5	4	3	2	1	0
Bit field:								
Value after reset:	—	—	—	—	—	—	—	—

ICDRS is an 8-bit shift register for data transmit and receive. During transmission, transmit data is transferred from ICDRT to ICDRS and is transmitted from the SDAn pin. During reception, data is transferred from ICDRS to ICDRR after 1 byte of data is received. ICDRS cannot be accessed directly.

32.3 Operation

32.3.1 Communication Data Format

The I2C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a start or restart condition is an address frame that specifies a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a stop condition is issued.

Figure 32.3 shows the I2C bus format, and Figure 32.4 shows the I2C bus timing.

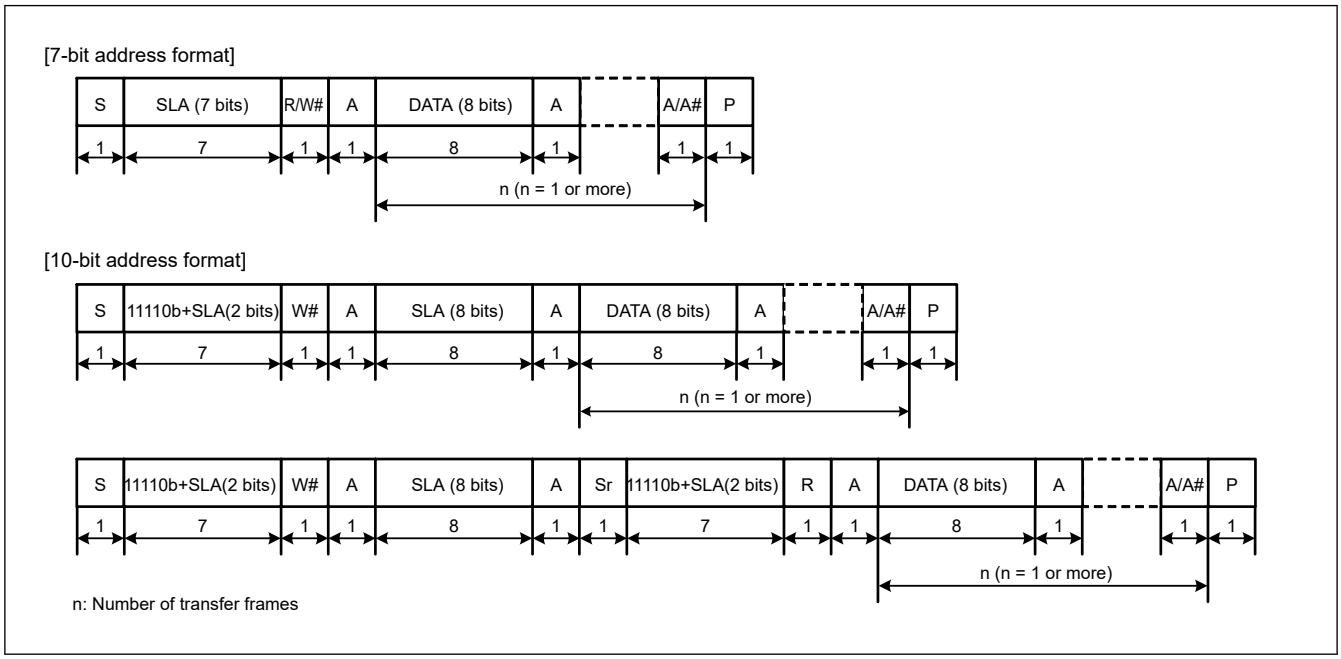
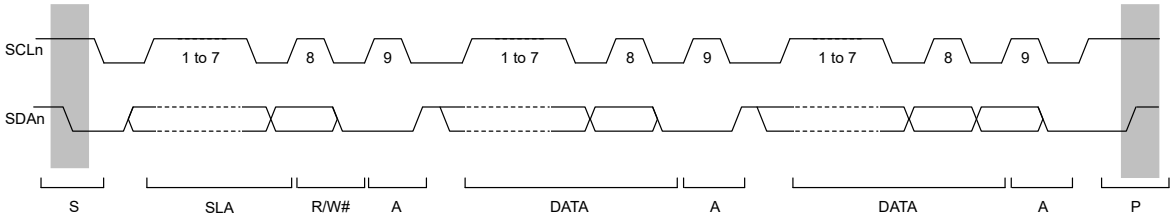


Figure 32.3 I2C bus format



Note: n = 0, 1

Figure 32.4 I²C bus timing when the SLA setting = 7 bits

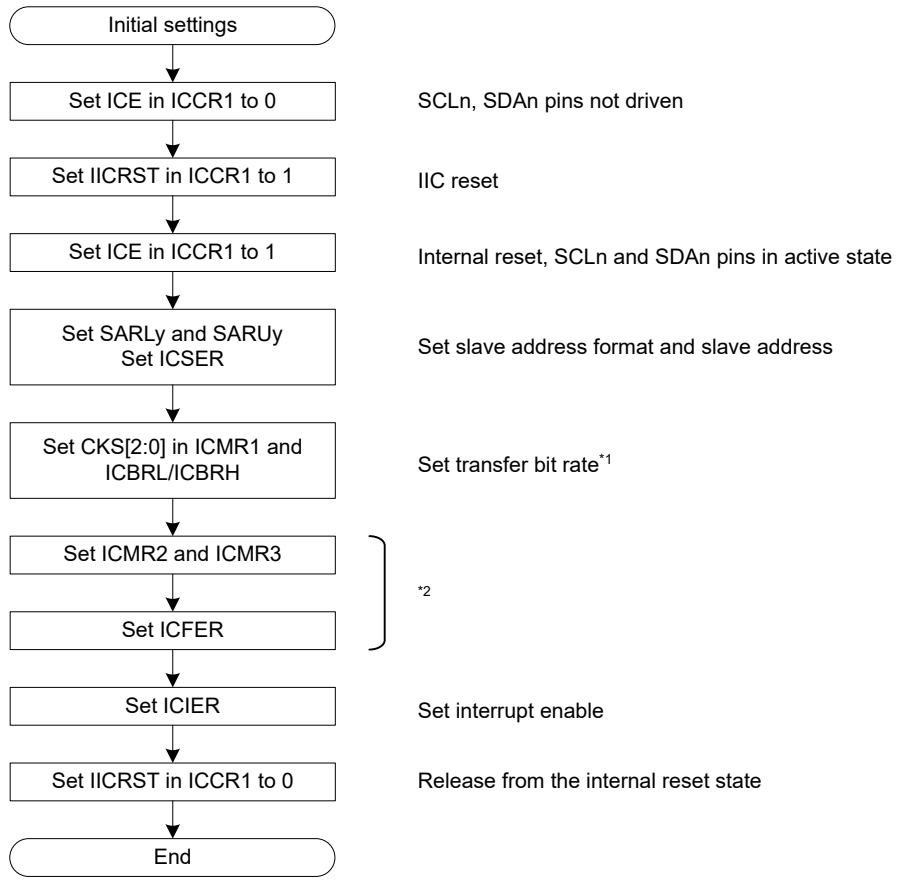
S:	Start condition. The master device drives the SDAn line low from high while the SCLn line is high.
SLA:	Slave address, by which the master device selects a slave device.
R/W#:	Indicates the direction of data transfer: from the slave device to the master device when R/W# is 1, or from the master device to the slave device when R/W# is 0.
A:	Acknowledge. The receive device drives the SDAn line low. In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.
A#:	Not Acknowledge. The receive device drives the SDAn line high.
Sr:	Restart condition. The master device drives the SDAn line low from the high level after the setup time has elapsed with the SCLn line high.
DATA:	Transmitted or received data.
P:	Stop condition. The master device drives the SDAn line high from low while the SCLn line is high.

32.3.2 Initial Settings

Before starting data transmission or reception, initialize the IIC using the procedure shown in Figure 32.5.

1. Set the ICCR1.ICE bit to 0 to set the SCLn and SDAn pins to the inactive state.
2. Set the ICCR1.IICRST bit to 1 to initiate IIC reset.
3. Set the ICCR1.ICE bit to 1 to initiate internal reset.
4. Set the SARLy, SARUy, ICSER, ICMR1, ICBRH, and ICBRL registers (y = 0 to 2), and set the other registers as required. For initial settings of the IIC, see Figure 32.5.
5. When the required register settings are complete, set the ICCR1.IICRST bit to 0 to release the IIC reset.

This procedure is not required if the IIC initialization is already complete.



Note: n = 0, 1, y = 0 to 2
Note 1. When the IIC is used only in slave mode, set the ICBRL register to a value longer than the data setup time.
Note 2. Set these registers as required.

Figure 32.5 Example IIC initialization flow

32.3.3 Master Transmit Operation

In master transmit operation, the IIC outputs the SCL clock and transmitted data signals as the master device, and the slave device returns acknowledgments. Figure 32.6 shows an example of master transmission, and Figure 32.7 to Figure 32.9 show the operation timing in master transmission.

To set up and perform master transmission:

1. Process initial settings. For details, see section 32.3.2. Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is free, and then set the ST bit in ICCR2 to 1 (start condition request). On receiving the request, the IIC issues a start condition. At the same time, the BBSY and START flags in ICSR2 automatically set to 1 and the ST bit automatically is set to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDAn line match while the ST bit is 1, the IIC recognizes that issuance of the start condition as requested by the ST bit is successfully complete, and the MST and TRS bits in ICCR2 automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 also automatically is set to 1 in response to the setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. When the transmit data is written to ICDRT, the TDRE flag automatically is set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag again is set to 1. After the byte containing the slave address and R/W# bit is transmitted, the value of the TRS bit automatically updates to select master transmit or master receive mode based on the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the IIC continues in master transmit mode.

Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.
For data transmission with an address in the 10-bit format, start by writing 11110b, the 2 higher-order bits of the slave address, and W to ICDRT as the first address transmission. For the second address transmission, write the 8 lower-order bits of the slave address to ICDRT.

4. Check that the TDRE flag in ICSR2 is 1, and then write the transmit data to the ICDRT register. The IIC automatically holds the SCLn line low until the transmit data is ready or a stop condition is issued.
5. After all bytes of transmit data are written to the ICDRT register, wait until the value in the TEND flag in ICSR2 returns to 1, and then set the SP bit in ICCR2 to 1 (stop condition requested). On receiving a stop condition request, the IIC issues the stop condition. Regarding issuing a stop condition, see section 32.11.3. Issuing a Stop Condition.
6. On detecting the stop condition, the IIC automatically sets the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Additionally, it automatically sets the TDRE and TEND flags to 0, and sets the STOP flag in ICSR2 to 1.
7. Check that the ICSR2.STOP flag is 1, and then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

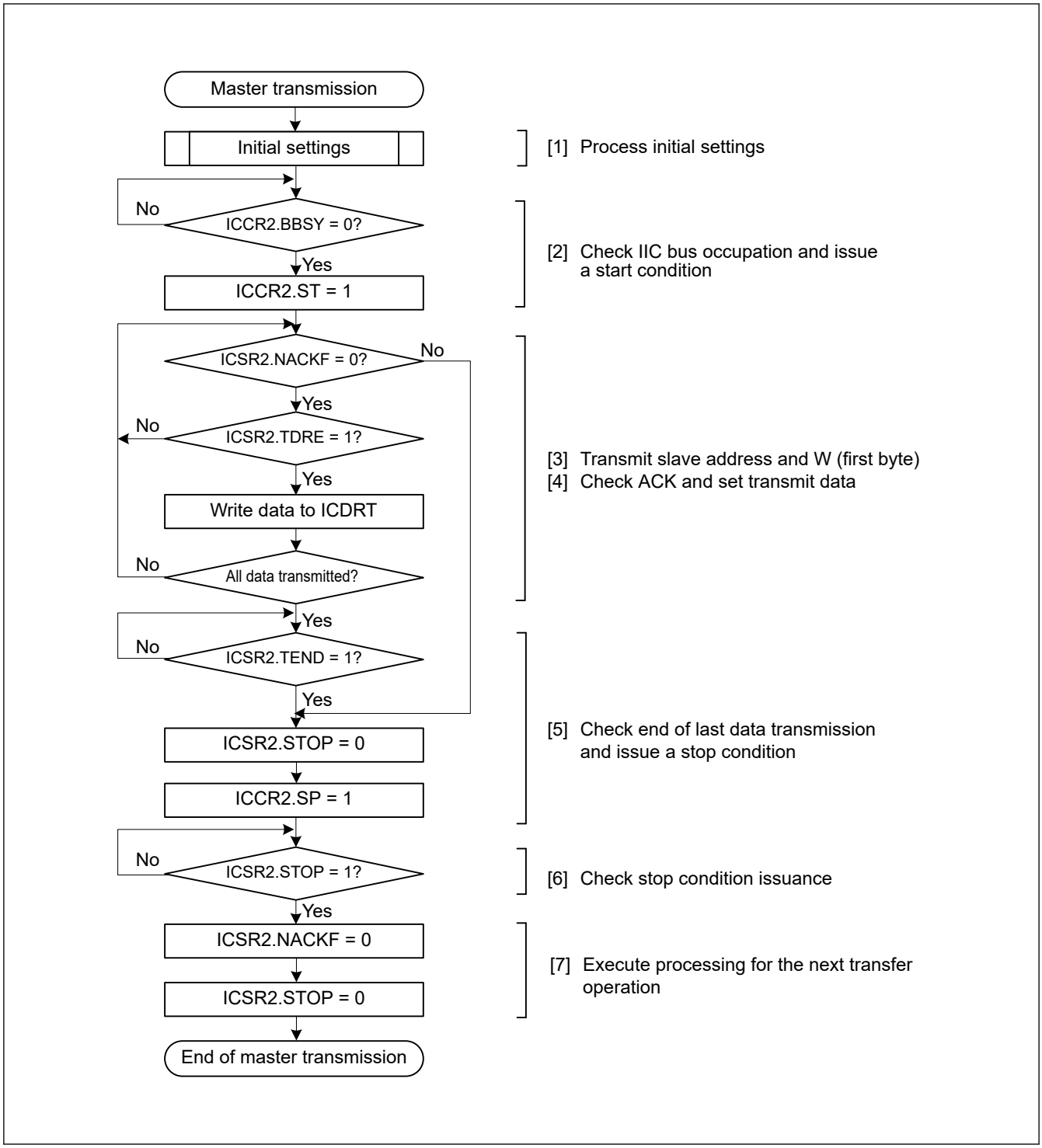


Figure 32.6 Example master transmission flow

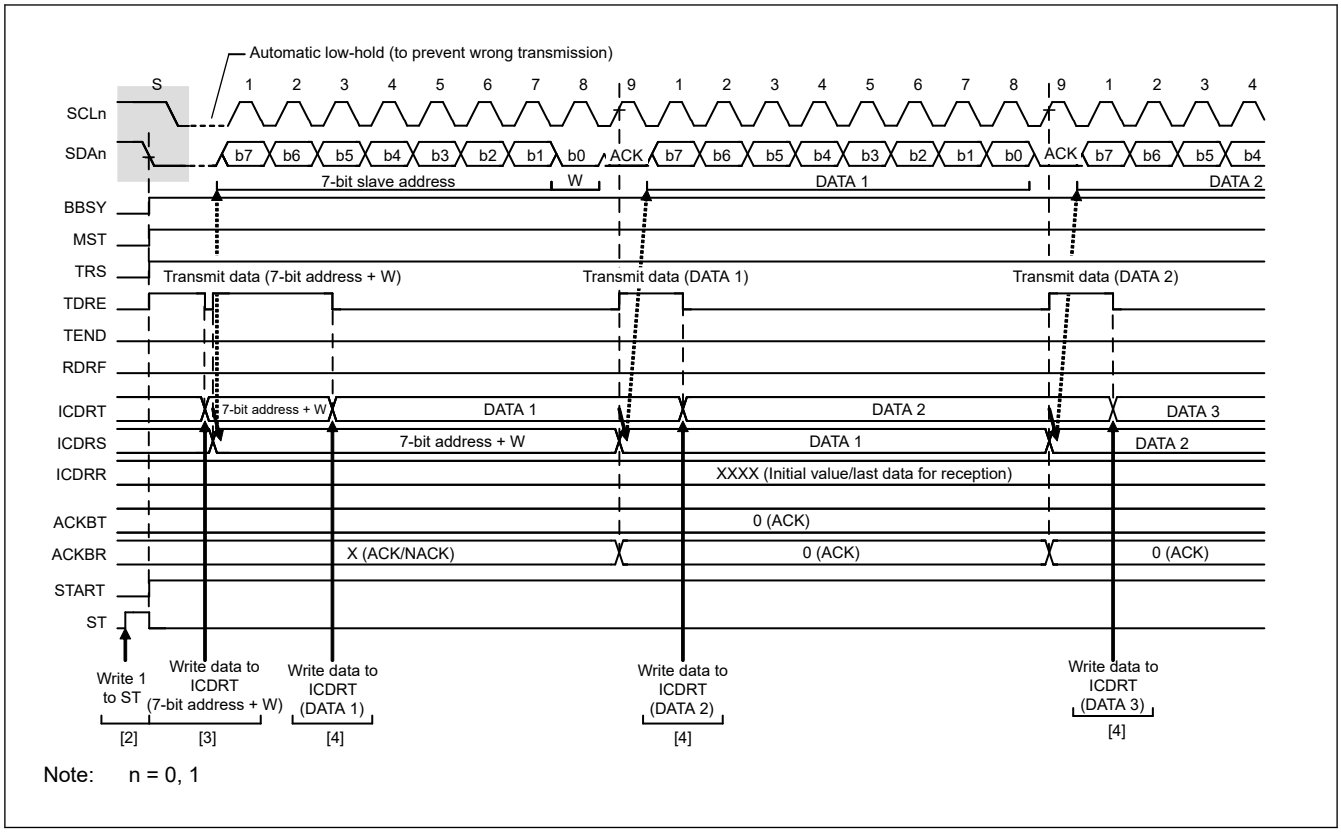


Figure 32.7 Master transmit operation timing (1) with 7-bit address format

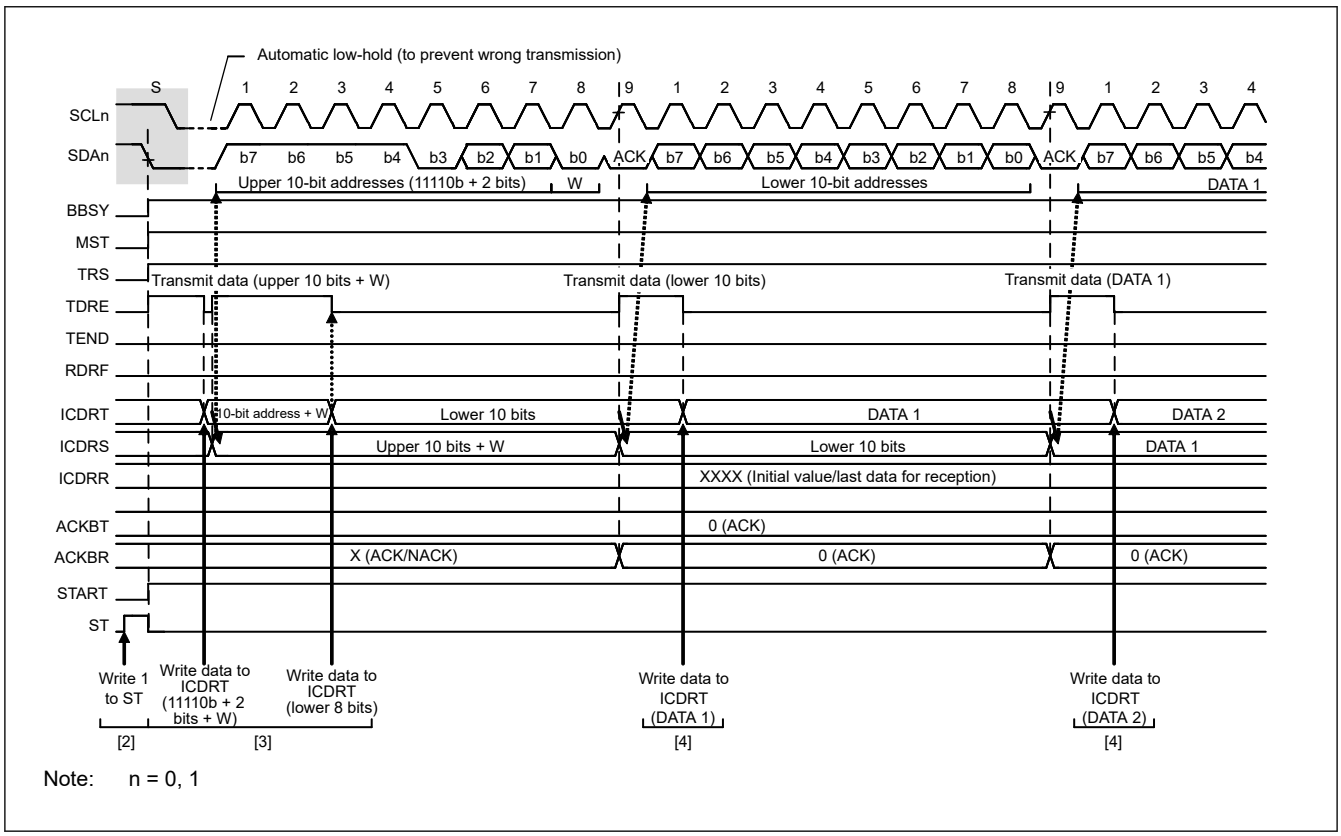
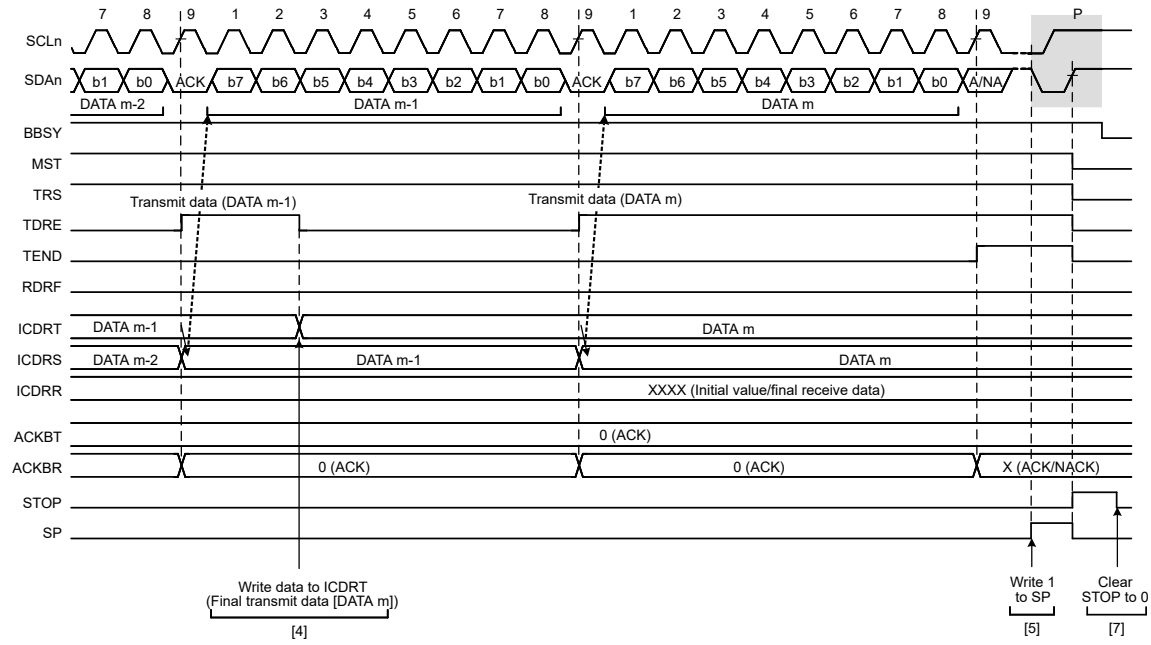


Figure 32.8 Master transmit operation timing (2) with 10-bit address format



Note: n = 0, 1

Figure 32.9 Master transmit operation timing (3)

32.3.4 Master Receive Operation

In master receive operation, the IIC as a master device outputs the SCL clock, receives data from the slave device, and returns acknowledgments. Because the IIC must start by sending a slave address to the associated slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.

Figure 32.10 and Figure 32.11 show examples of master reception (7-bit address format), and Figure 32.12 to Figure 32.14 show the operation timing in master reception.

To set up and perform master reception:

1. Process initial settings. For details, see section 32.3.2. Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is free, and then set the ST bit in ICCR2 to 1 (start condition request). On receiving the request, the IIC issues a start condition. When the IIC detects the start condition, the BBSY and START flags in ICSR2 automatically set to 1, and the ST bit automatically is set to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDAn line match while the ST bit is 1, the IIC recognizes that issuance of the start condition as requested by the ST bit is successfully complete, and the MST and TRS bits in ICCR2 automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 also automatically is set to 1 in response to the setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to ICDRT. When the transmit data is written to ICDRT, the TDRE flag automatically is set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag again is set to 1. When the byte containing the slave address and R/W# bit is transmitted, the value of the ICCR2.TRS bit automatically updates to select transmit or receive mode based on the value of the transmitted R/W# bit. If the value of the R/W# bit is 1, the TRS bit is set to 0 on the rising edge of the ninth cycle of the SCL clock, placing the IIC in master receive mode. At this time, the TDRE flag is set to 0 and the ICSR2.RDRF flag automatically is set to 1.
   Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.
   For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue a restart condition. After that, transmitting 11110b, the two higher-order bits of the slave address, and the R bit places the IIC in master receive mode.

4. Dummy read ICDRR after confirming that the RDRF flag in ICSR2 is 1. This makes the IIC start output of the SCL clock and start data reception.
5. After 1 byte of data is received, the RDRF flag in ICSR2 is set to 1 on the rising edge of the 8th or 9th cycle of the SCL clock, as selected in the RDRFS bit in ICMR3. Reading ICDRR at this time produces the received data, and the RDRF flag is automatically set to 0 at the same time. Additionally, the value of the acknowledgment field received during the ninth cycle of the SCL clock is returned as the value set in the ICMR3.ACKBT bit. If the next byte to be received is the second-to-last byte, set the ICMR3.WAIT bit to 1 for wait insertion before reading ICDRR, containing the second-to-last byte. In addition to enabling NACK output, even when interrupts or other operations result in delays in setting the ICMR3.ACKBT bit to 1 (NACK) in step (6), this fixes the SCLn line to the low level on the rising edge of the ninth clock cycle in reception of the last byte, which enables the issuing of a stop condition.
6. When the ICMR3.RDRFS bit is 0, and the slave device must be notified that it is to end transfer for data reception after transfer of the next and final byte, set the ICMR3.ACKBT bit to 1 (NACK).
7. After reading the second-to-last byte from the ICDRR register, if the value of the ICSR2.RDRF flag is 1, write 1 to the SP bit in ICCR2 (stop condition requested), and then read the last byte from ICDRR. When ICDRR is read, the IIC is released from the wait state and issues the stop condition after low-level output in the ninth clock cycle is complete or the SCLn line is released from the low-hold state.
8. On detecting the stop condition, the IIC automatically sets the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Additionally, detection of the stop condition sets the ICSR2.STOP flag to 1.
9. Check that the ICSR2.STOP flag is 1, then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

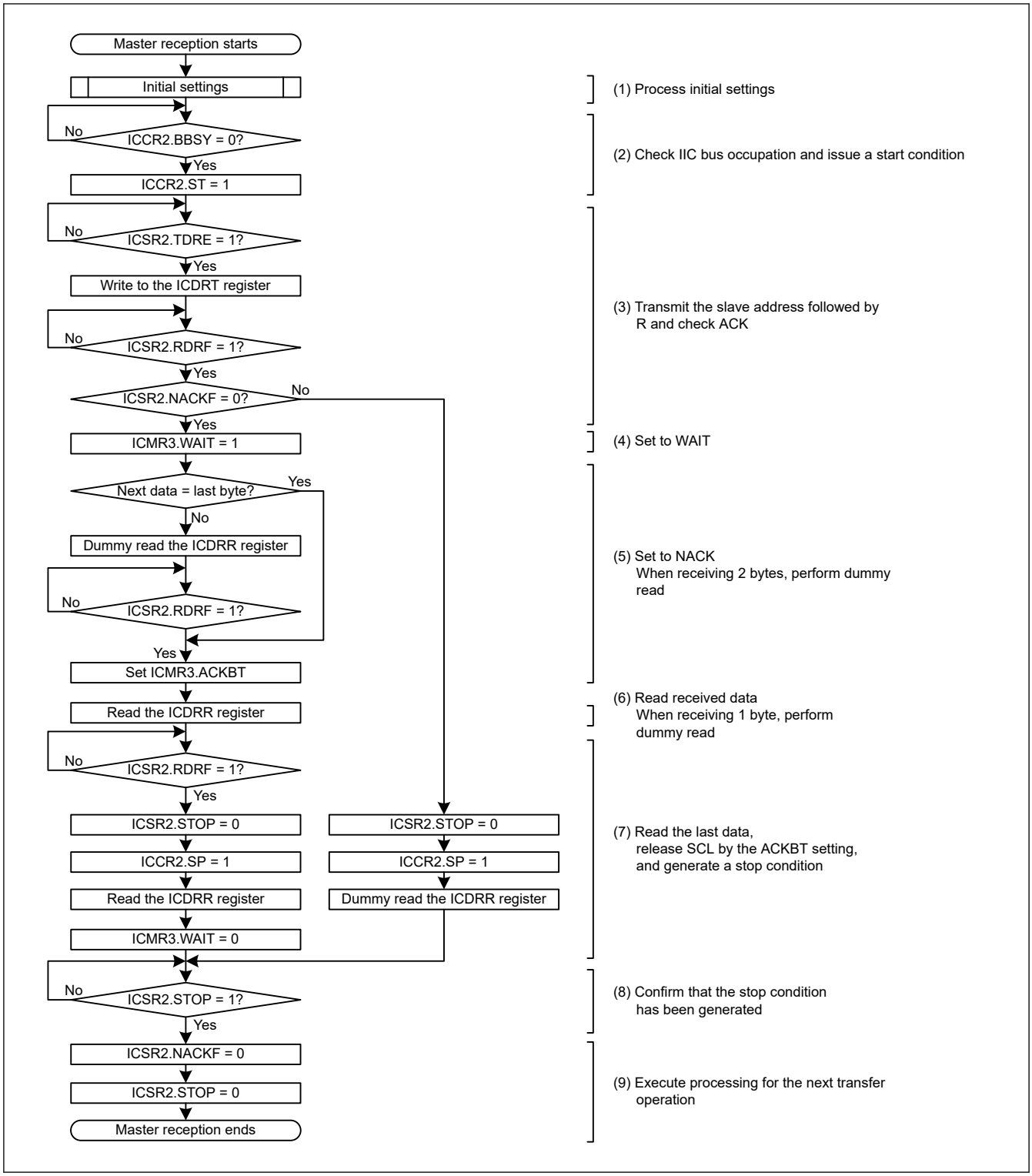


Figure 32.10 Example master reception flow with 7-bit address format of 1 byte or 2 bytes

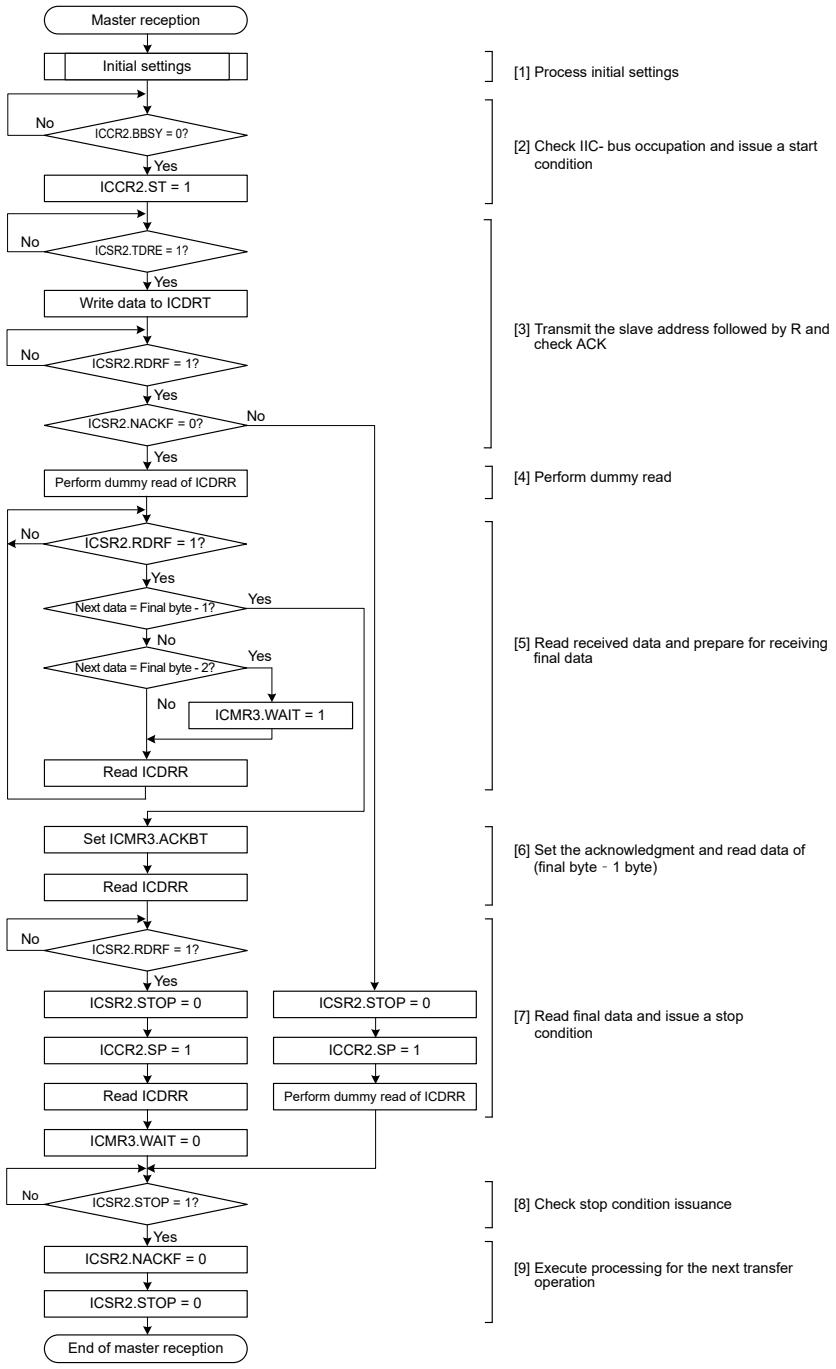
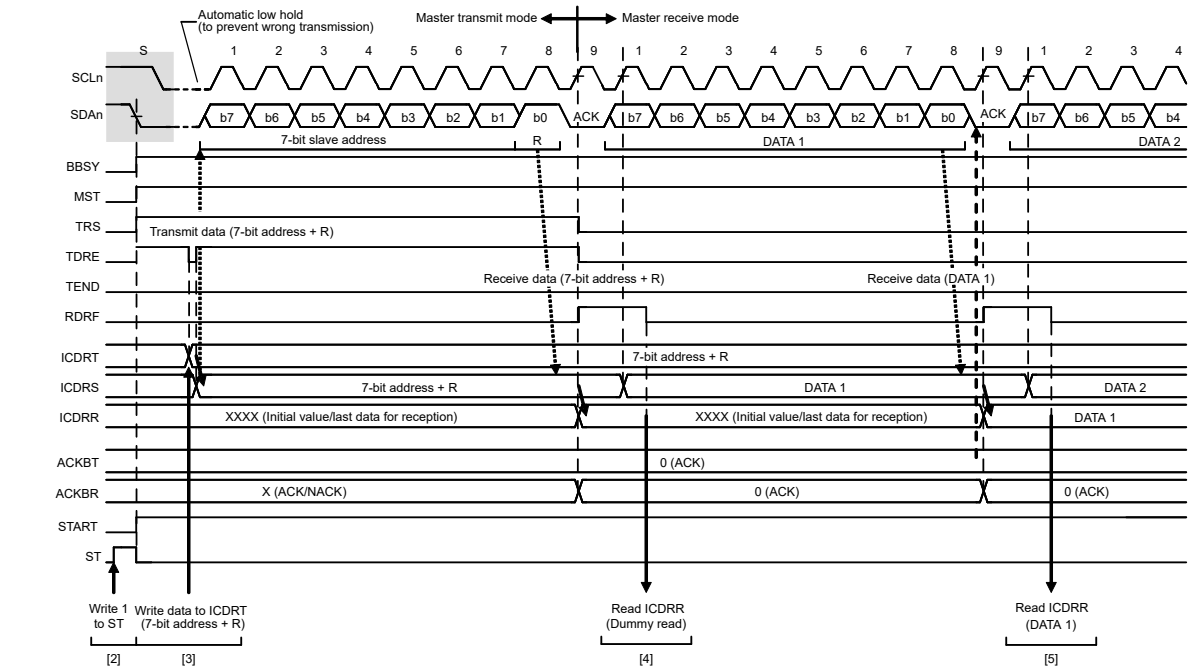
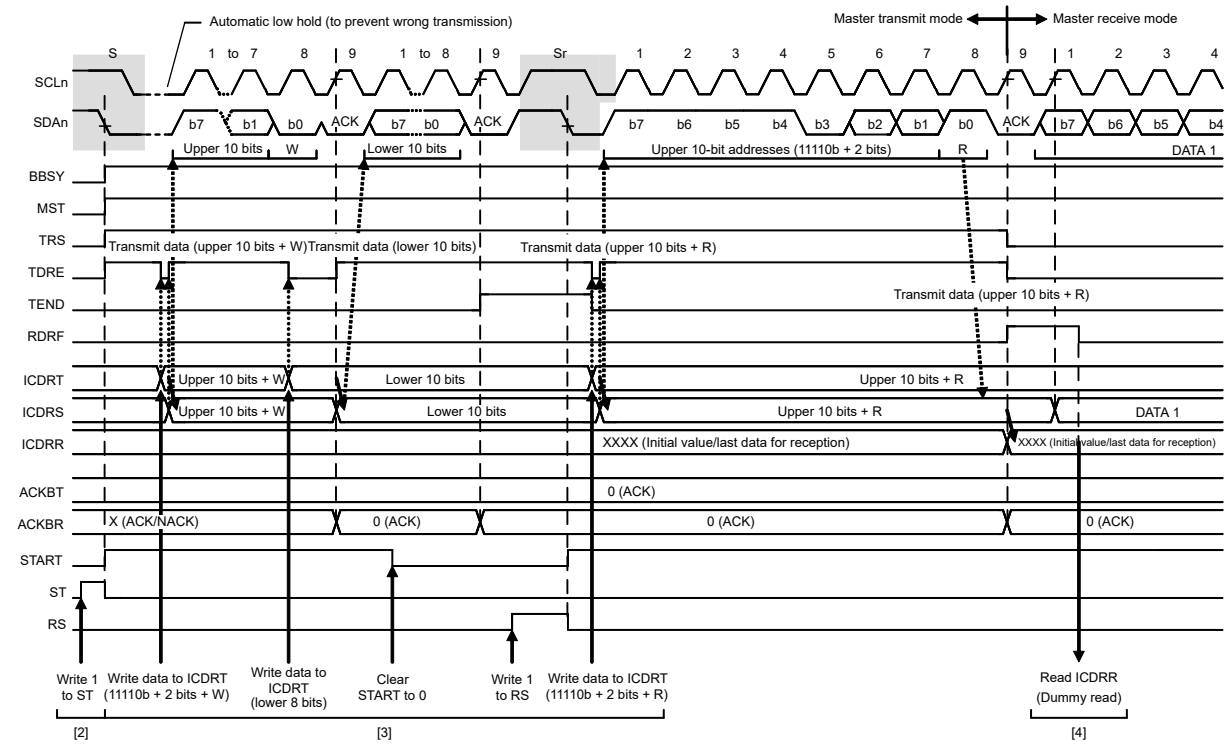


Figure 32.11 Example master reception flow with 7-bit address format of 3 or more bytes



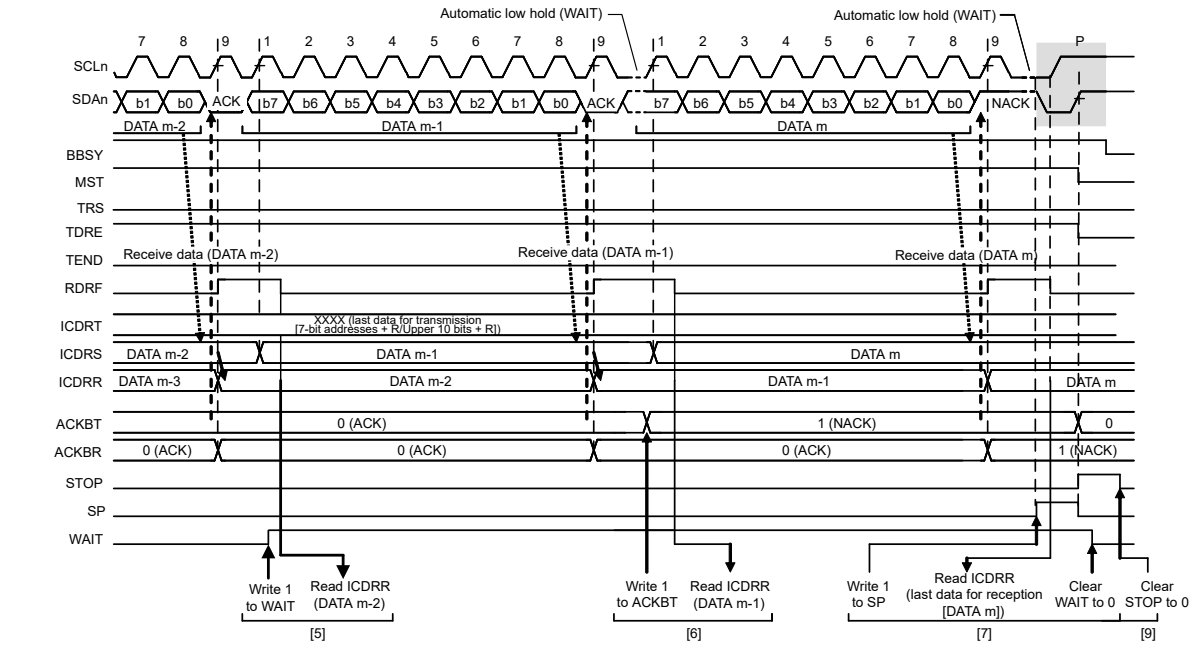
Note: n = 0, 1

Figure 32.12 Master receive operation timing (1) with 7-bit address format when RDRFS = 0



Note: n = 0, 1

Figure 32.13 Master receive operation timing (2) with 10-bit address format when RDRFS = 0



Note: n = 0, 1

Figure 32.14 Master receive operation timing (3) when RDRFS = 0

32.3.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL clock, the IIC transmits data as a slave device, and the master device returns acknowledgments.

Figure 32.15 shows an example of slave transmission, and Figure 32.16 and Figure 32.17 show the operation timing in slave transmission.

To set up and perform slave transmission:

1. Initialize the IIC using the procedure in section 32.3.2. Initial Settings.
   After initialization, the IIC stays in the standby state until it receives a slave address that matches.
2. After receiving a matching slave address, the IIC sets one of the associated ICSR1.HOA, GCA, and AASy flags (y = 0 to 2) to 1 on the rising edge of the ninth cycle of the SCL clock and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of the SCL clock. If the value of the R/W# bit is 1, the IIC automatically places itself in slave transmit mode by setting both the ICCR2.TRS bit and the ICSR2.TDRE flag to 1.
3. Check that the ICSR2.TDRE flag is 1, then write the transmit data to the ICDRT register. If the IIC receives no acknowledge from the master device (receives an NACK signal) while the ICFER.NACKE bit is 1, the IIC suspends transfer of the next data.
4. Wait until the ICSR2.TEND flag is set to 1 while the ICSR2.TDRE flag is 1, after the ICSR2.NACKF flag is set to 1 or the last byte for transmission is written to the ICDRT register. When the ICSR2.NACKF flag or the TEND flag is 1, the IIC drives the SCLn line low on the ninth falling edge of the SCL clock.
5. When the ICSR2.NACKF or ICSR2.TEND flag is 1, dummy read ICDRR to complete the processing. This releases the SCLn line.
6. On detecting the stop condition, the IIC automatically sets the ICSR1.HOA, GCA, and AASy flags (y = 0 to 2), the ICSR2.TDRE and TEND flags, and the ICCR2.TRS bit to 0, and enters slave receive mode.
7. Check that the ICSR2.STOP flag is 1, then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

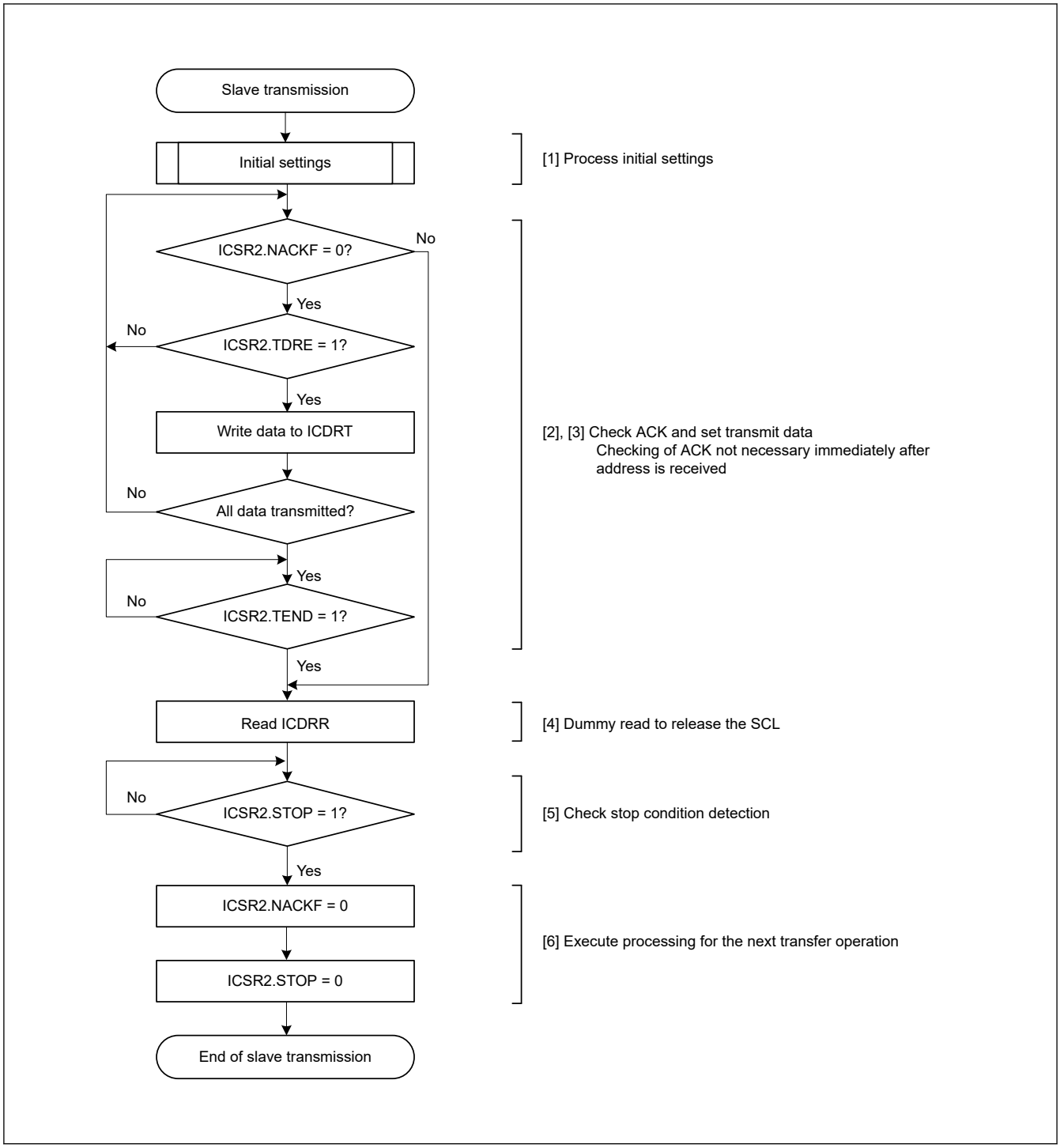
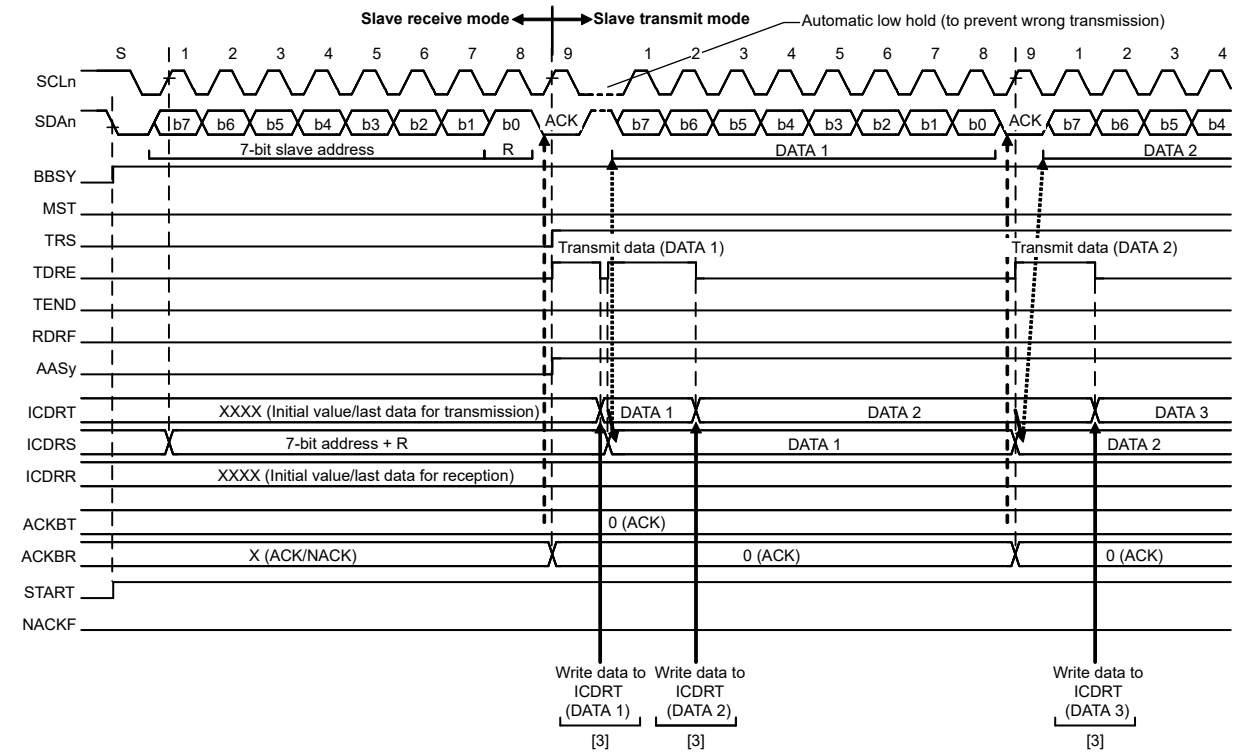
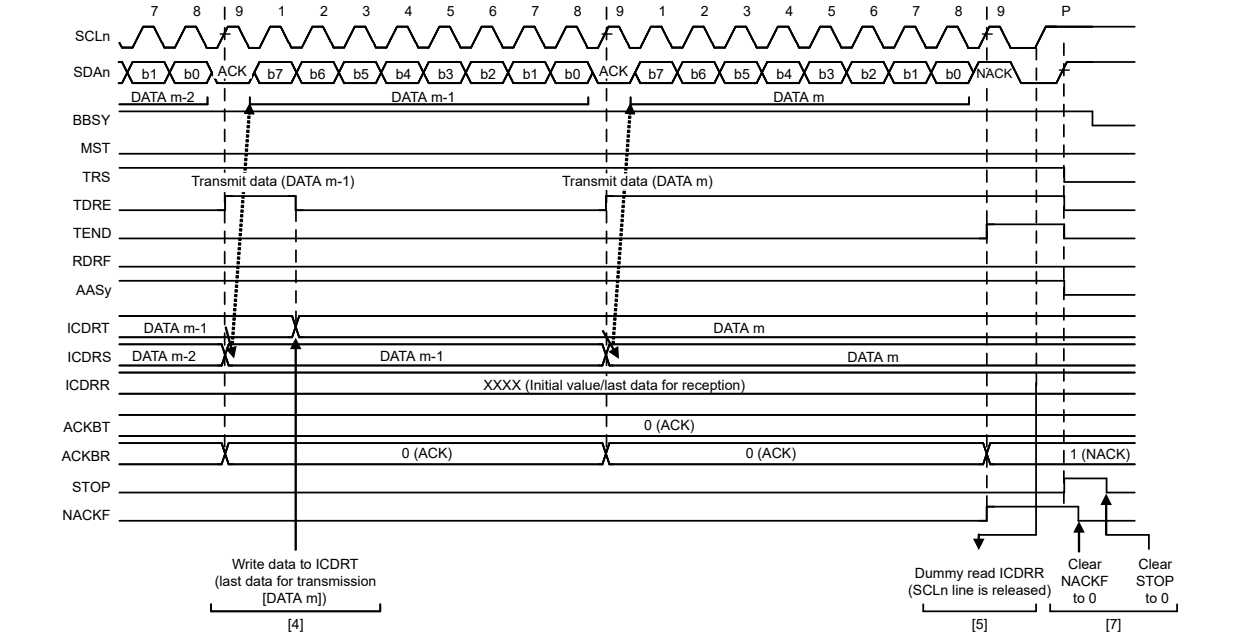


Figure 32.15 Example slave transmission flow



Note: n = 0, 1, y = 0 to 2

Figure 32.16 Slave transmit operation timing (1) with 7-bit address format



Note: n = 0, 1, y = 0 to 2

Figure 32.17 Slave transmit operation timing (2)

32.3.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the IIC returns acknowledgments as a slave device.

Figure 32.18 shows an example of slave reception, and Figure 32.19 and Figure 32.20 show the operation timing in slave reception.

To set up and perform slave reception:

1. Initialize the IIC using the procedure in section 32.3.2. Initial Settings.
   After initialization, the IIC stays in the standby state until it receives a slave address that matches.
2. After receiving a matching slave address, the IIC sets one of the associated ICSR1.HOA, GCA, and AASy flags (y = 0 to 2) to 1 on the rising edge of the ninth cycle of the SCL clock and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of the SCL clock. If the value of the R/W# bit is 0, the IIC continues to place itself in slave receive mode and sets the RDRF flag in ICSR2 to 1.
3. Check that the ICSR2.STOP flag is 0 and the ICSR2.RDRF flag is 1, then dummy read ICDRR. The dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8 bits when the 10-bit address format is selected.
4. When ICDRR is read, the IIC automatically sets the ICSR2.RDRF flag to 0. If reading of ICDRR is delayed and a next byte is received while the RDRF flag is still set to 1, the IIC holds the SCLn line low at the SCL clock one cycle before the point where the RDRF is set. In this case, reading ICDRR releases the SCLn line from being held low.
   When the ICSR2.STOP flag is 1 and the ICSR2.RDRF flag is also 1, read ICDRR until all the data is completely received.
5. On detecting the stop condition, the IIC automatically clears the ICSR1.HOA, GCA, and AASy flags (y = 0 to 2) to 0.
6. Check that the ICSR2.STOP flag is 1, then set the ICSR2.STOP flag to 0 for the next transfer operation.

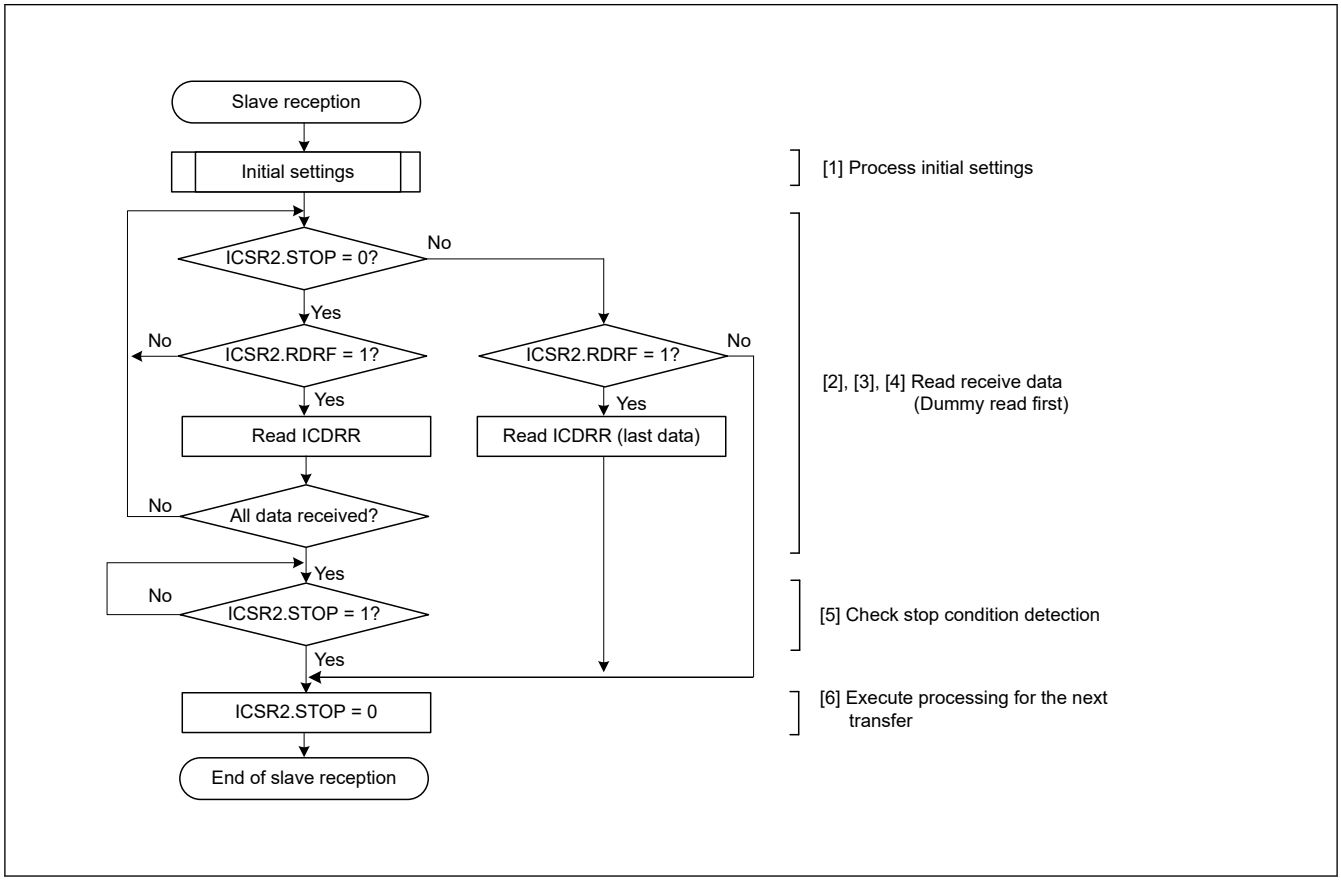
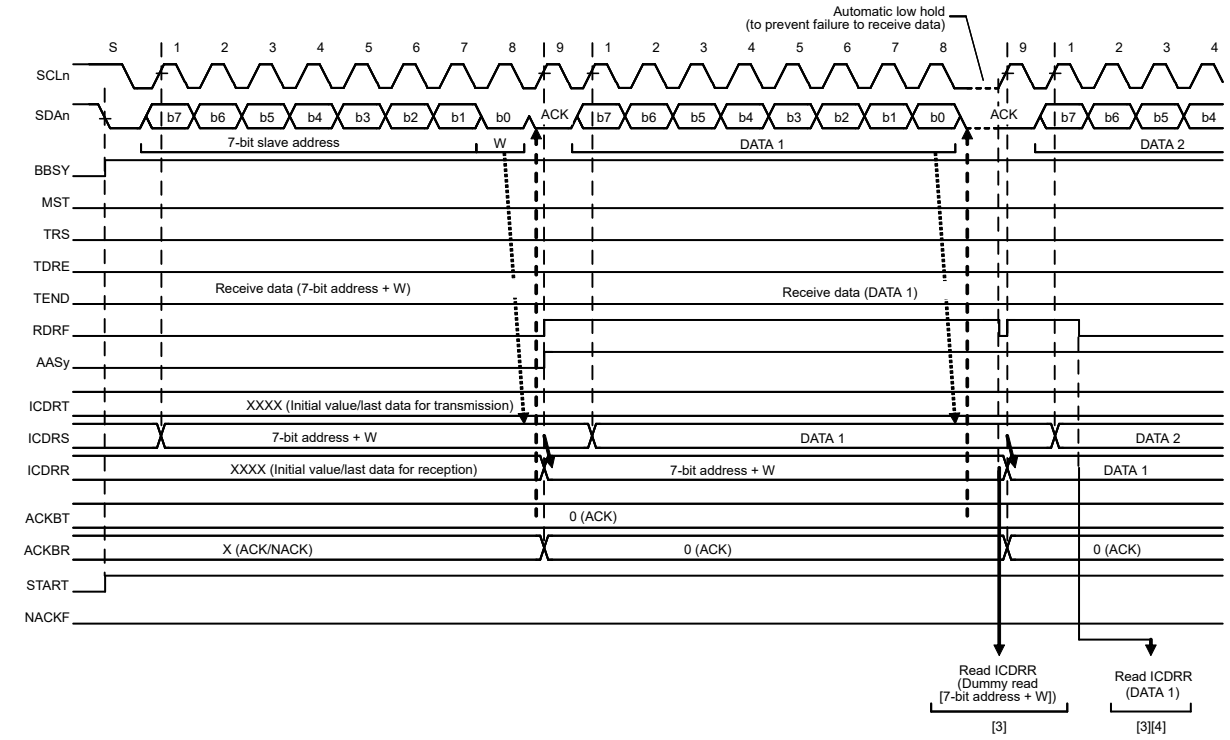
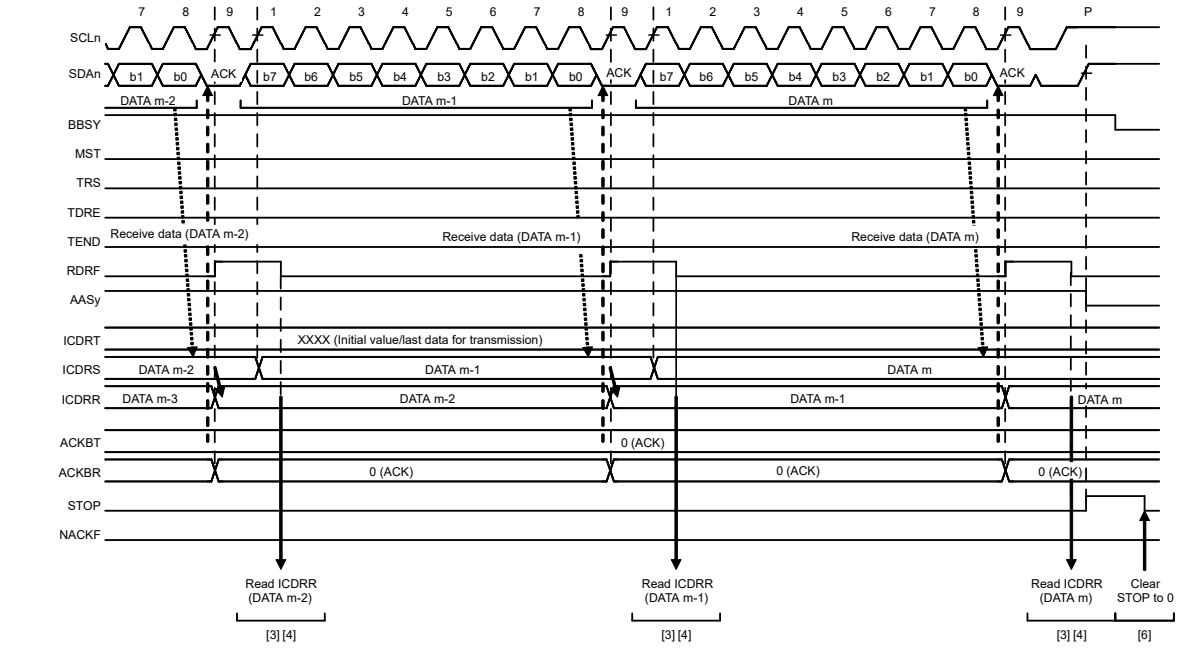


Figure 32.18 Example slave reception flow



Note: n = 0, 1, y = 0 to 2

Figure 32.19 Slave receive operation timing (1) with 7-bit address format when RDRFS = 0



Note: n = 0, 1, y = 0 to 2

Figure 32.20 Slave receive operation timing (2) when RDRFS = 0

32.4 SCL Synchronization Circuit

For generation of the SCL clock, the IIC starts counting the value for the high-level period specified in ICBRH when it detects a rising edge on the SCLn line, and it drives the SCLn line low when it completes counting. When the IIC detects the falling edge of the SCLn line, it starts counting the value for the low-level period specified in ICBRL, and then it stops driving the SCLn line, releasing the line, when it completes counting. The IIC repeats this process to generate the SCL clock.

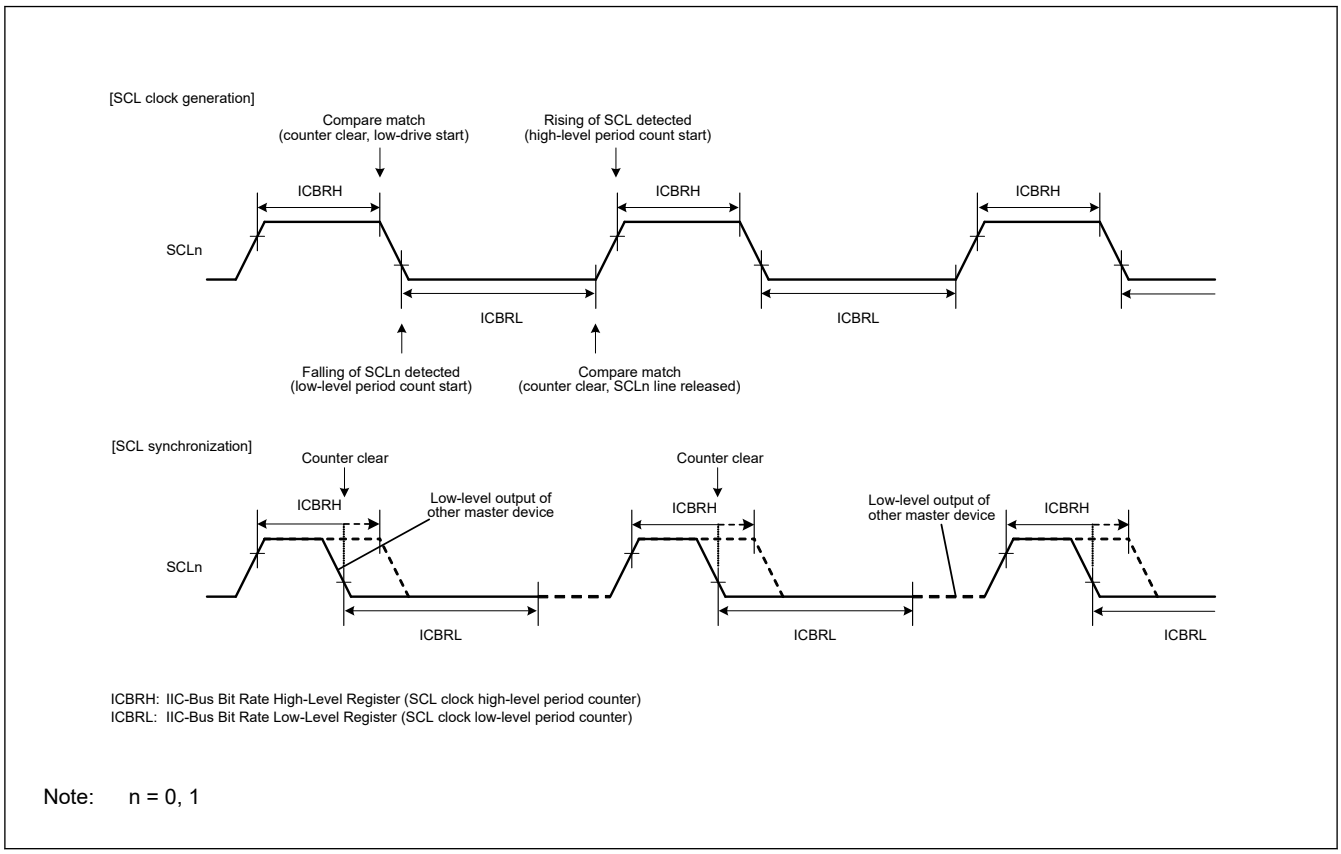
If multiple master devices are connected to the I^2C bus, a collision of SCL signals might arise because of contention with another master device. In such cases, the master devices must synchronize their SCL signals. Because this synchronization of SCL signals must be bit-by-bit, the IIC is equipped with an SCL synchronization circuit to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCLn line while in master mode.

When the IIC detects a rising edge on the SCLn line and starts counting the high-level period specified in ICBRH.BRH[4:0], and the level on the SCLn line falls because an SCL signal is being generated by another master device, the IIC performs the following:

1. Stops counting when it detects the falling edge.
2. Drives the level on the SCLn line low.
3. Starts counting the low-level period specified in ICBRL.BRL[4:0].

When the IIC finishes counting the low-level period, it stops driving the SCLn line low to release the line. If the low-level period of the SCL clock signal from the other master device is longer than the low-level period set in the IIC, the low-level period of the SCL signal is extended. When the low-level period for the other master device ends, the SCL signal rises because the SCLn line is released.

When the IIC finishes outputting the low-level period of the SCL clock, the SCLn line is released and the SCL clock rises. That is, when SCL signals from more than one master are contending, the high-level period of the SCL signal is synchronized with that of the clock with the narrower period, and the low-level period of the SCL signal is synchronized with that of the clock with the broader period. However, such synchronization of the SCL signal is only enabled when the SCLE bit in ICFER is set to 1.



Note: n = 0, 1

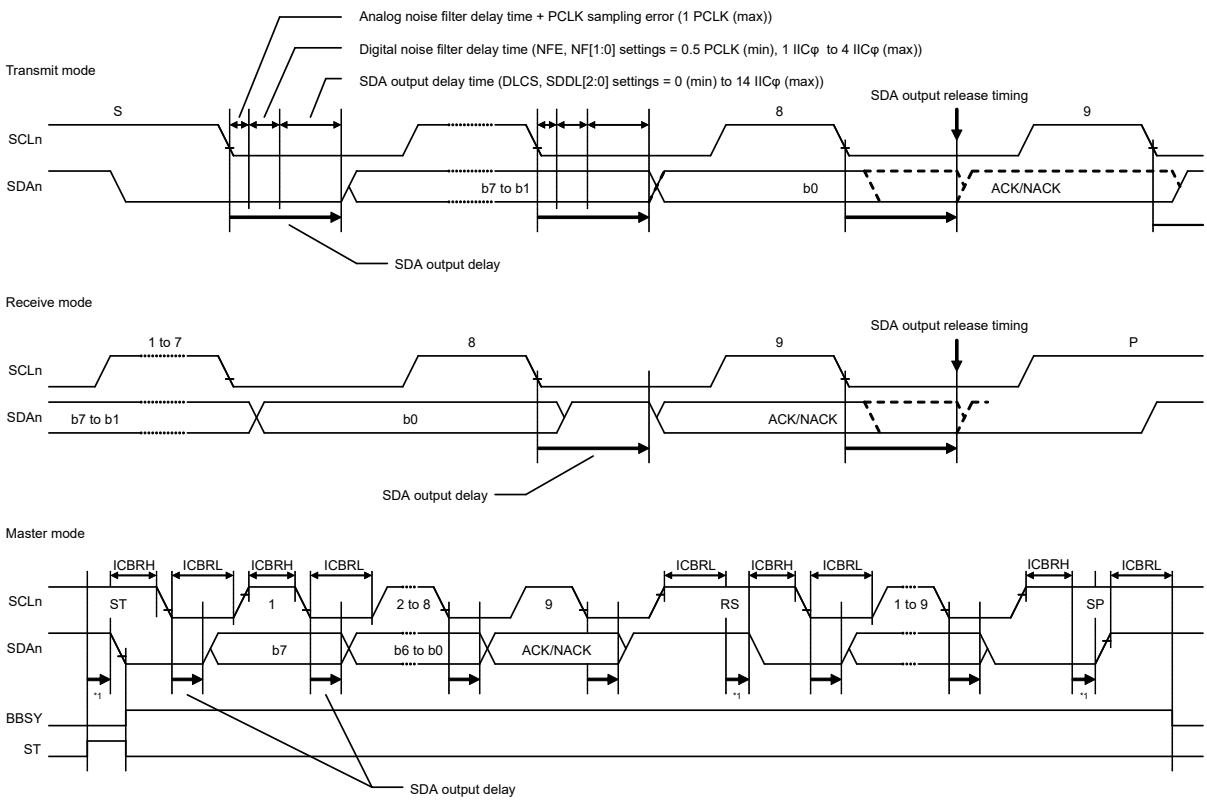
Figure 32.21 Generation and synchronization of SCL signal from IIC

32.5 SDA Output Delay Function

The IIC module incorporates a function for delaying output on the SDA line. The delay can be applied to all output on the SDA line, including issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals.

With this function, SDA output is delayed from the detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval during which the SCL clock is low. This approach helps prevent erroneous operation of communications devices, with the aim of satisfying the 300-ns minimum data-hold time requirement of the SMBus specification. The output delay function is enabled by setting the SDDL[2:0] bits in ICMR2 to a value other than 000b, and disabled by setting the same bits to 000b.

When the SDA output delay function is enabled, for example, the DLCS bit in ICMR2 selects the clock source for the SDA output delay counter, either as the internal base clock (IICφ) for the IIC module or as the internal base clock divided by 2 (IICφ/2). The counter counts the number of cycles set in the SDDL[2:0] bits in ICMR2. When the delay cycles count is reached, the IIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.



Note: PCLK = PCLKB
Note: n = 0, 1
Note 1. The output delay function is set by the DLCS and SDDL[2:0] bits when a start (ST), restart (RS), or stop (SP) condition is issued.

Figure 32.22 SDA output delay function

32.6 Digital Noise Filter Circuits

The internal circuitry sees the states of the SCLn and SDAn pins through analog and digital noise-filter circuits. Figure 32.23 shows a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of the IIC consists of four flip-flop circuit stages connected in series and a match-detection circuit. The number of valid stages in the digital noise filter is selected in the NF[1:0] bits in ICMR3. The selected number of valid stages determines the noise-filtering capability as a period from 1 to 4 IICφ cycles.

The input signal to the SCLn pin (or SDAn pin) is sampled on falling edges of the IICφ signal. When the input signal level matches the output level of the number of valid flip-flop circuit stages as selected in the NF[1:0] bits in ICMR3, the signal level is seen in the subsequent stage. If the signal levels do not match, the previous value is saved.

If the ratio between the frequency of the internal operating clock (PCLKB) and the transfer rate is small, for example, if data transfer is 400 kbps with PCLKB = 4 MHz, the characteristics of the digital noise filter might lead to the elimination of required signals as noise. In such cases, it is possible to disable the digital noise-filter circuit by setting the ICFER.NFE bit to 0, and use only the analog noise filter circuit.

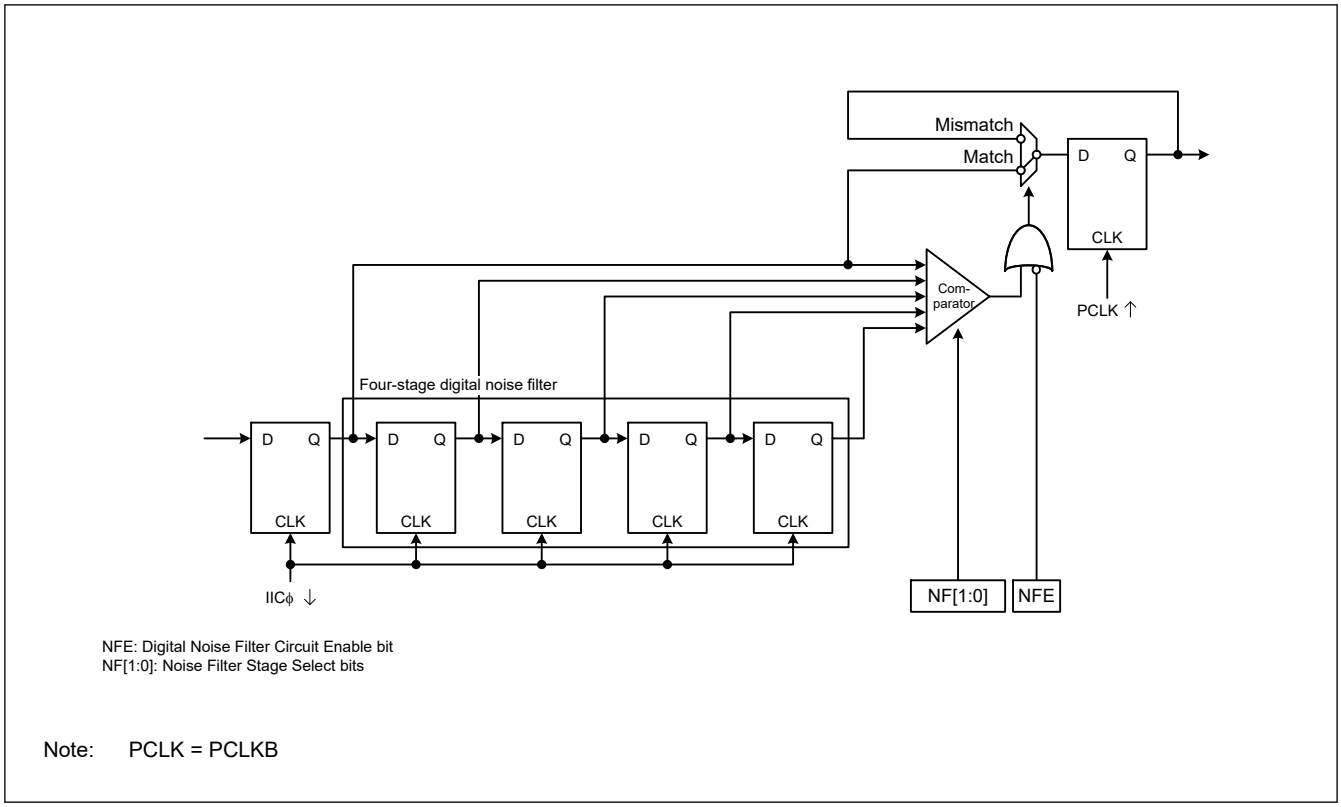


Figure 32.23 Digital noise filter circuit block diagram

32.7 Address Match Detection

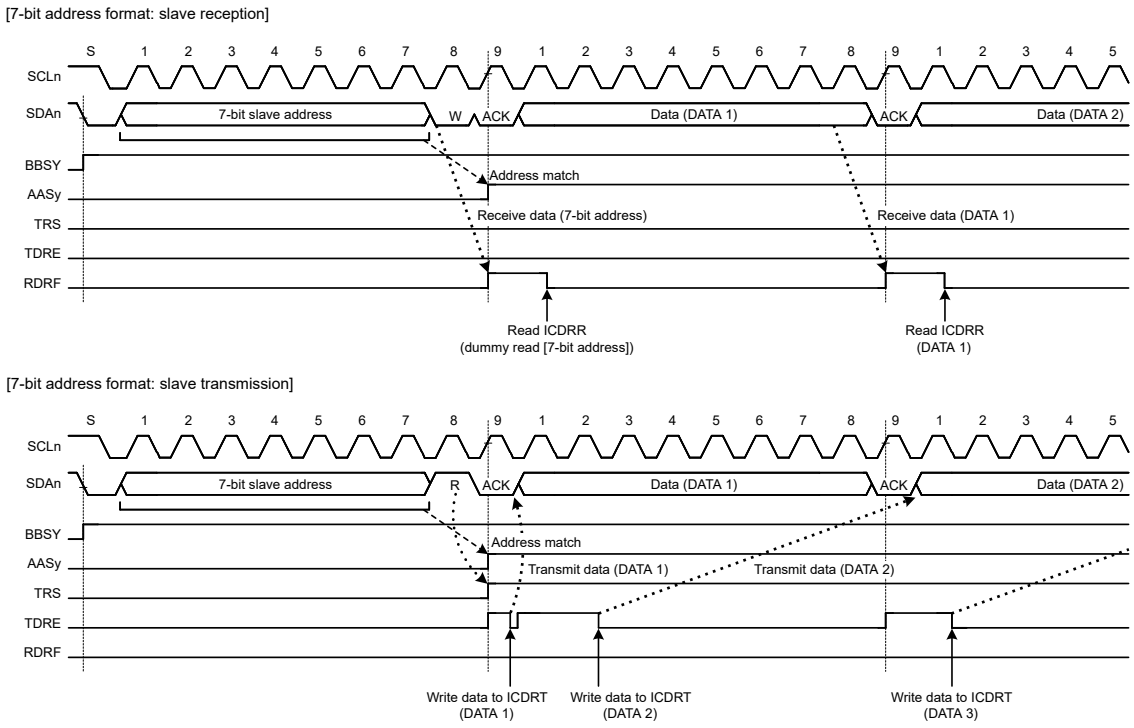
The IIC can set three unique slave addresses in addition to the general call address and host address. The slave addresses can be 7-bit or 10-bit slave addresses.

32.7.1 Slave-Address Match Detection

The IIC can set three unique slave addresses and has a slave address detection function for each unique slave address. When the SARyE bit (y = 0 to 2) in ICSER is set to 1, the slave addresses set in SARUy and SARLy (y = 0 to 2) can be detected.

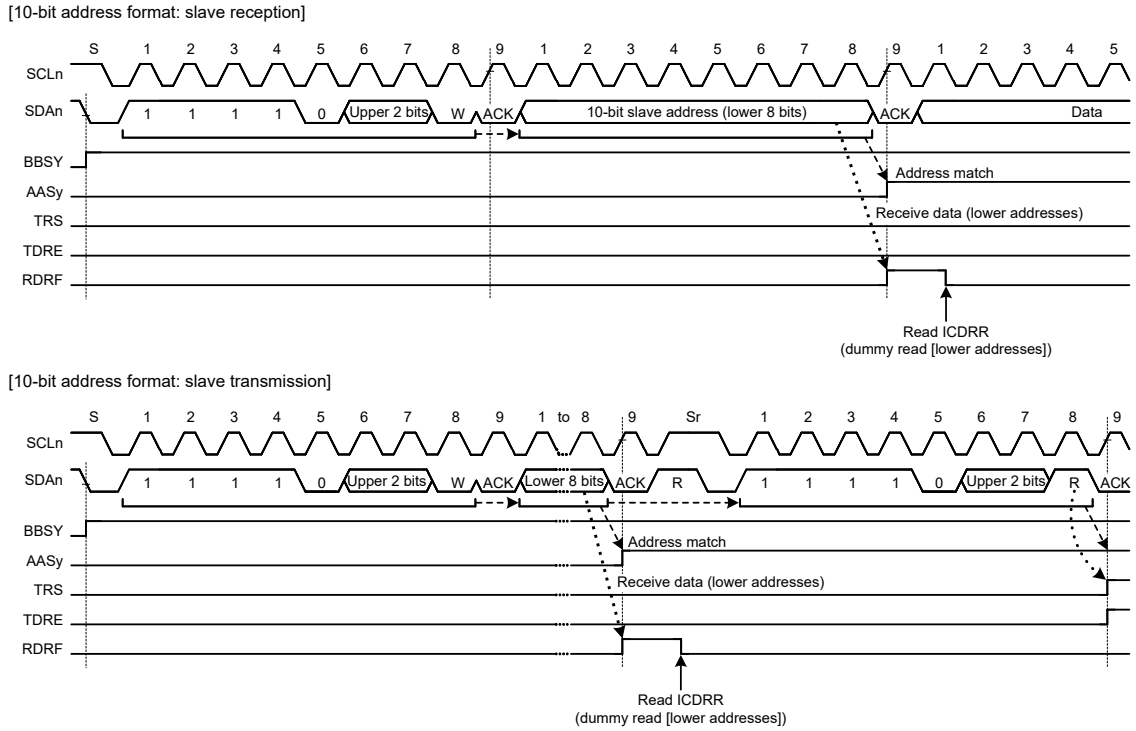
When the IIC detects a match of the set slave address, the associated AASy flag (y = 0 to 2) in ICSR1 is set to 1 on the rising edge of the ninth SCL clock cycle, and the RDRF flag in ICSR2 or the TDRE flag in ICSR2 is set to 1 by the subsequent R/W# bit. This causes a receive data full interrupt (IICn_RXI) or transmit data empty interrupt (IICn_TXI) to be generated. The AASy flag identifies which slave address is specified.

Figure 32.24 to Figure 32.26 show the AASy flag set timing in three cases.



Note: n = 0, 1, y = 0 to 2

Figure 32.24 AASy flag set timing with 7-bit address format



Note: n = 0, 1, y = 0 to 2

Figure 32.25 AASy flag set timing with 10-bit address format

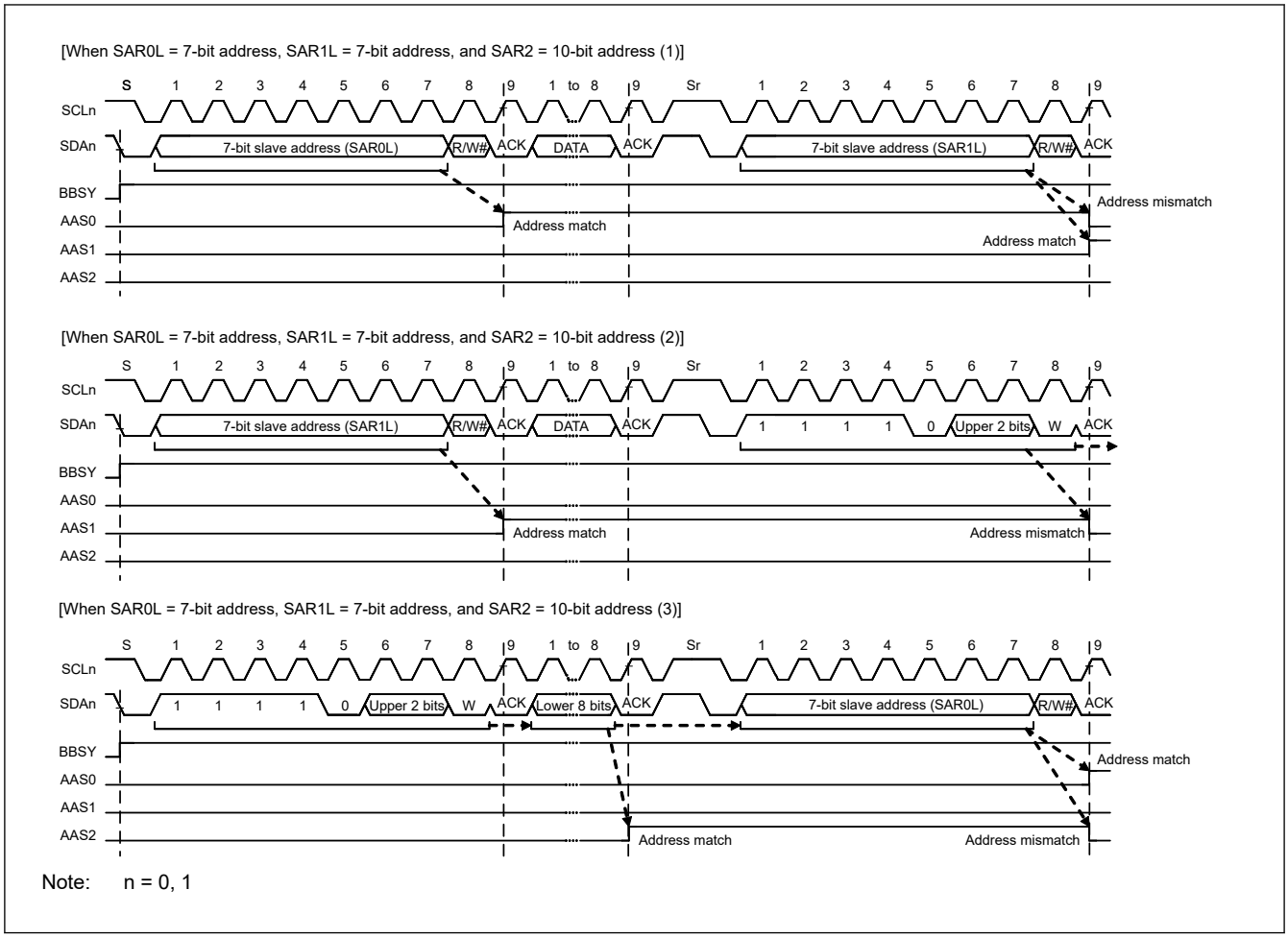


Figure 32.26 AASy flag set and clear timing with mixed 7-bit and 10-bit address formats

32.7.2 Detection of General Call Address

The IIC provides detection of the general call address (0000 000b + 0 [W]). This is enabled by setting the GCAE bit in ICSER to 1.

If the address received after a start or restart condition is issued is 0000 000b + 1[R] (start byte), the IIC recognizes this as the address of a slave device with an all-zero address, but not as the general call address.

When the IIC detects the general call address, both the GCA flag in ICSR1 and the RDRF flag in ICSR2 set to 1 on the rising edge of the ninth cycle of the SCL clock. This leads to the generation of a receive data full interrupt (IICn_RXI). The value of the GCA flag can be checked to confirm that the general call address was transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

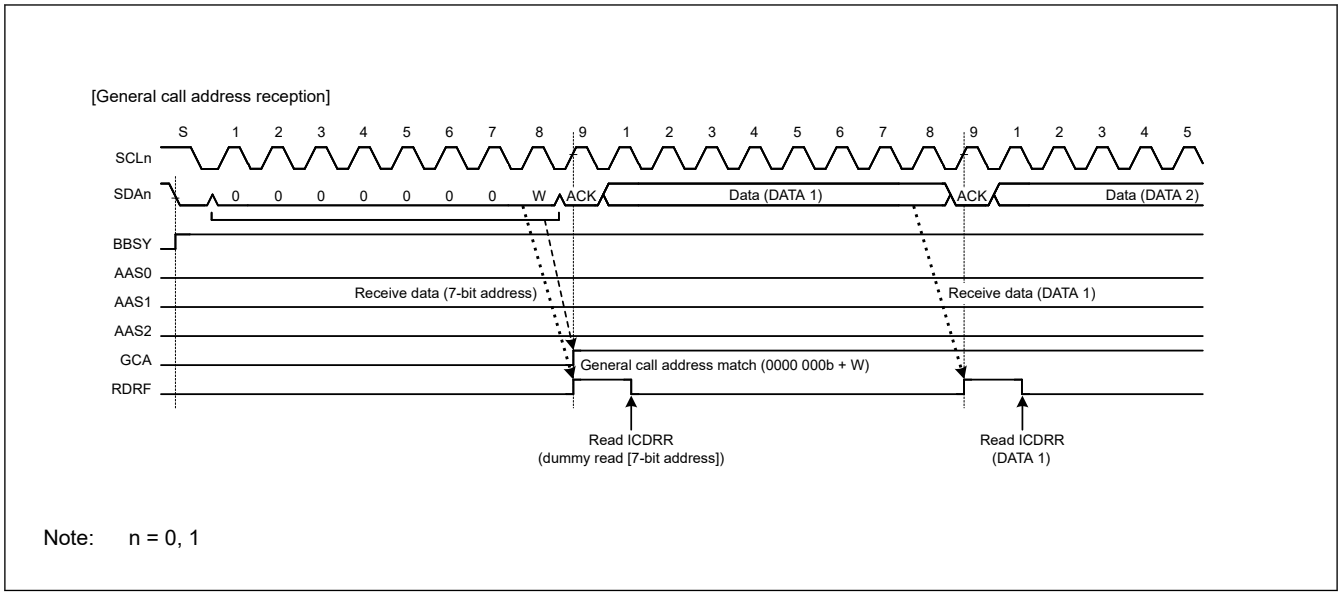


Figure 32.27 Timing of GCA flag setting during reception of general call address

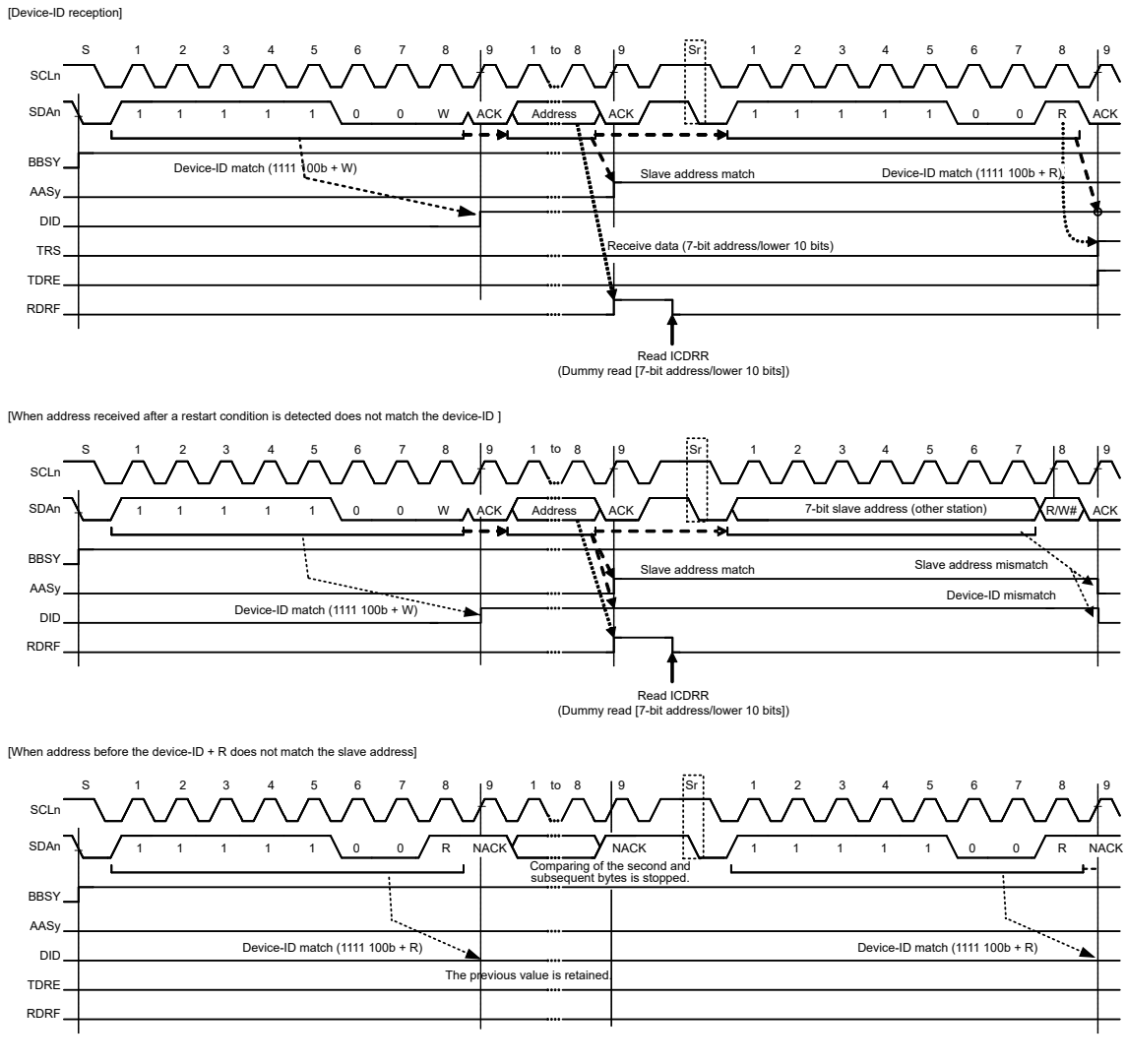
32.7.3 Device-ID Address Detection

The IIC module provides detection of device-ID address compliant with the I^2C bus specification (revision 03). When the IIC receives 1111 100b as the first byte after a start or restart condition is issued with the DIDE bit in ICSER set to 1, it recognizes the address as a device ID, sets the DID flag in ICSR1 to 1 on the rising edge of the ninth SCL clock cycle when the subsequent R/W# bit is 0, then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the IIC sets the associated AASy flag (y = 0 to 2) in ICSR1 to 1.

When the first byte received after the issue of a start or restart condition matches the device ID address (1111 100b) again and the subsequent R/W# bit is 1, the IIC does not compare the second and subsequent bytes and sets the ICSR2.TDRE flag to 1.

In the device ID address detection function, the IIC sets the DID flag to 0 if a match with the IIC slave address is not obtained or a match with the device ID address is not obtained after a match with the IIC slave address and a restart condition is not detected. If the first byte after detection of a start or restart condition matches the device ID address (1111 100b), and the R/W# bit is 0, the IIC sets the DID flag to 1 and compares the second and subsequent bytes with the slave address of the IIC. If the R/W# bit is 1, the DID flag holds the previous value and the IIC does not compare the second and subsequent bytes. Therefore, the reception of a device ID address can be checked by reading the DID flag after confirming that TDRE = 1.

Additionally, prepare the device ID fields (3 bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal transmit data. For details on the information that must be included in device ID fields, contact NXP Semiconductors.



Note: n = 0, 1, y = 0 to 2

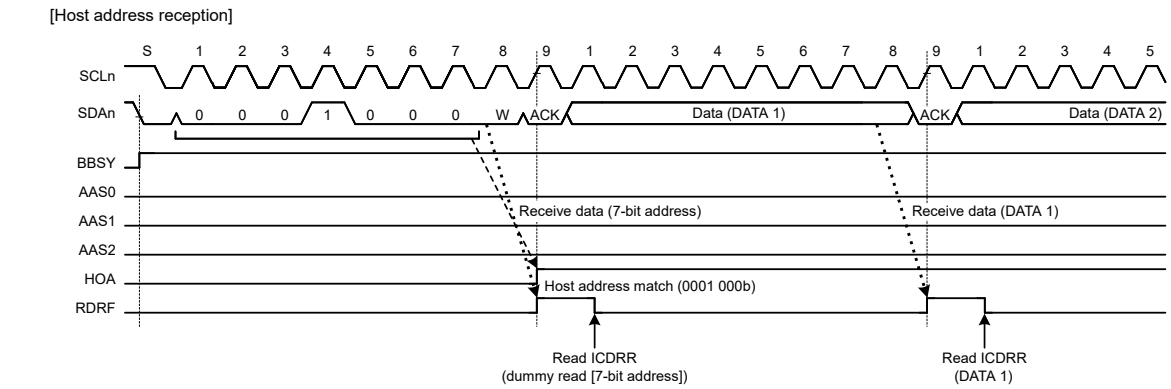
Figure 32.28 AASy and DID flag set and clear timing during reception of device ID

32.7.4 Host Address Detection

The IIC provides host address detection when operating in SMBus. When the HOAE bit in ICSER is set to 1 while the SMBS bit in ICMR3 is 1, the IIC can detect the host address (0001 000b) in slave receive mode (MST and TRS bits = 00b in ICCR2).

When the IIC detects the host address, the HOA flag in ICSR1 is set to 1 on the rising edge of the 9th SCL clock cycle, and at the same time, the RDRF flag in ICSR2 is set to 1 when the R/W# bit is 0 (Wr bit). This causes a receive data full interrupt (IICn_RXI) to be generated. The HOA flag indicates that the host address was sent from another device.

If the bit following the host address (0001 000b) is an Rd bit (R/W# bit = 1), the IIC can also detect the host address. After the host address is detected, the IIC operates in the same manner as in normal slave operation.



Note: n = 0, 1

Figure 32.29 HOA flag set timing during reception of host address

32.8 Wakeup Function

The IIC provides a wakeup function that causes the MCU to transition from Software Standby mode to normal operation. The wakeup function enables the reception of data when the peripheral module clock (PCLKB) is stopped, and generates a wakeup interrupt signal on a match of the slave address of the received data. This wakeup interrupt signal triggers the return to normal operation. After the wakeup interrupt occurs, switch the IIC to PCLKB synchronous operation so that communication can continue.

The wakeup function has four operation modes:

- Normal wakeup mode 1
- Normal wakeup mode 2
- Command recovery mode
- EEP response mode

Table 32.9 describes the behavior in these modes.

Table 32.9 Wakeup operation modes

Operation mode	ACK response timing	ACK response before wakeup to PCLKB synchronous operation	SCL state during wakeup to PCLKB synchronous operation
Normal wakeup mode 1	Before wakeup to PCLKB synchronous operation*1	ACK	Fixed low
Normal wakeup mode 2	After wakeup to PCLKB synchronous operation*2	Before wakeup: no response (NACK level retained) After wakeup: ACK response	Fixed low
Command recovery mode	Before wakeup to PCLKB synchronous operation*1	ACK	Open
EEP response mode	Before recovery to PCLKB synchronous operation*1	NACK	Open

Note 1. Switching timing from PCLKB asynchronous operation to PCLKB synchronous operation is the falling edge of the 9th clock of the SCL.
Note 2. Switching timing from PCLKB asynchronous operation to PCLKB synchronous operation is the falling edge of the 8th clock of the SCL.

The following can be selected as wakeup interrupt sources:

- Host address detection (valid when ICSER.HOAE = 1)
- General call address detection (valid when ICSER.GCAE = 1)

- Slave address 0[*1] detection (valid when ICSER.SAR0E = 1)
- Slave address 1[*1] detection (valid when ICSER.SAR1E = 1)
- Slave address 2[*1] detection (valid when ICSER.SAR2E = 1)

Note 1. Only 7-bit address can be set. Set the FS bit in SARUy (y = 0 to 2) to 0.

Precautions on the use of the wakeup function

- Do not change the content of the IIC registers except the ICIER register and WUSEN bit in ICWUR2 while the WUASYF flag in ICWUR2 is 1 (during PCLKB asynchronous operation).
- Set ICWUR.WUE and ICWUR.WUIE to 1, and ICCR2.MST and ICCR2.TRS to 0 (slave reception mode) before switching to PCLKB asynchronous mode.
- The device ID and the 10-bit slave address cannot be selected for the wakeup interrupt source. Set the DIDE bit in ICSER and FS bit in SARUy (y = 0 to 2) to 0.
- Set bits TIE, TEIE, RIE, NAKIE, SPIE, STIE, ALIE, and TMOIE in the ICIER register to 0 (interrupt disabled) after switching to PCLKB asynchronous operation (ICWUR2.WUASYF = 1).
- When the wakeup function is enabled, do not use the timeout function (ICWUR.WUE = 1)
- Even when a wakeup interrupt is generated during PCLKB asynchronous operation (when ICWUR2.WUASYF = 1), if the slave addresses match in PCLKB synchronous mode (ICWUR2.WUASYF = 0), the wakeup interrupt does not occur and the WUF flag is not set.
- If the timing of writing 0 to the ICWUR2.WUSEN bit and the timing of detecting a start condition conflict, the IIC might start the next reception in PCLKB synchronous operation mode. In this case, ICWUR2.WUASYF flag becomes 1 (switch to PCLKB asynchronous mode) when data communication is complete, a stop condition is detected, and detection of a wakeup event starts.
- If you want to switch from PCLKB asynchronous operation to PCLKB synchronous operation without address match detection, it will switch in the stop condition detection. When the ICWUR2.WUSEN bit was set to 1 in a bus free state, it is continued PCLKB asynchronous operation (Reception operation: waiting communication frame). ICWUR2.WUSYF flag becomes to 1 when IIC detect the stop condition of the next communication frame, and IIC switches to PCLKB synchronous operation.
- After writing 0 to the WUSEN bit in ICWUR2, do not change registers relate to the IIC operation mode setting (ICMR3, ICSER, and SARLy) until the mode is switched to PCLKB asynchronous operation from PCLKB synchronous operation (while the ICWUR2.WUASYF flag is 1). If the register value changes during this period by an interrupt handling or another factor, the IIC might malfunction before switching to the asynchronous operation.
- During PCLKB asynchronous operation (ICWUR2.WUSYF = 0 (or WUASYF = 1)), do not refer to each flag of ICSR1, ICSR2 register and ICCR2.BBSY flag.

32.8.1 Normal Wakeup Mode 1

This section describes the behavior, the timing, and an example operation of normal wakeup mode 1.

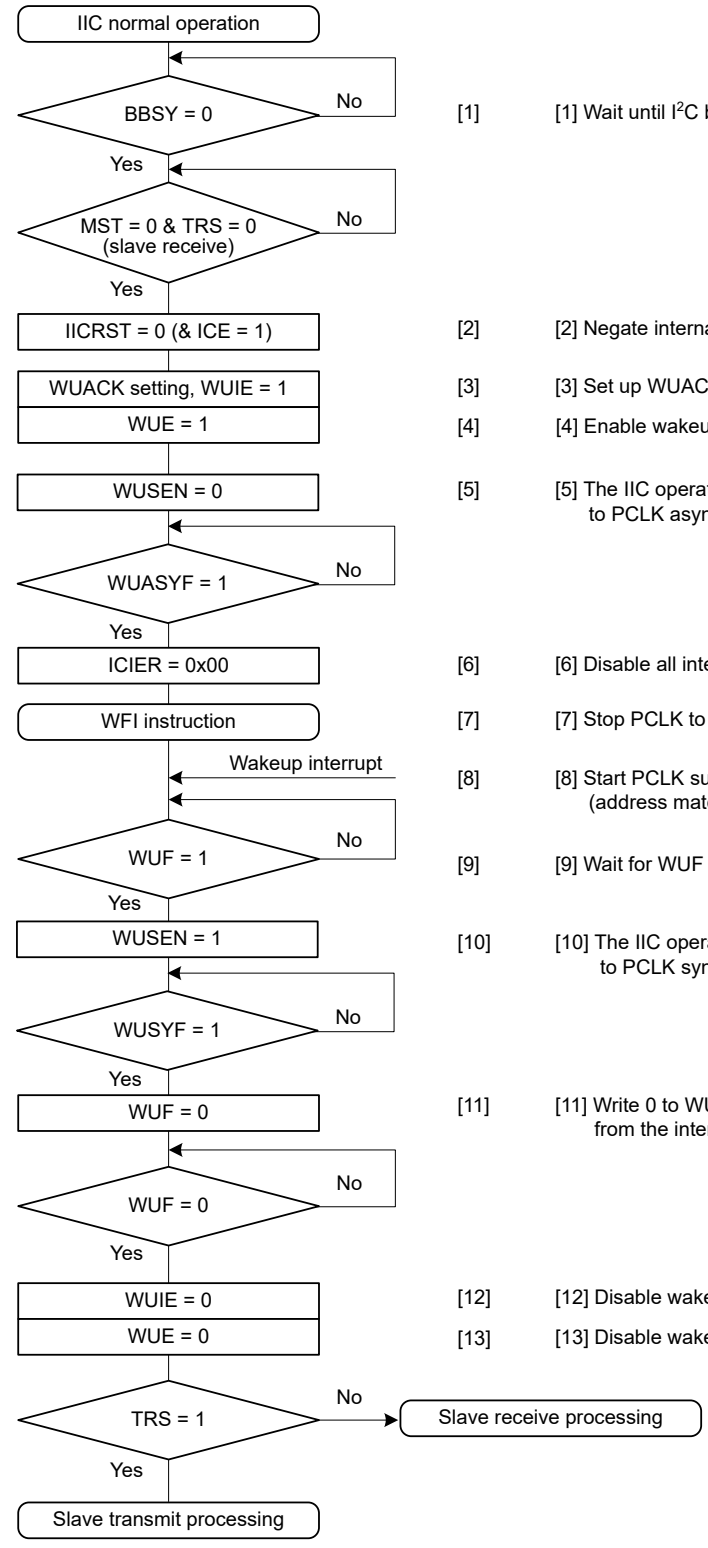
In normal wakeup mode 1, a wakeup interrupt triggered by the match of the slave address initiates the transition to normal operation as follows:

Before wakeup:	ACK is sent in response to the data received with its own slave address of the IIC.
During wakeup:	ACK response is made on the 9th clock cycle of SCL, after which SCL is held low[*1].
After wakeup:	Normal operation continues.

Note 1. Between the 9th clock cycle and 1st clock cycle during wakeup, ICMR1.WAIT = 1 is invalid.

If the slave address does not match, the SCL line is not held low after the 9th clock cycle of SCL, and the slave operation continues. Figure 32.30 shows an operation example, and Figure 32.32 shows the detailed timing.

If the transition from Software Standby mode is triggered by an interrupt other than a wakeup interrupt, for example the IRQn, the WUF flag is not set to 1. Figure 32.31 shows an operation example.



[1] [1] Wait until I^2C bus is open and stay in standby state.

[2] [2] Negate internal reset (if asserted).

[3] [3] Set up WUACK for the desired wakeup mode. Enable wakeup interrupt.

[4] [4] Enable wakeup function.

[5] [5] The IIC operating state is changed from PCLK synchronous to PCLK asynchronous.

[6] [6] Disable all interrupt requests except WUI.

[7] [7] Stop PCLK to IIC. IIC continues to receive.

[8] [8] Start PCLK supply to IIC on wakeup interrupt (address match).

[9] [9] Wait for WUF = 1.

[10] [10] The IIC operating state is changed from PCLK asynchronous to PCLK synchronous.

[11] [11] Write 0 to WUF. Read and check that WUF = 0 before returning from the interrupt handling.

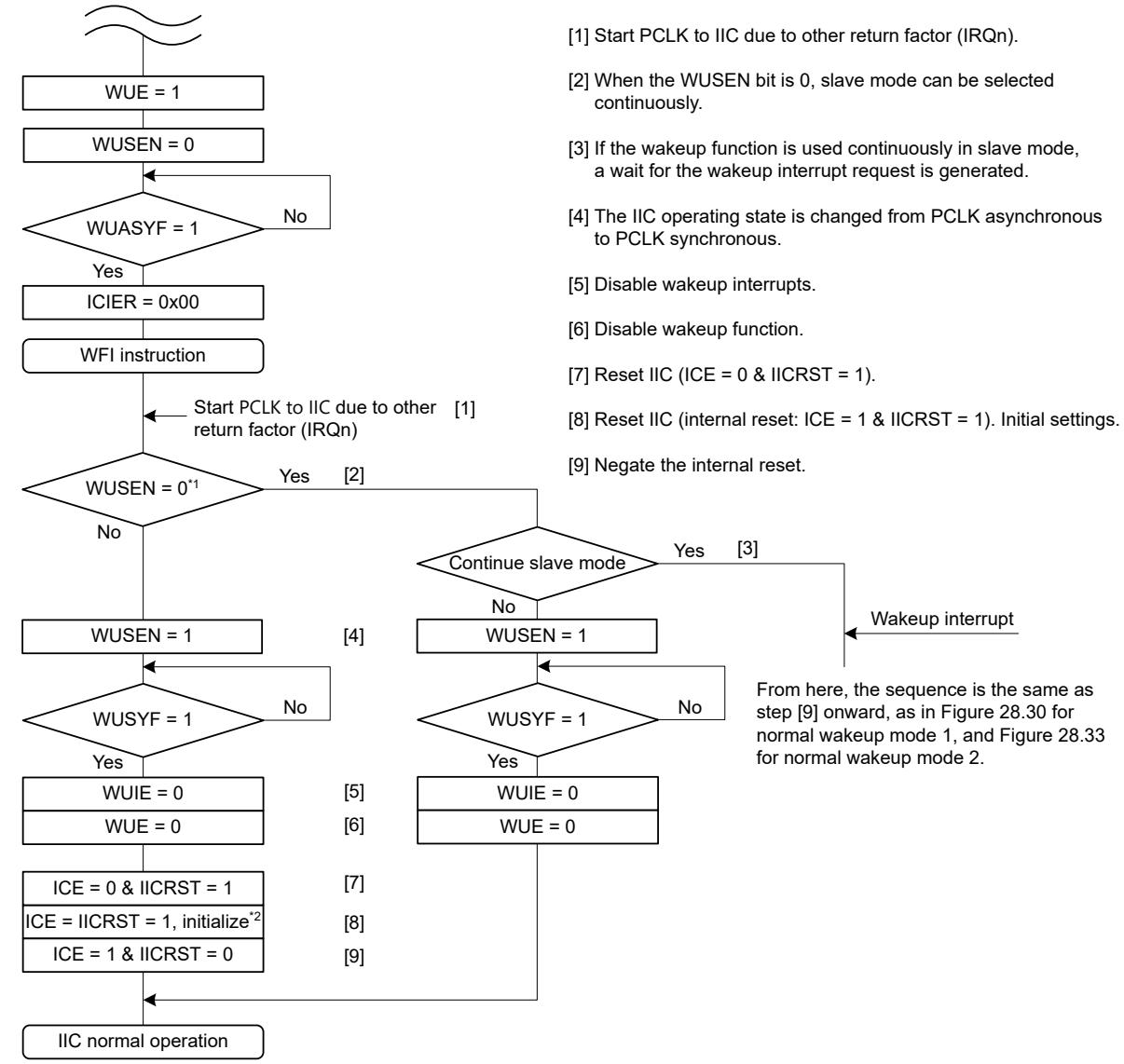
[12] [12] Disable wakeup Interrupt.

[13] [13] Disable wakeup function.

Note: PCLK = PCLKB

Figure 32.30 Example operation of normal wakeup mode 1 when wakeup is triggered by a wakeup interrupt on match of the slave address

Note: See Precautions on the use of the wakeup function.



Note: PCLK = PCLKB
Note 1. In the MCU, WUSEN bit is always 0 in the IIC.
Note 2. For details on the IIC initial settings, see section 32.3.2. Initial Settings.

Figure 32.31 Example operation of normal wakeup modes 1 and 2 when wakeup is triggered by an interrupt other than IIC wakeup interrupt, for example, the IRQn

Note: For details on the IIC initial settings, see section 32.3.2. Initial Settings.

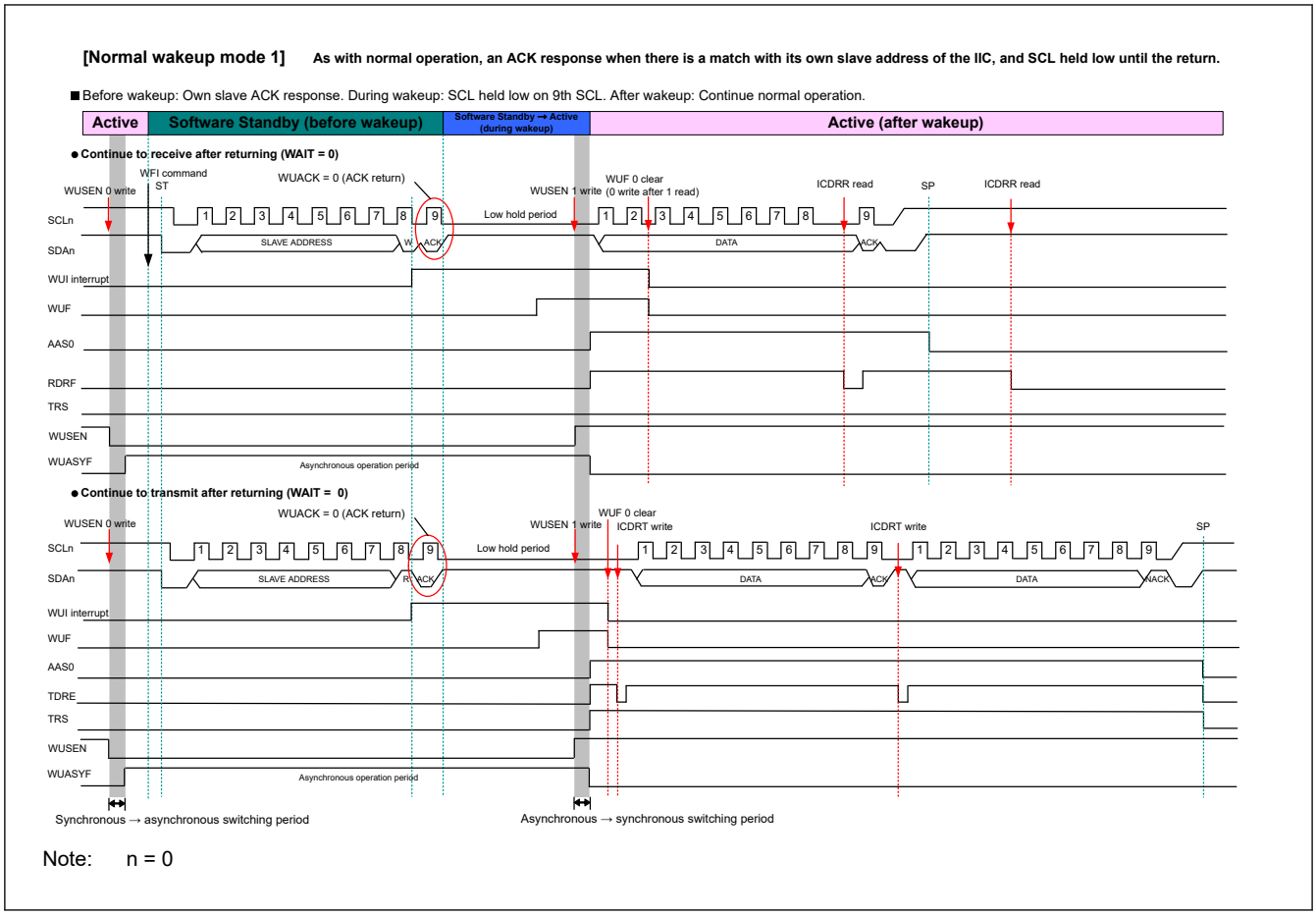


Figure 32.32 Timing of normal wakeup mode 1

32.8.2 Normal Wakeup Mode 2

This section describes the behavior, the timing, and an example operation of normal wakeup mode 2.

In normal wakeup mode 2, a wakeup interrupt triggered by a match of the slave address initiates the transition to normal operation as follows:

Before wakeup:	No response to data received with its own slave address until the end of the 8th SCL cycle.
During wakeup:	SCL line held low during the 8th and 9th clock cycles.
After wakeup:	ACK returns on the 9th clock cycle of SCL, and normal operation continues.

If the slave address does not match, the SCL line is not held low after the 8 th SCL clock cycle, and the slave operation continues. Figure 32.33 shows an example operation, and Figure 32.34 shows the detailed timing.

If the transition from Software Standby mode is triggered by an interrupt other than a wakeup interrupt, such as the IRQn, for example, the WUF flag is not set to 1. Figure 32.31 shows an operation example.

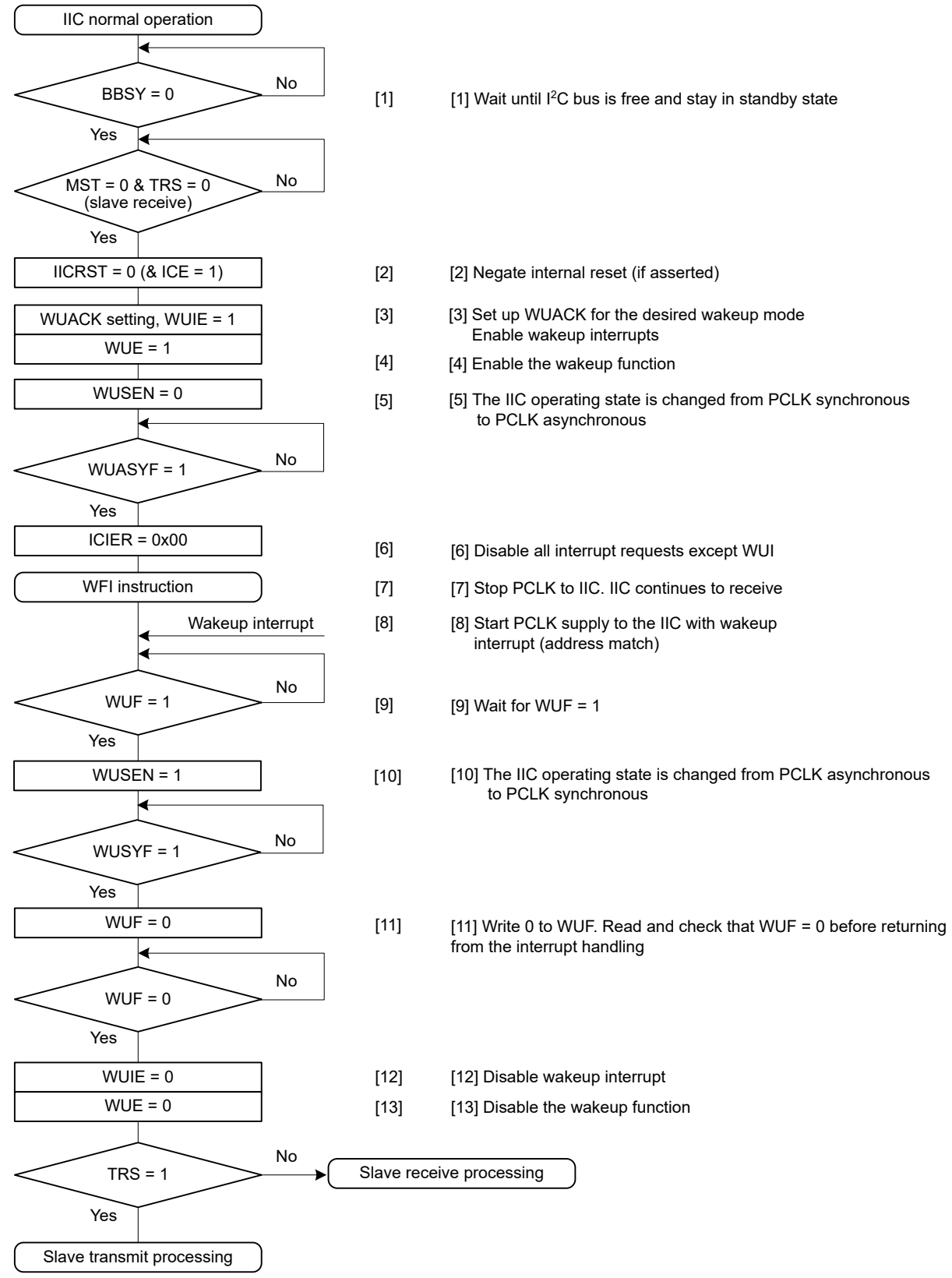
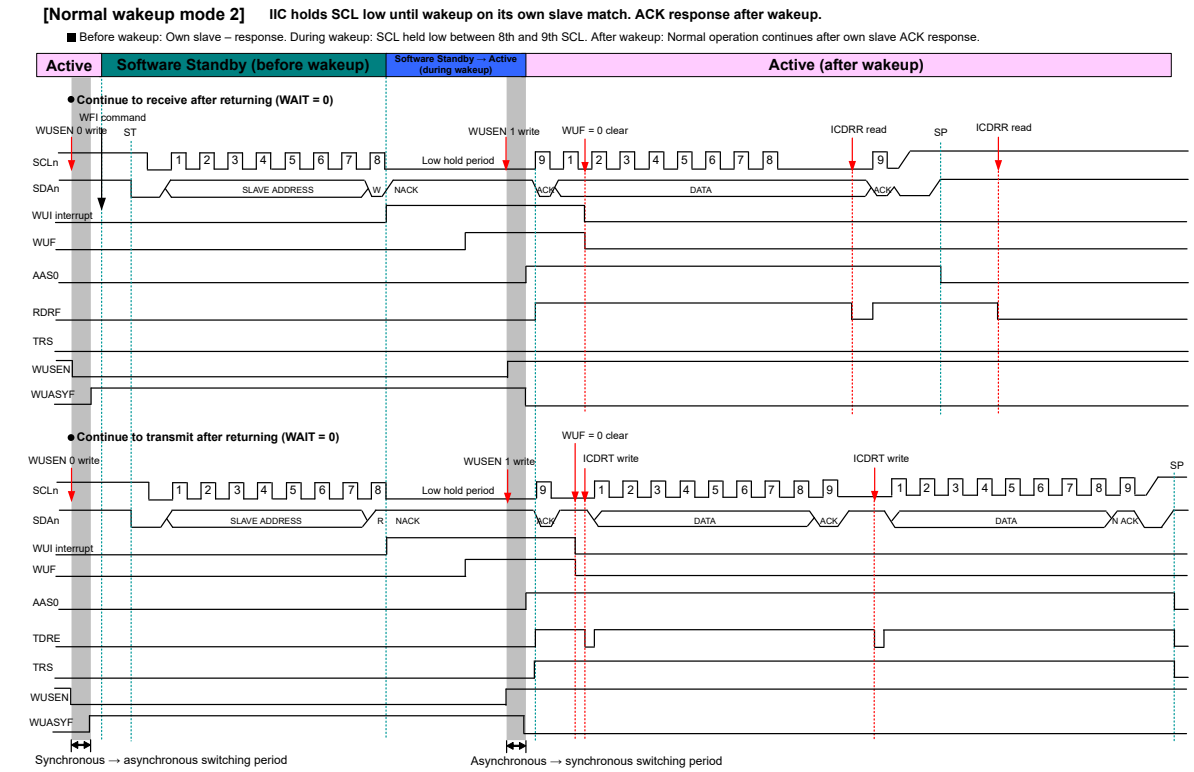


Figure 32.33 Example operation of normal wakeup mode 2 when wakeup is triggered by a wakeup interrupt on match of the slave address

Note: See Precautions on the use of the wakeup function.



Note: n = 0

Figure 32.34 Timing of normal wakeup mode 2

32.8.3 Command Recovery Mode and EEP Response Mode (Special Wakeup Modes)

This section describes the behavior, the timing, and example operations of the command recovery and EEP response modes.

In the command recovery and EEP response modes, the SCL line is not held low during the wakeup period (after the rise of the 9th clock cycle of SCL). Therefore, other IIC devices can use the I^2C bus during this period.

A wakeup interrupt triggered by a match of the slave address initiates the transition to normal operation as follows:

Before wakeup:	In response to the data received with its own slave address, the IIC returns ACK (command recovery mode) or NACK (EEP response mode).
During wakeup:	The SCL line is not held low.
After wakeup:	Normal operation continues after IIC initialization.

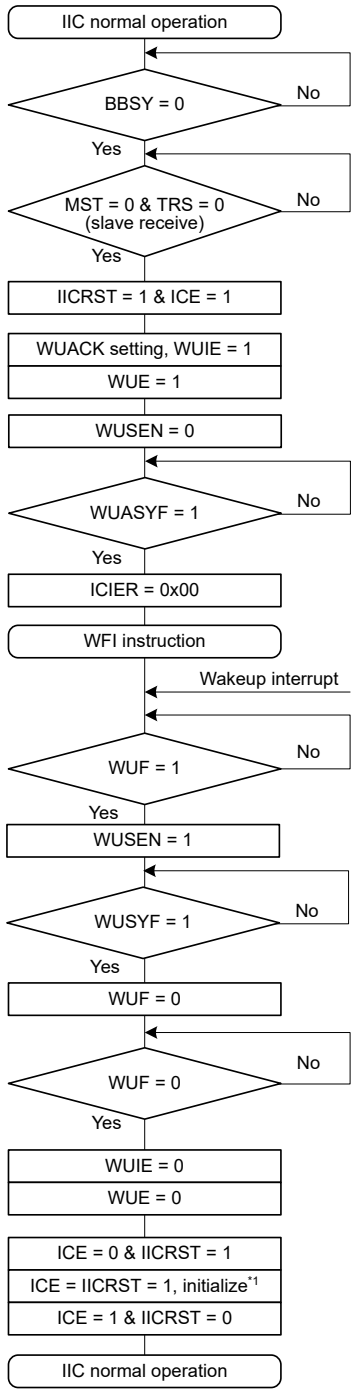
If the slave address does not match, the slave operation continues.

Note: Because the SCL line is not held low during wakeup, transmission or reception of the data that follows the slave address is not possible.

Note: The command recovery and EEP response modes are internal reset states (ICE = IICRST = 1). Therefore, the match of the slave address does not set the flags, HOA, GCA, ASS0, ASS1, and ASS2 in the ICSR1 register.

Figure 32.35 shows an example operation in recovery and EEP response modes. Figure 32.37 shows the detailed timing.

If the transition from Software Standby mode is triggered by an interrupt other than a wakeup interrupt, such as IRQn for example, the WUF flag is not set to 1. Follow the processing shown in Figure 32.36.



[1] [1] Wait until I^2C bus is open and stay in standby state.

[2] [2] Internal reset is asserted.

[3] [3] Set up WUACK for the desired wakeup mode. Enable wakeup interrupt.

[4] [4] Enable wakeup function.

[5] [5] The IIC operating state is changed from PCLK synchronous to PCLK asynchronous.

[6] [6] Disable all interrupt requests except WUI.

[7] [7] Stop PCLK to IIC. IIC continues to receive.

[8] [8] Start PCLK supply to the IIC with wakeup interrupt (address match).

[9] [9] Wait for WUF = 1.

[10] [10] The IIC operating state is changed from PCLK asynchronous to PCLK synchronous.

[11] [11] Write 0 to WUF. Read and check that WUF = 0 before returning from the interrupt handling.

[12] [12] Disable wakeup interrupt.

[13] [13] Disable wakeup function.

[14] [14] Reset IIC (ICE = 0 & IICRST = 1).

[15] [15] Reset IIC (internal reset: ICE = 1 & IICRST = 1). Initial settings.

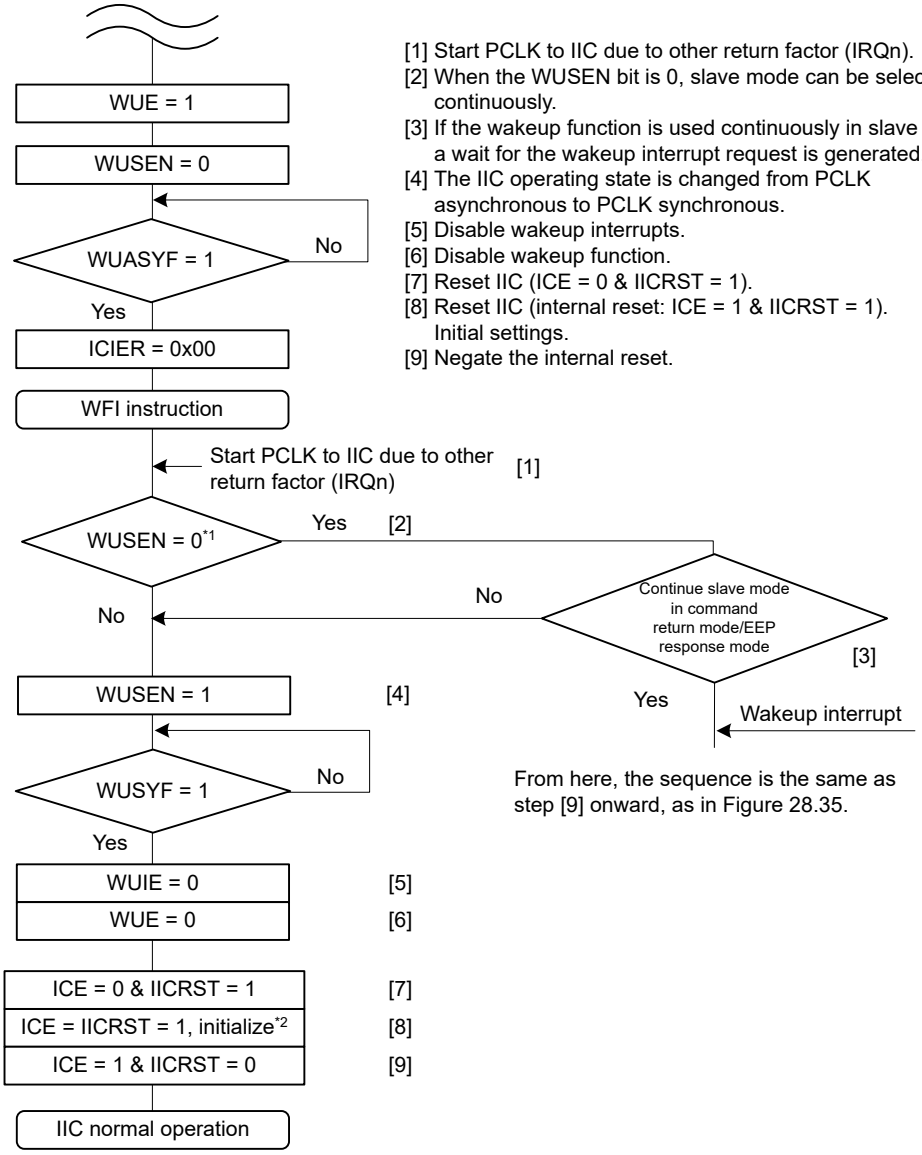
[16] [16] Negate the internal reset.

Note: PCLK = PCLKB

Note 1. For details on IIC initialization, see section 32.3.2. Initial Settings.

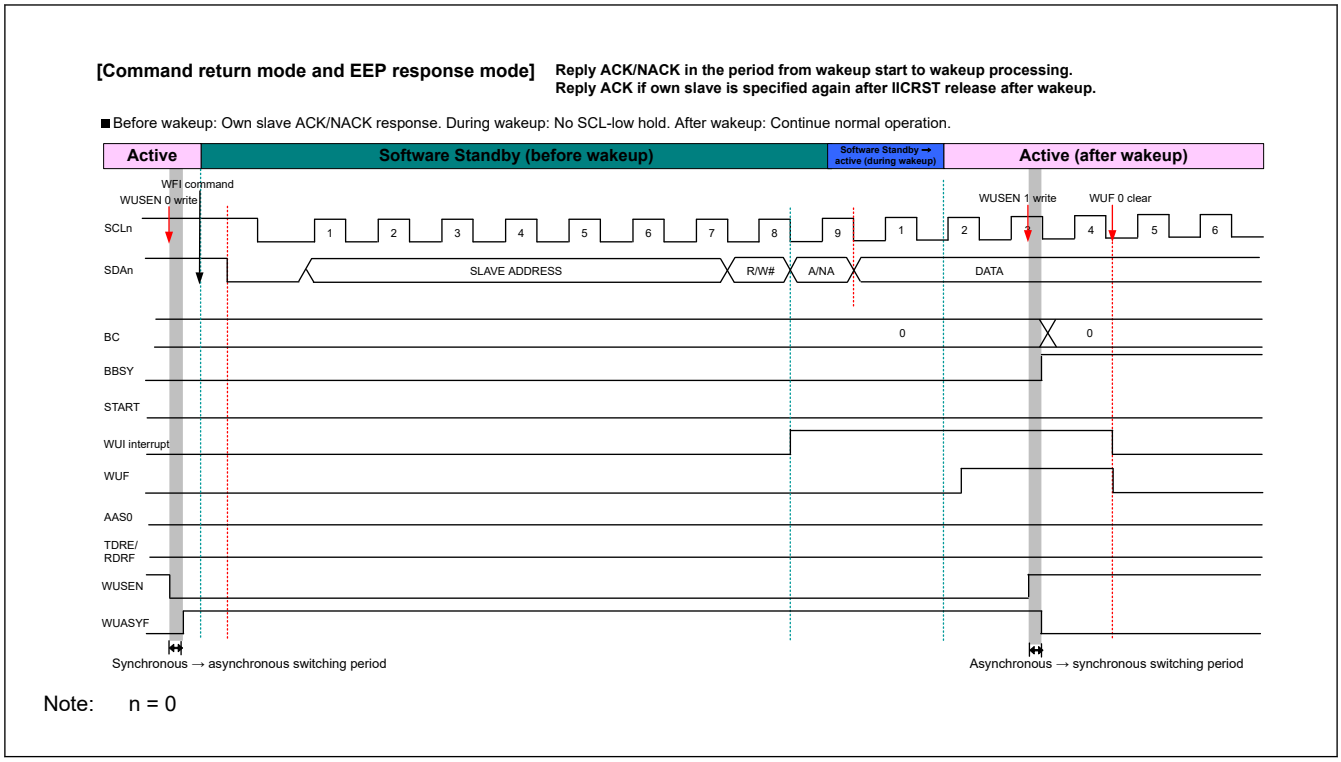
Figure 32.35 Example operation of command recovery mode and EEP response mode when wakeup is triggered by a wakeup interrupt on match of the slave address

Note: See Precautions on the use of the wakeup function.



Note: PCLK = PCLKB
Note 1. In the MCU, WUSEN bit is always 0 in the IIC.
Note 2. For details on IIC initialization, see section 32.3.2. Initial Settings.

Figure 32.36 Example operation of command recovery and EEP response modes when wakeup is triggered by an interrupt other than IIC wakeup interrupt, for example, the IRQn



Note: n = 0

Figure 32.37 Timing of command recovery and EEP response modes

32.9 Automatic Low-Hold Function for SCL

32.9.1 Function to Prevent Wrong Transmission of Transmit Data

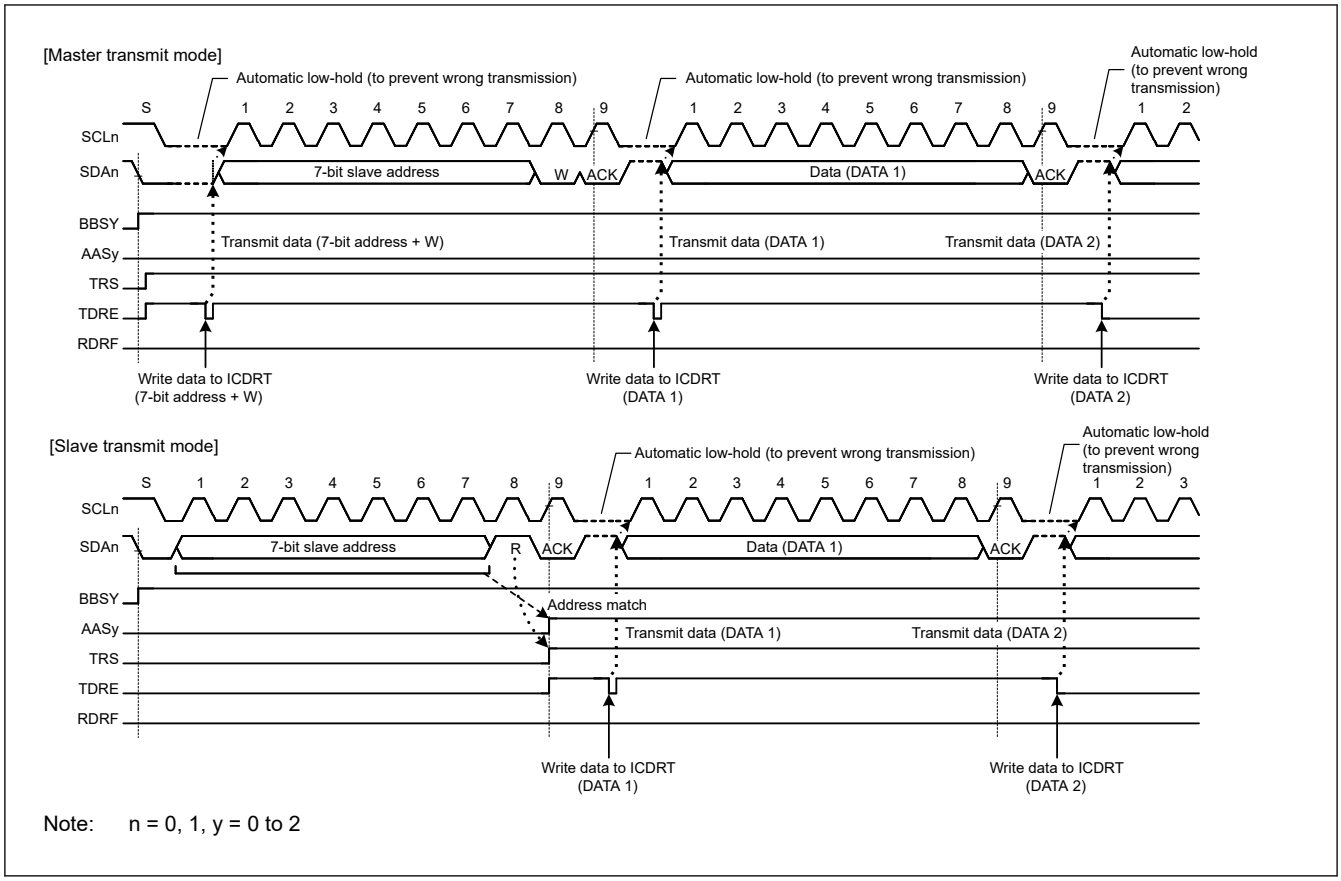
If the I^2C Bus Shift Register (ICDRS) is empty and data has not been written to the I^2C Bus Transmit Data Register (ICDRT) with the IIC in transmission mode (TRS bit = 1 in ICCR2), the SCLn line is automatically held low over the subsequent intervals. This low-hold period is extended until the transmit data is written, which prevents the unintended transmission of erroneous data.

Master transmit mode:

- Low-level interval after a start or restart condition is issued
- Low-level interval between the 9th clock cycle of one transfer and the 1st clock cycle of the next.

Slave transmit mode:

- Low-level interval between the 9th clock cycle of one transfer and the 1st clock cycle of the next.



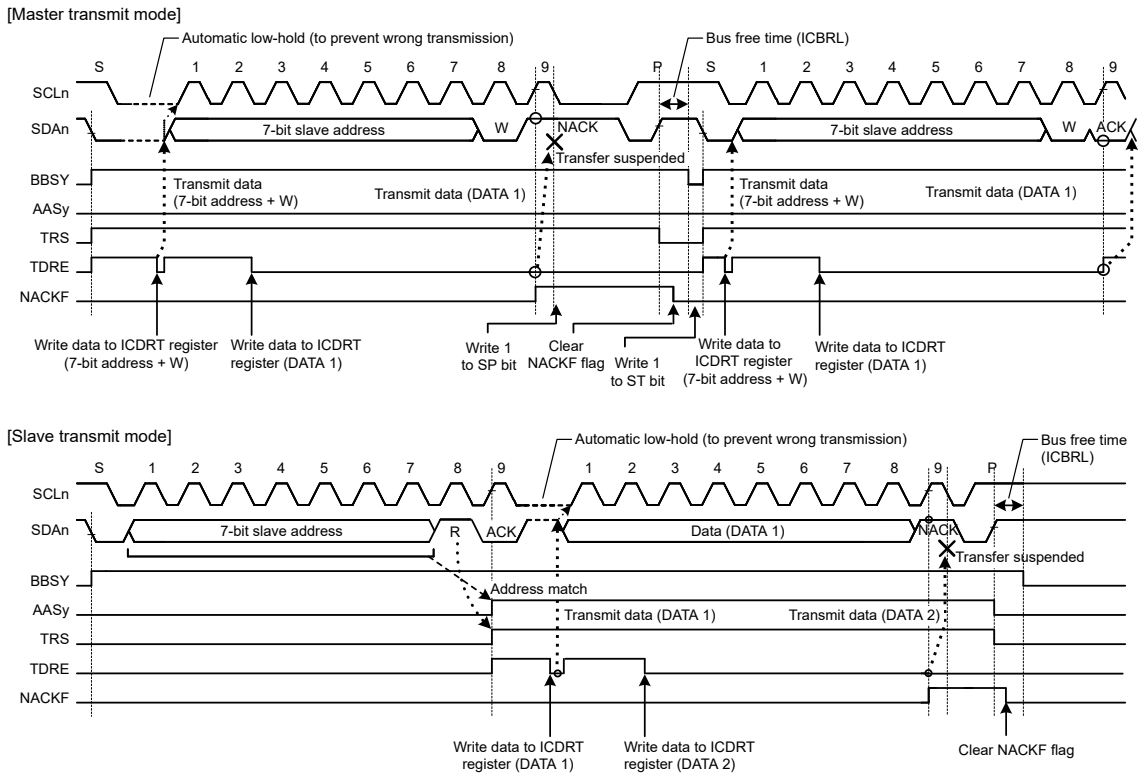
Note: n = 0, 1, y = 0 to 2

Figure 32.38 Automatic low-hold operation in transmit mode

32.9.2 NACK Reception Transfer Suspension Function

This function suspends transfer operation when NACK is received in transmit mode (TRS bit = 1 in ICCR2). This function is enabled when the NACKE bit in ICFER is set to 1. If the next transmit data is already written (TDRE flag = 0 in ICSR2) when NACK is received, the next data transmission on the falling edge of the 9th SCL clock cycle is automatically suspended. This prevents the SDAn line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (NACKF flag = 1 in ICSR2), transmit and receive operations are discontinued. To restore transmit or receive operation, after issuing the restart condition, you need to set the NACKF flag to 0 and try again, or set the NACKF flag to 0 after issuing the stop condition and then start again from issuing the start condition.



Note: n = 0, 1, y = 0 to 2

Figure 32.39 Suspension of data transfer when NACK is received, when NACKE = 1

32.9.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (ICDRR) read is delayed for a period of one transfer frame or more with receive data full (RDRF flag = 1 in ICSR2) in receive mode (TRS = 0 in ICCR2), the IIC holds the SCLn line low automatically immediately before the next data is received to prevent a failure to receive data.

This function is enabled even if the read processing of the final receive data is delayed and, in the meantime, the IIC slave address is designated after a stop condition is issued. This function does not interfere with other communication because the IIC does not hold the SCLn line low when a mismatch with its own slave address occurs after a stop condition is issued.

Periods in which the SCLn line is held low can be selected with a combination of the WAIT and RDRFS bits in ICMR3.

(1) 1-byte receive operation and automatic low-hold function using the WAIT bit

When the WAIT bit in ICMR3 is set to 1, the IIC performs a 1-byte receive operation using the WAIT bit function. Additionally, when the ICMR3.RDRFS bit is 0, the IIC automatically sends the ACKBT bit value in ICMR3 for the acknowledge bit in the period from the falling edge of the 8th SCL clock cycle to the falling edge of the 9th SCL clock cycle, and automatically holds the SCLn line low on the falling edge of the 9th SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from ICDRR, which enables byte-wise receive operation.

The WAIT bit function is enabled for receive frames after a match with the IIC slave address, including the general call address and host address, is obtained in master or slave receive mode.

(2) 1-byte receive operation (ACK/NACK transmission control) and automatic low-hold function using the RDRFS bit

When the RDRFS bit in ICMR3 is set to 1, the IIC performs a 1-byte receive operation using the RDRFS bit function. When the RDRFS bit is set to 1, the RDRF flag in ICSR2 is set to 1 (receive data full) on the rising edge of the eighth SCL clock cycle, and the SCLn line is automatically held low on the falling edge of the eighth SCL clock cycle. This low-hold

is released by writing a value to the ACKBT bit in ICMR3, but cannot be released by reading data from ICDRR, which enables receive operation through the ACK or NACK transmission control based on the data received in byte units.

The RDRFS bit function is enabled for receive frames after a match with the IIC slave address, including the general call address and host address, is obtained in master or slave receive mode.

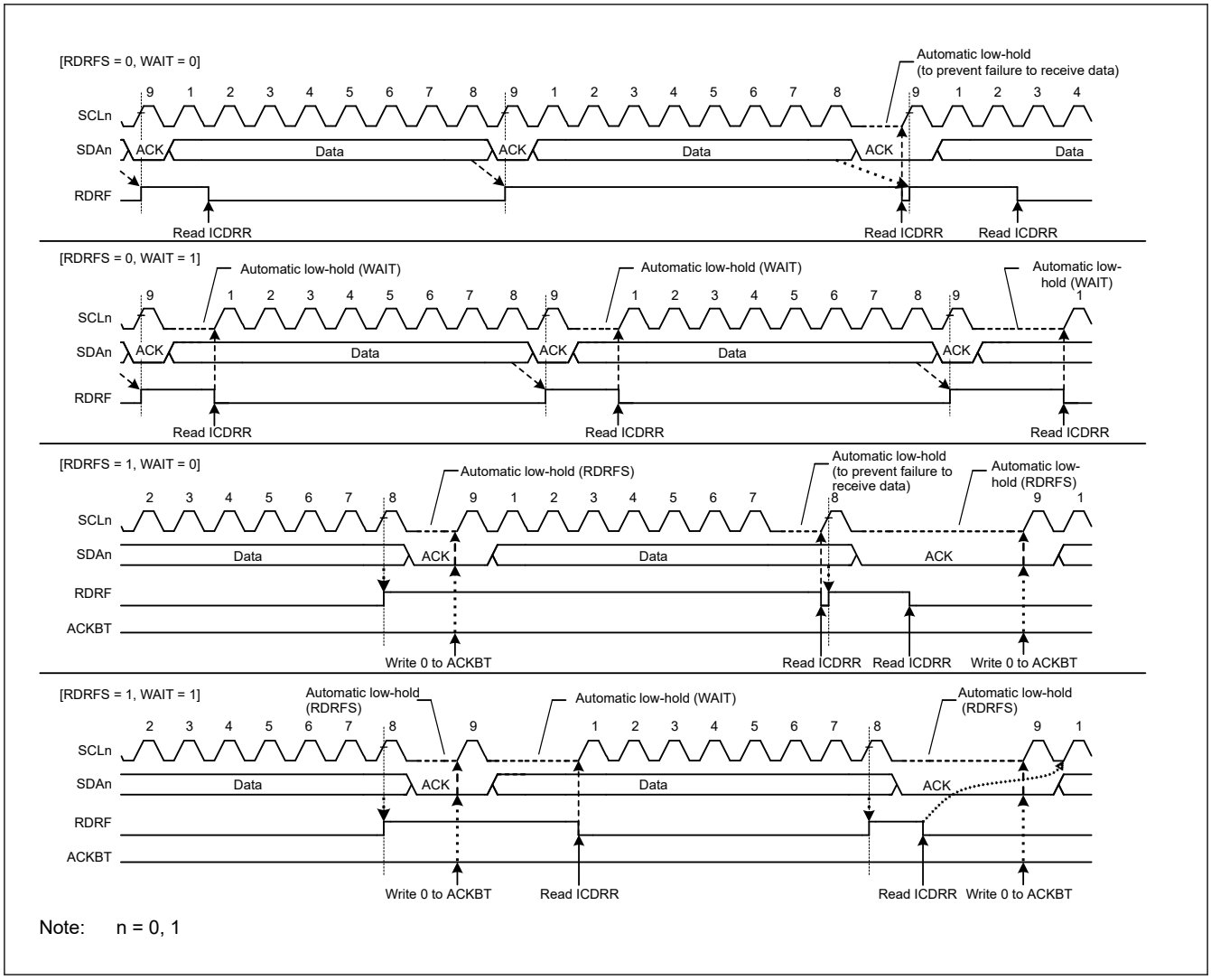


Figure 32.40 Automatic low-hold operation in receive mode using the RDRFS and WAIT bits

32.10 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I^2C bus standard, the IIC provides functions to prevent double-issue of a start condition, detect arbitration-lost during transmission of NACK, and detect arbitration-lost in slave transmit mode.

32.10.1 Master Arbitration-Lost Detection (MALE Bit)

The IIC drives the SDAn line low to issue a start condition. However, if the SDAn line was already driven low by another master device issuing a start condition, the IIC regards its own start condition as an error and considers this a loss in arbitration. Priority is given to transfer by the other master device. Similarly, if a request to issue a start condition is made by setting the ST bit in ICCR2 to 1 while the bus is busy (BBSY flag = 1 in ICCR2), the IIC regards this as a double-issuing-of-start-condition error and considers itself to have lost in arbitration. This prevents a failure of transfer resulting from a start condition being issued while transfer is in progress.

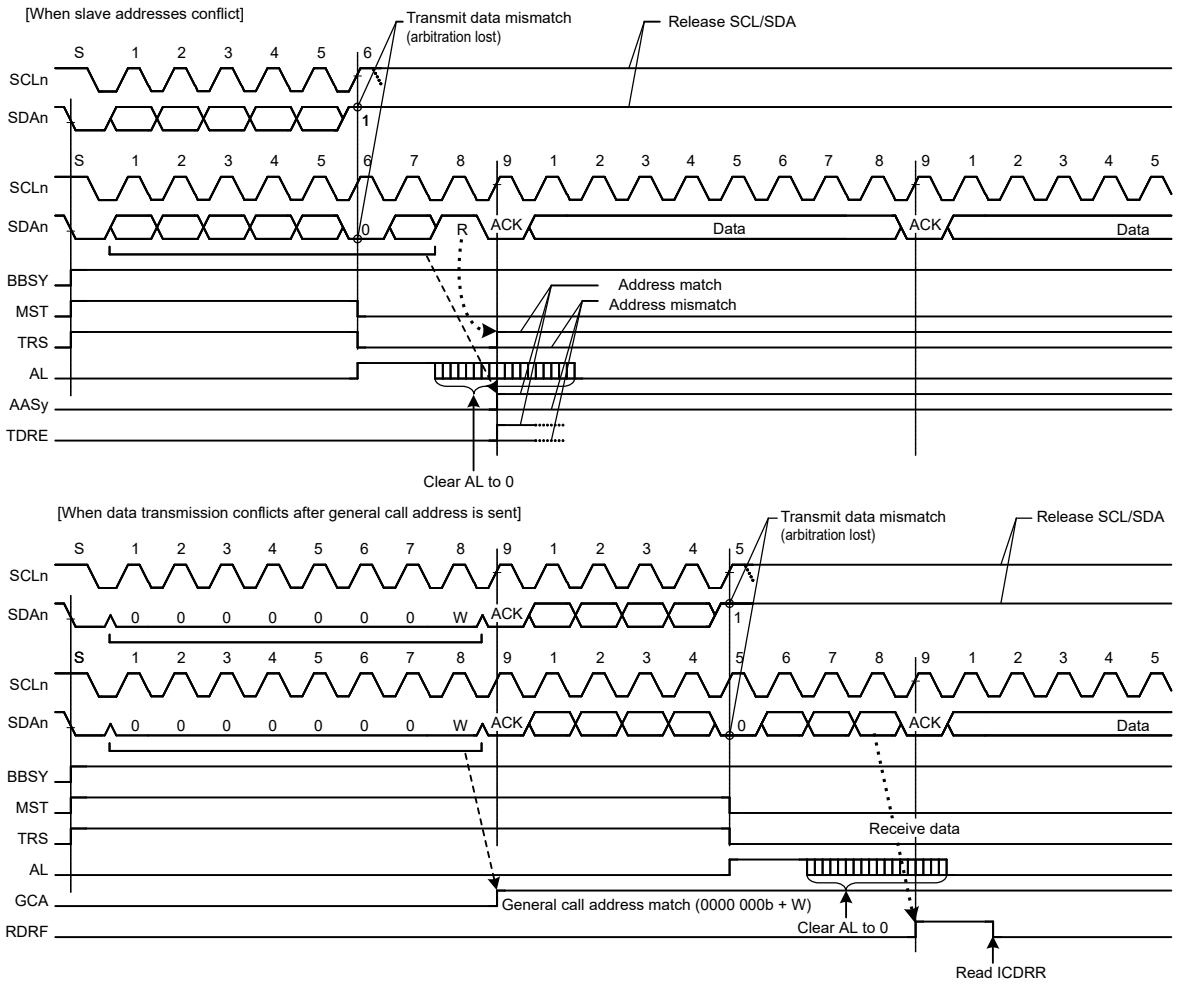
When a start condition is issued successfully, if the transmit data including the address bits (internal SDA output level) and the level on the SDAn line do not match (high output as the internal SDA output, meaning the SDAn pin is in the high-impedance state and a low level is detected on the SDAn line), the IIC loses in arbitration.

After a loss in arbitration of mastership, the IIC immediately enters slave receive mode. If a slave address, including the general call address, matches its own address at this time, the IIC continues in slave operation.

A loss in arbitration of mastership is detected when the following conditions are met while the MALE bit in ICFER is 1 (master arbitration-lost detection enabled).

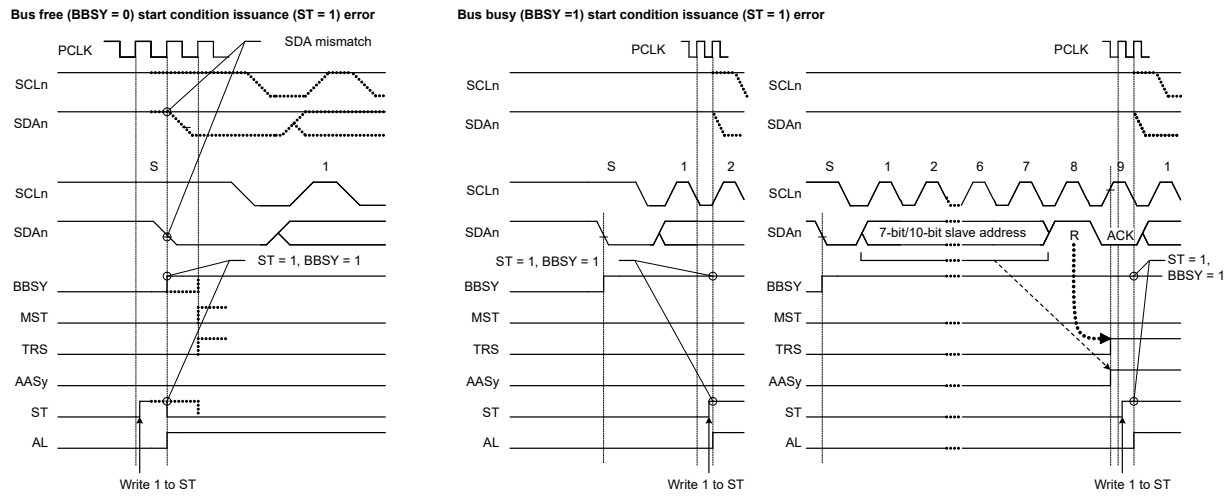
[Master arbitration-lost conditions]

- Mismatching of the internal level for output on SDA and the level on the SDAn line after a start condition was issued by setting the ST bit in ICCR2 to 1 while the BBSY flag in ICCR2 is set to 0 (erroneous issuing of a start condition)
- Setting of the ST bit in ICCR2 to 1 (start condition double-issue error) while the BBSY flag is 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDAn line in master transmit mode (MST and TRS bits = 11b in ICCR2).



Note: n = 0, 1

Figure 32.41 Examples of master arbitration-lost detection when MALE = 1

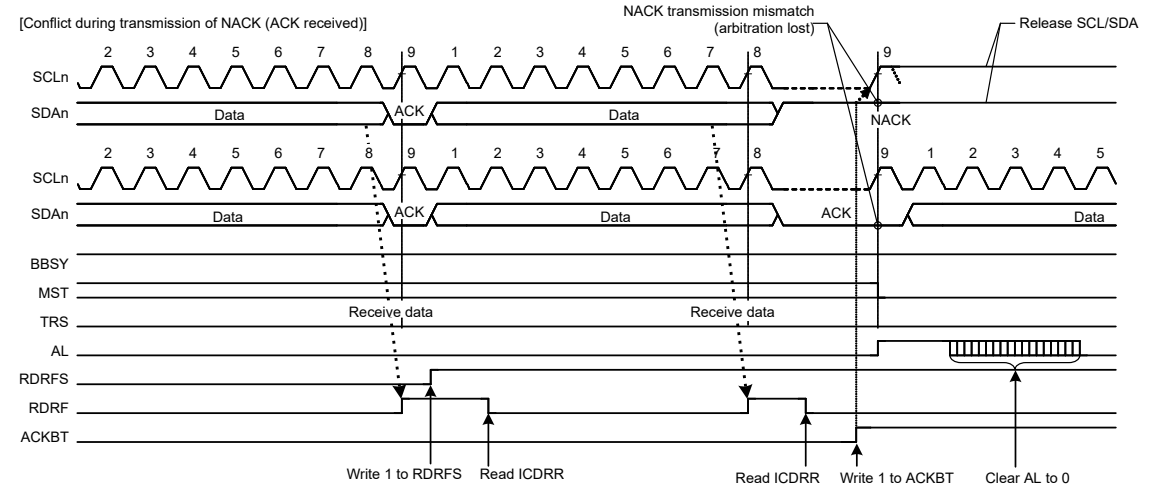


Note: PCLK = PCLKB
Note: n = 0, 1, y = 0 to 2

Figure 32.42 Arbitration-lost when start condition is issued when MALE = 1

32.10.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)

This function causes arbitration to be lost if the internal SDA output level does not match the level on the SDAn line (high output as the internal SDA output, meaning the SDAn pin is in the high-impedance state) and the low level is detected on the SDAn line during transmission of NACK in receive mode. Arbitration is lost because of a conflict between NACK and ACK transmissions when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such a conflict occurs when multiple master devices send or receive the same information through a single slave device. Figure 32.43 shows an example of arbitration-lost detection during transmission of NACK.



Note: n = 0, 1

Figure 32.43 Example of arbitration-lost detection during transmission of NACK when NALE = 1

The following explains arbitration-lost detection using an example in which two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives 2 bytes of data from the slave device, and master B receives 4 bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in either master A or master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Master A sends NACK when it has received 2 final bytes of

data from the slave device. Meanwhile, master B sends ACK because it has not received the required 4 bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect the ACK transmitted by master B and issues a stop condition. The stop condition issue conflicts with the SCL clock output of master B, which disrupts communication.

When the IIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost. If arbitration is lost during transmission of NACK, the IIC immediately cancels the slave match condition and enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing, such as 0xFF transmission processing, which is required if the UDID (Unique Device Identifier) of the assigned address does not match in the Get UDID general processing after the Assign Address command.

The IIC detects arbitration-lost during transmission of NACK when the following condition is met with the NALE bit in ICFER set to 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

- When the internal SDA output level does not match the SDAn line (ACK is received) during transmission of NACK (ACKBT bit = 1 in ICMR3).

32.10.3 Slave Arbitration-Lost Detection (SALE Bit)

This function causes arbitration to be lost if the transmit data (internal SDA output level) and the level on the SDAn line do not match (high output as the internal SDA output, meaning the SDAn pin is in the high-impedance state), and the low level is detected on the SDAn line in slave transmit mode. This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.

When the IIC loses slave arbitration, the IIC is immediately released from the slave-matched state and enters slave receive mode. This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminates subsequent redundant processing for the transmission of 0xFF.

The IIC detects slave arbitration-lost when the following condition is met with the SALE bit in ICFER set to 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

- When transmit data excluding acknowledge (internal SDA output level) does not match the SDAn line in slave transmit mode (MST and TRS bits = 01b in ICCR2).

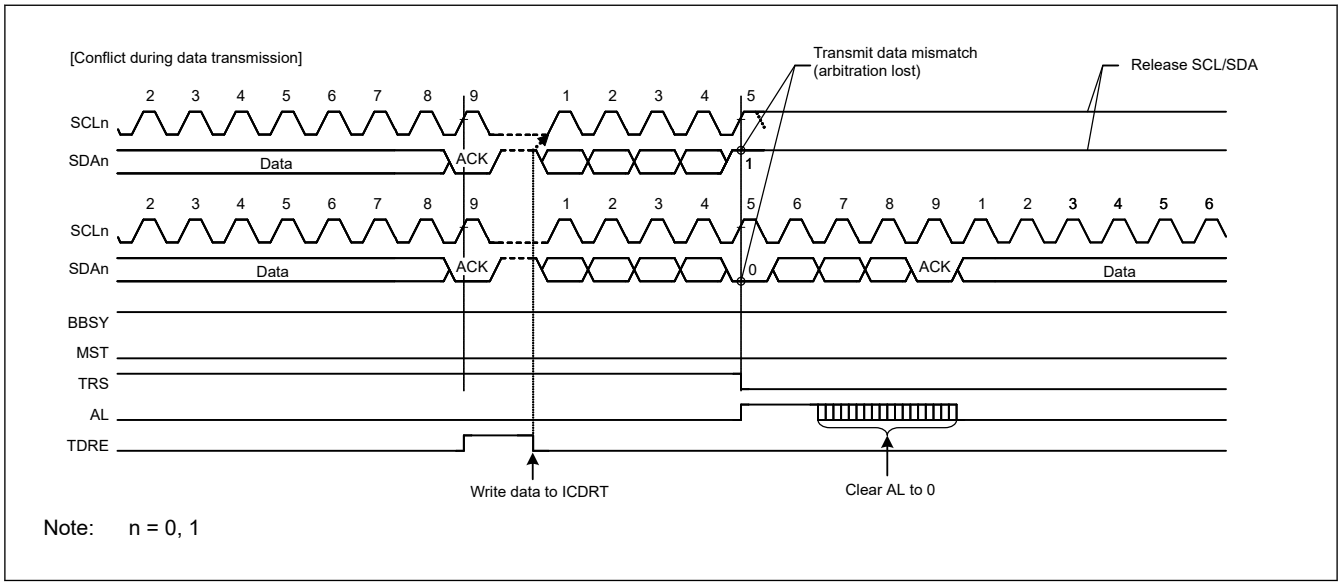


Figure 32.44 Example of slave arbitration-lost detection when SALE = 1

32.11 Start, Restart, and Stop Condition Issuing Function

32.11.1 Issuing a Start Condition

The IIC issues a start condition when the ST bit in ICCR2 is set to 1. When the ST bit is set to 1, a start condition request is made, and the IIC issues a start condition when the BBSY flag in ICCR2 is 0 (bus free state). When a start condition is issued normally, the IIC automatically shifts to the master transmit mode.

To issue a start condition:

1. Drive the SDAn line low (high level to low level).
2. Ensure that the time set in ICBRH and the start condition hold time elapse.
3. Drive the SCLn line low (high level to low level).
4. Detect low level on the SCLn line and ensure the low-level period of the SCLn line set in ICBRL elapses.

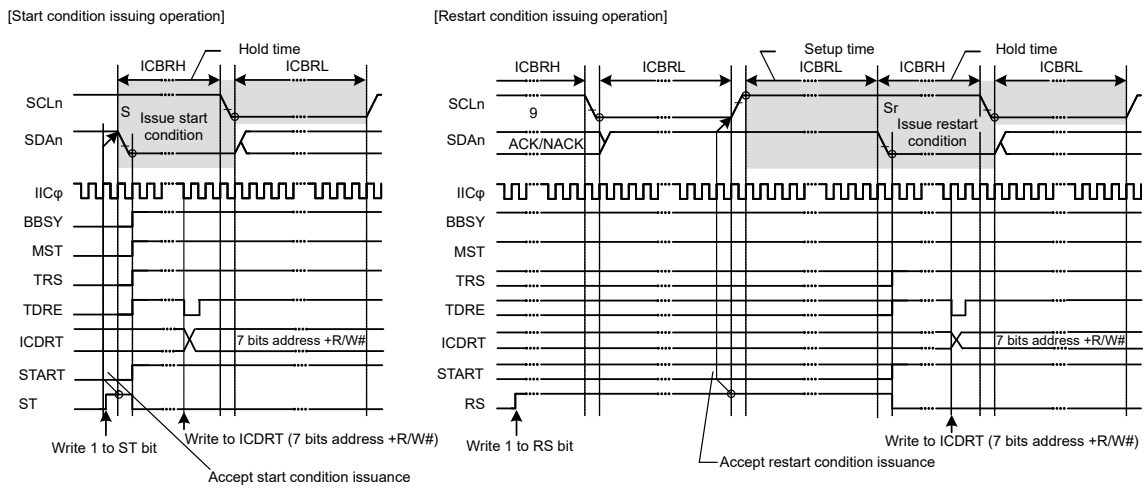
32.11.2 Issuing a Restart Condition

The IIC issues a restart condition when the RS bit in ICCR2 is set to 1. When the RS bit is set to 1, a restart condition request is made, and the IIC issues a restart condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).

To issue a restart condition:

1. Release the SDAn line.
2. Ensure the low-level period of the SCLn line set in ICBRL elapses.
3. Release the SCLn line (low level to high level).
4. Detect a high level on the SCLn line and ensure the time set in ICBRL and the restart condition setup time elapse.
5. Drive the SDAn line low (high level to low level).
6. Ensure the time set in ICBRH and the restart condition hold time elapse.
7. Drive the SCLn line low (high level to low level).
8. Detect a low level on the SCLn line and ensure the low-level period of the SCLn line set in ICBRL elapses.

Note: When issuing restart condition requests, write the slave address to ICDRT after confirming that ICCR2.RS = 0. Data written while ICCR2.RS = 1 is not forwarded because of the retransmission condition before the occurrence.



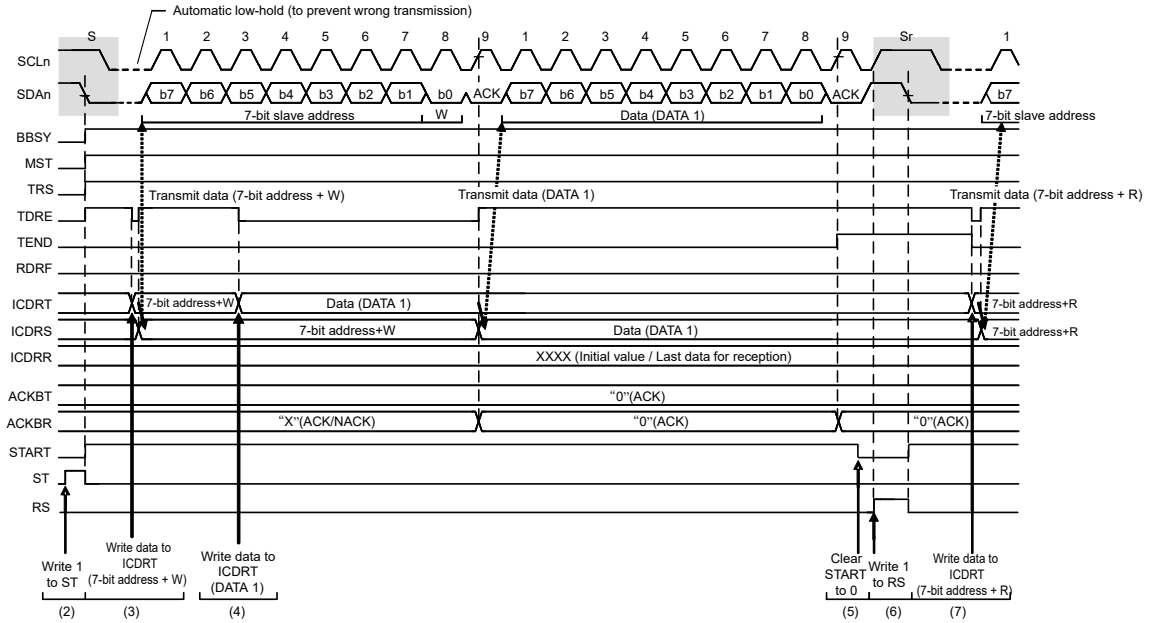
Note: n = 0, 1

Figure 32.45 Start and restart condition issue timing using the ST and RS bits

Figure 32.46 shows the operation timing when a restart condition is issued after the master transmission.

[To issue a restart condition after the master transmission:]

1. Initialize the IIC using the procedure in section 32.3.2. Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is open, then set the ST bit in ICCR2 to 1 (start condition issuance request). On receiving the request, the IIC issues a start condition. At the same time, the BBSY and the START flags in ICSR2 are automatically set to 1 and the ST bit is automatically set to 0. If the start condition is detected and the internal levels for the SDA output state and the levels on the SDAn line match while the ST bit is 1, the IIC recognizes that a start condition is successfully issued as requested by the ST bit has been successfully completed. The MST and TRS bits in ICCR2 are automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 is also automatically set to 1 when the TRS bit is set to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. After the data for transmission is written to ICDRT, the TDRE flag is automatically set to 0, data is transferred from ICDRT to ICDRS, and the TDRE flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode according to the value of the transmitted R/W# bit. If the value of the R/W# bit is 0, the IIC continues in master transmit mode. If the ICSR2.NACKF flag is 1 at this time, indicating that no slave device recognized the address or that there was an error in communications, write 1 to ICCR2.SP bit to issue a stop condition. To transmit data with an address in the 10-bit format, start by writing 1111 0b, the 2 upper bits of the slave address, and W to ICDRT as the first address transmission. Then, as the second address transmission, write the 8 lower bits of the slave address to ICDRT.
4. After confirming that the TDRE flag in ICSR2 is 1, write data for transmission to the ICDRT register. The IIC automatically holds the SCLn line low until data for transmission is ready, a restart condition is issued or a stop condition is issued.
5. After all bytes of data for transmission are written to the ICDRT register, wait until the value of the TEND flag in ICSR2 returns to 1. Then after checking that the START flag in ICSR2 is 1, set the START flag in ICSR2 to 0.
6. Set the RS bit in ICCR2 to 1 (restart condition issue request). On receiving the request, the IIC issues a restart condition.
7. After checking that the START flag in ICSR2 is 1, write the value for transmission (the slave address and the R/W# bit) to ICDRT.



Note: n = 0, 1

Figure 32.46 Restart condition issue timing after master transmission.

32.11.3 Issuing a Stop Condition

The IIC issues a stop condition when the SP bit in ICCR2 is set to 1. When the SP bit is set to 1, a stop condition request is made, and the IIC issues a stop condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).

To issue a stop condition:

1. Drive the SDAn line low (high level to low level).
2. Ensure the low-level period of the SCLn line set in ICBRL elapses.
3. Release the SCLn line (low level to high level).
4. Detect a high level on the SCLn line and ensure the time set in ICBRH and the stop condition setup time elapse.
5. Release the SDAn line (low level to high level).
6. Ensure the time set in ICBRL and the bus free time elapse.
7. Clear the BBSY flag to 0 to release the bus mastership.

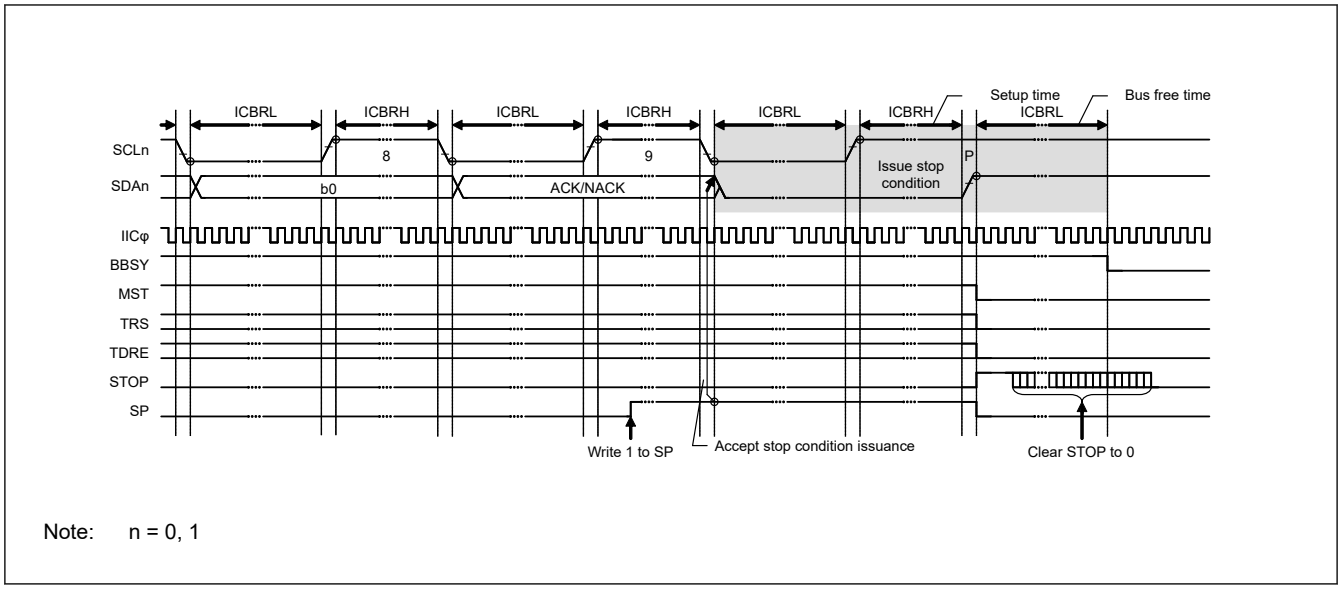


Figure 32.47 Stop condition issue timing using the SP bit

32.12 Bus Hanging

If the clock signals from the master and slave devices are out of synchronization because of noise or other factors, the I^2C bus might hang with a fixed level on the SCLn line or SDAn line.

To manage bus hanging, the IIC has a timeout function to detect hanging by monitoring the SCLn line, and a function for outputting an extra SCL clock cycle to release the bus from:

- A timeout function to detect hanging by monitoring the SCLn line
- The IIC reset function
- An internal reset function.

By checking the SCLO, SDAO, SCLI, and SDAI bits in ICCR1, it is possible to see whether the IIC or its communicating partner is placing the low level on the SCLn or SDAn line.

32.12.1 Timeout Function

The timeout function can detect when the SCLn line is stuck longer than the predetermined time. The IIC can detect an abnormal bus state by monitoring that the SCLn line is stuck low or high for a predetermined time.

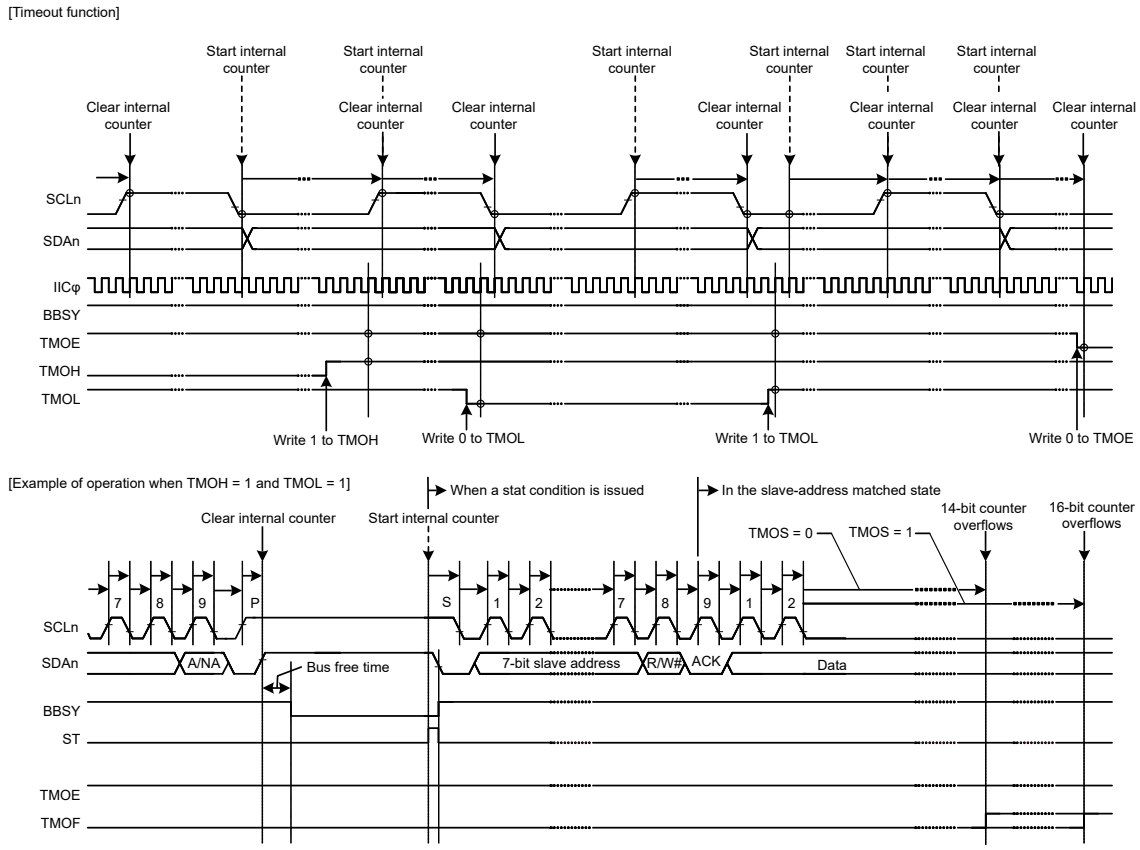
The timeout function monitors the SCLn line state and counts the low- or high-level period using the internal counter. The timeout function resets the internal counter each time the SCLn line changes (rises or falls), but continues to count unless the SCLn line changes. If the internal counter overflows because no SCLn line changes, the IIC can detect the timeout and report the bus hung state.

This timeout function is enabled when the ICFER.TMOE bit is 1. It detects a hung state when the SCLn line is stuck low or high during the following conditions:

- The bus is busy (ICCR2.BBSY flag is 1) in master mode (ICCR2.MST bit is 1)
- The IIC slave address is detected (ICSR1 register is not 0x00) and the bus is busy (ICCR2.BBSY flag is 1) in slave mode (ICCR2.MST bit is 0)
- The bus is open (ICCR2.BBSY flag is 0) while a start condition is requested (ICCR2.ST bit is 1).

The internal counter of the timeout function uses the internal reference clock (IICφ) set in the CKS[2:0] bits in ICMR1 as a count source. It functions as a 16-bit counter when long mode is selected (TMOS bit = 0 in ICMR2) or a 14-bit counter when short mode is selected (TMOS bit = 1).

The SCLn line level (low, high, or both levels) during which this counter is activated can be selected in the TMOH and TMOL bits in ICMR2. If both TMOL and TMOH bits are set to 0, the internal counter is disabled.



Note: n = 0, 1

Figure 32.48 Timeout function using the TMOE, TMOS, TMOH, and TMOL bits

32.12.2 Extra SCL Clock Cycle Output Function

In master mode, this function outputs extra SCL clock cycles to release the SDAn line of the slave device from being held low because the master is out of synchronization with the slave device. This function is mainly used in master mode to release the SDAn line of the slave device from being fixed low by including extra cycles of SCL output from the IIC. It uses single cycles of the SCL clock for a bus error where the IIC cannot issue a stop condition because the slave device is holding the SDAn line at the low level. Do not use this function in normal situations. Using it when communications are proceeding correctly leads to malfunctions.

When the CLO bit in ICCR1 is set to 1 in master mode, a single cycle of the SCL clock at the frequency specified in the CKS[2:0] bits in ICMR1, and in the ICBRH and ICBRL registers, is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit is automatically set to 0. At this time, if ICCR2.BBSY = 1, the SCL pin goes low, and when ICCR2.BBSY = 0, the SCL pin goes high. After confirming that the CLO bit is 0 by software, write 1 to the CLO bit to output the additional clock continuously.

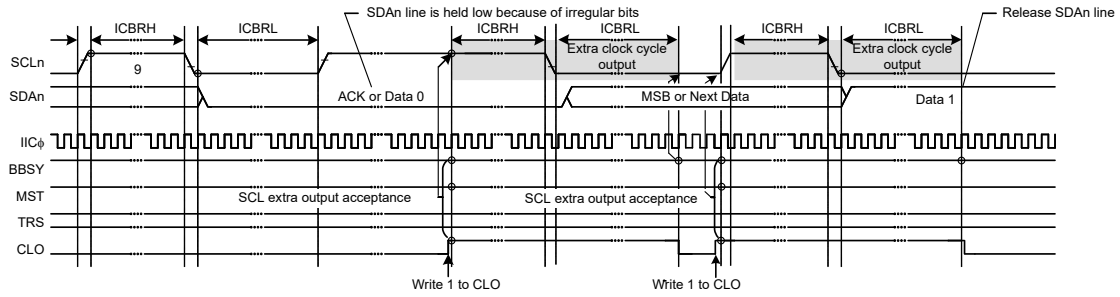
When the IIC module is in master mode and the slave device is holding the SDAn line low because synchronization with the slave device is lost because of effects like noise, the output of a stop condition is not possible. This function can be used to output extra cycles of SCL one by one to make the slave device release the SDAn line from being held low, and so recover the bus from an unusable state. Release of the SDAn line by the slave device can be monitored by reading the SDAI bit in ICCR1. After confirming the release of the SDAn line by the slave device, complete communications by reissuing the stop condition.

[Output conditions for using the CLO bit in ICCR1]

- When the bus is open (BBSY flag in ICCR2 = 0) or in master mode (MST bit = 1 and BBSY flag = 1 in ICCR2)

- When the communication device does not hold the SCLn line low.

Figure 32.49 shows the operation timing of the extra SCL clock cycle output function (CLO bit).



Note: n = 0, 1

Figure 32.49 Extra SCL clock cycle output function using the CLO bit

32.12.3 IIC Reset and Internal Reset

The IIC module incorporates a function for resetting itself. It uses two types of resets:

- An IIC reset, which initializes all registers, including the BBSY flag in ICCR2.
- An internal reset, which releases the IIC from the slave-address matched state and initializes the internal counter while saving other settings.

After issuing a reset, always set the IICRST bit in ICCR1 to 0. Both types of resets are valid for release from bus-hung states, because both restore the output state of the SCLn and SDAn pins to the high-impedance state.

Issuing a reset during slave operation might lead to a loss of synchronization between the master device clock and the slave device clock, so avoid this when possible. In addition, monitoring of the bus state, such as for the presence of a start condition, is not possible during an IIC reset (ICE and IICRST bits = 01b in ICCR1).

For a detailed description of the IIC and internal resets, see section 32.15. State of Registers When Issuing Each Condition.

32.13 SMBus Operation

The IIC supports data communication conforming to the SMBus Specification (version 2.0). To perform SMBus communication, set the SMBS bit in ICMR3 to 1. To use the transfer rate within a range of 10 to 100 kbps of the SMBus standard, set the CKS[2:0] bits in ICMR1, the ICBRH, and ICBRL registers. In addition, specify the values in the DLCS bit in ICMR2 and the SDDL[2:0] bits in ICMR2 to meet the data hold time specification of 300 ns or more. When the IIC is used only as a slave device, the transfer rate setting is not required, but ICBRL must be set to a value longer than the data setup time (250 ns).

For the SMBus device default address (1100 001b), use one of the slave address registers L0 to L2 (SARL0, SARL1, and SARL2), and set the associated FS bit (7- or 10-bit address format select) in SARUy (y = 0 to 2) to 0 (7-bit address format).

When transmitting the UDID (Unique Device Identifier), set the SALE bit in ICFER to 1 to enable the slave arbitration-lost detection function.

32.13.1 SMBus Timeout Measurement

(1) Measuring slave device timeout

The following period (timeout interval: $T_{LOW:\ SEXT}$) must be measured for slave devices in SMBus communication:

- From start condition to stop condition.

To measure timeout for slave devices, measure the period from start condition detection to stop condition detection with the GPT using the IIC start condition detection interrupt (STIn) and stop condition detection interrupt (SPIn). The measured timeout period must be within the total clock low-level period [slave device] $T_{LOW:\ SEXT}$: 25 ms (maximum) of the SMBus standard.

If the time measured with the GPT exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (minimum) of the SMBus standard, the slave device must release the bus by writing 1 to the IICRST bit in ICCR1 to issue an internal reset of the IIC. When an internal reset is issued, the IIC stops driving the bus for the SCLn and SDAn pins, making them output high-impedance, which releases the bus.

(2) Measuring master device timeout

The following periods (timeout interval: $T_{LOW:\ MEXT}$) must be measured for master devices in SMBus communication:

- From start condition to acknowledge bit
- Between acknowledge bits
- From acknowledge bit to stop condition.

To measure timeout for master devices, measure these periods with the GPT using the IIC start condition detection interrupt (STIn), stop condition detection interrupt (SPIn), transmit end interrupt (IICn_TEI), or receive data full interrupt (IICn_RXI). The measured timeout period must be within the total clock low-level extended period (master device) $T_{LOW:\ MEXT}$: 10 ms (maximum) of the SMBus standard, and the total of all $T_{LOW:\ MEXT}$ values from start condition to stop condition must be within $T_{LOW:\ SEXT}$: 25 ms (maximum).

For the ACK receive timing (rising edge of the 9th SCL clock cycle), monitor the TEND flag in ICSR2 in master transmit mode (master transmitter) and the RDRF flag in ICSR2 in master receive mode (master receiver). Perform byte-wise transmit operations in master transmit mode, and hold the RDRFS bit in ICMR3 at 0 until the byte immediately before reception of the final byte in master receive mode. While the RDRFS bit is 0, the RDRF flag is set to 1 on the rising edge of the 9th SCL clock cycle.

If the period measured with the GPT exceeds the total clock low-level extended period (master device) $T_{LOW:\ MEXT}$: 10 ms (maximum) of the SMBus standard or the total of measured periods exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (minimum) of the SMBus standard, the master device must stop the transaction by issuing a stop condition. In master transmit mode, immediately stop the transmit operation (stop writing data to ICDRT).

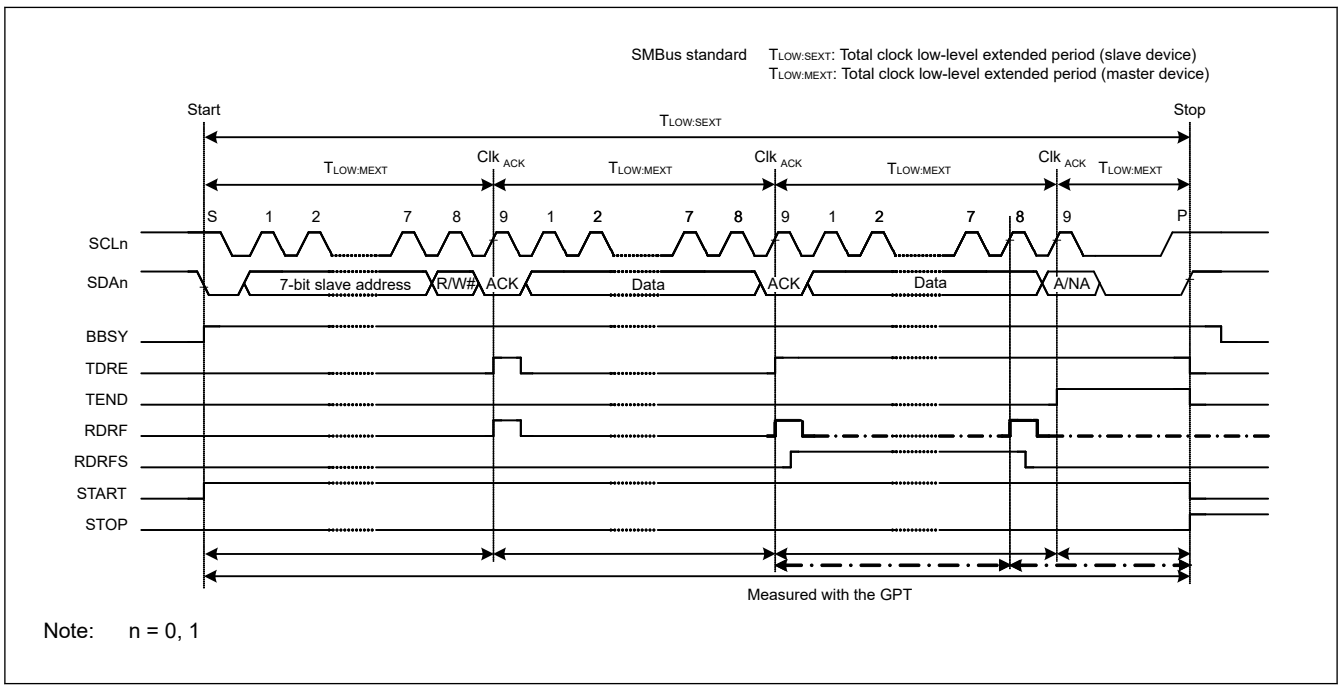


Figure 32.50 SMBus timeout measurement

32.13.2 Packet Error Code (PEC)

The MCU provides a CRC calculator that enables transmission of a Packet Error Code (PEC) or allows checking of the received data in SMBus data communication for the IIC. For the CRC-generating polynomials of the CRC calculator, see section 41, Cyclic Redundancy Check (CRC).

In master transmit mode, the PEC data can be generated by writing all transmit data to the CRC Data Input Register (CRCDIR) in the CRC calculator.

In master receive mode, the PEC data can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC Data Output Register (CRCDOR) with the received PEC data.

To send ACK or NACK based on the match or mismatch result when the final byte is received as a result of the PEC code check, set the RDRFS bit in ICMR3 to 1 before the rising edge of the 8th SCL clock cycle during reception of the final byte, and hold the SCLn line low on the falling edge of the 8th clock cycle.

32.13.3 SMBus Host Notification Protocol (Notify ARP Master Command)

In communicating over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of its own slave address, or to request its own slave address from the SMBus host.

For a product using the MCU to operate as an SMBus host or ARP master, the host address (0001 000b) sent from the slave device must be detected as a slave address, and so the IIC provides a function for detecting the host address. To detect the host address as a slave address, set the SMBS bit in ICMR3 and the HOAE bit in ICSER to 1. Operation after the host address is detected is the same as normal slave operation.

32.14 Interrupt Sources

The IIC issues five types of interrupt requests:

- Transfer error or event generation (arbitration-lost, NACK detection, timeout detection, start condition detection, and stop condition detection)
- Receive data full
- Transmit data empty
- Transmit end
- Address match during wakeup function

Table 32.10 lists details about the interrupt requests. The receive data full and transmit data empty interrupts can activate data transfer by the DTC or DMAC.

Table 32.10 Interrupt sources

Symbol	Interrupt source	Interrupt flag	DTC or DMAC activation	Interrupt condition
IICn_EEI*5	Transfer error or event occurrence	AL	Not possible	AL = 1, ALIE = 1
		NACKF		NACKF = 1, NAKIE = 1
		TMOF		TMOF = 1, TMOIE = 1
		START		START = 1, STIE = 1
		STOP		STOP = 1, SPIE = 1
IICn_RXI*2 *5	Receive data full	RDRF	Possible	RDRF = 1, RIE = 1
IICn_TXI*1 *5	Transmit data empty	RDRF	Possible	TDRE = 1, TIE = 1
IICn_TEI*3 *5	Transmit end	TEND	Not possible	TEND = 1, TEIE = 1
IIC0_WUI*4	Slave address match during wakeup function	WUF	Not possible	Slave address match Slave receive complete RWAK operation ASY0 = 1 WUIE = 1

Note: There is a delay between the execution of a write instruction for a peripheral module by the CPU and the actual writing to the module. When an interrupt flag is cleared or masked, read the relevant flag again to check whether clearing or masking is complete, then return from interrupt handling. Not doing so creates the possibility of repeated processing of the same interrupt.

Note 1. Because IICn_TXI is edge-detected, it does not require clearing. Additionally, the TDRE flag in ICSR2 (condition for IICn_TXI) is automatically set to 0 when transmit data is written to the ICDRT register or a stop condition is detected (STOP flag = 1 in ICSR2).
Note 2. Because IICn_RXI is edge-detected, it does not require clearing. Additionally, the RDRF flag in ICSR2 (condition for IICn_RXI) is automatically set to 0 when data is read from ICDRR.
Note 3. When using the IICn_TEI interrupt, clear the TEND flag in ICSR2 in the IICn_TEI interrupt handling. The TEND flag in ICSR2 automatically is set to 0 when transmit data is written to the ICDRT register or a stop condition is detected (STOP flag = 1 in ICSR2).
Note 4. Only channel 0 has a wakeup function, so IIC0_WUI is for channel 0 only.
Note 5. Channel number (n = 0 to 1).

Clear or mask each flag during interrupt handling.

32.14.1 Buffer Operation for IICn_TXI and IICn_RXI Interrupts

If the conditions for generating an IICn_TXI or IICn_RXI interrupt are satisfied while the associated IR flag is 1, the interrupt request is not output for the ICU but is saved internally. One request per source can be saved internally.

An interrupt request that is saved in the ICU is output when the ICU.IELSRn.IR flag becomes 0. Internally saved interrupt requests are automatically cleared under normal conditions. They can also be cleared by writing 0 to the interrupt enable bit within the associated peripheral module.

32.15 State of Registers When Issuing Each Condition

The IIC has two dedicated resets, IIC reset and Internal reset. Table 32.11 lists the registers states when issuing each condition.

Table 32.11 Register states when issuing each condition (1 of 2)

Registers		Reset	IIC reset (ICE = 0, IICRST = 1)	Internal reset (ICE = 1, IICRST = 1)	Start or restart condition detection	Stop condition detection
ICCR1	ICE, IICRST	In reset	Saved	Saved	Saved	Saved
	SCLO, SDAO		In reset	In reset		
	Others			Saved		
ICCR2	BBSY	In reset	In reset	Saved	Set	In reset
	ST, RS			In reset	In reset	Saved
	SP					In reset
	TRS				Set or saved	
	MST					
ICMR1	BC[2:0]	In reset	In reset	In reset	In reset	Saved
	Others			Saved	Saved	
ICMR2		In reset	In reset	Saved	Saved	Saved
ICMR3	ACKBT	In reset	In reset	Saved	Saved	In reset
	Others					Saved
ICFER		In reset	In reset	Saved	Saved	Saved
ICSER		In reset	In reset	Saved	Saved	Saved
ICIER		In reset	In reset	Saved	Saved	Saved
ICSR1		In reset	In reset	In reset	Saved	In reset
ICSR2	TEND	In reset	In reset	In reset	Saved	In reset
	TDRE				Set or saved	
	START				Set	
	STOP				Saved	Set
	Others				Saved	Saved
ICWUR		In reset	In reset	Saved	Saved	Saved

Table 32.11 Register states when issuing each condition (2 of 2)

Registers		Reset	IIC reset (ICE = 0, IICRST = 1)	Internal reset (ICE = 1, IICRST = 1)	Start or restart condition detection	Stop condition detection
SARL0, SARL1, SARL2 SARU0, SARU1, SARU2		In reset	In reset	Saved	Saved	Saved
ICBRH, ICBRL		In reset	In reset	Saved	Saved	Saved
ICDRT		In reset	In reset	Saved	Saved	Saved
ICDRR		In reset	In reset	Saved	Saved	Saved
ICDRS		In reset	In reset	In reset	Saved	Saved
Timeout function		In reset	In reset	In reset	Operating	Operating
Bus free time measurement		In reset	In reset	Operating	Operating	Operating
ICWUR2	WUSEN	In reset	In reset	Saved	Saved	Saved
	Others					Saved or set or reset

32.16 Event Link Output

The IIC0 module handles the event output for the Event Link Controller (ELC) for the following sources:

(1) Transfer error event

When a transfer error event occurs, the associated event signal can be output to another module by the ELC.

(2) Receive data full

When a receive data register becomes full, the associated event signal can be output to another module by the ELC.

(3) Transmit data empty

When a transmit data register becomes empty, the associated event signal can be output to another module by the ELC.

(4) Transmit end

On completion of the transfer, the associated event signal can be output to another module by the ELC.

32.16.1 Interrupt Handling and Event Linking

Each of the IIC interrupt types (see Table 32.10) has an enable bit to control enabling and disabling of the associated interrupt signal. An interrupt request signal is output to the CPU when an interrupt source condition is satisfied while the associated enable bit is set.

The associated event link output signals are sent to other modules as event signals by the ELC when the interrupt source conditions are satisfied, regardless of the interrupt enable bit settings. For details on interrupt sources, see Table 32.10.

32.17 Usage Notes

32.17.1 Settings for the Module-Stop Function

The Module Stop Control Register B (MSTPCRB) can enable or disable IIC operation. The IIC is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

32.17.2 Notes on Starting Transfer

If the IR flag associated with the IIC interrupt is 1 when transfer is started (ICCR1.ICE bit = 1), follow the procedure in this section to clear the interrupts before enabling operations. Starting transfer with the IR flag set to 1 while the ICCR1.ICE bit is 1 leads to an interrupt request being internally saved after transfer starts, and this can lead to unexpected behavior of the IR flag.

To clear interrupts before starting transfer operation:

1. Confirm that the ICCR1.ICE bit is 0.

2. Set the relevant interrupt enable bits, such as ICIER.TIE to 0.
3. Read the relevant interrupt enable bits, such as ICIER.TIE, and confirm that the value is 0.
4. Set the IR flag to 0.

33. I3C Bus Interface (I3C)

Specific HDR Modes are not covered by conformance test suite (CTS).

When CTS for HDR Modes is available, Renesas will confirm CTS tests.

33.1 Overview

33.1.1 Functional Overview

The I3C bus interface (I3C) has one channel. The I3C module conforms with and provides a subset of the NXP I^2C (Inter-Integrated Circuit) bus interface functions and a subset of the MIPI I3C.

In this section, PCLK refers to PCLKA, TCLK refers to I3CCLK.

Table 33.1 lists the I^2C specifications and Table 33.2 lists the I3C specifications.

Table 33.1 I^2C specifications (1 of 2)

Parameter	Specifications
Operation mode	Master mode and slave mode selectable
Data handler	Single buffer transfer
Communication protocol	• I^2C bus format – Standard-mode (Sm) : 0 to 100 kbps – Fast-mode (Fm) : 0 to 400 kbps – Fast-mode Plus (Fm+) : 0 to 1 Mbps – High-speed mode (Hs-mode) : 0 to 3.4 Mbps • SMBus format : 10 to 100 kbps
Address format	• 7-bit address • 10-bit address
Address detection	• Slave address (static address) (max 3 address) • General call address • Hs-mode master code • Device ID • Host address • 10-bit slave addressing
Clock stretching	Clock stretching capability
Noise-filter	• Analog noise-filter • Digital noise-filter
Interrupt source	• Normal Rx data buffer full • Normal Tx data buffer empty • START condition detection • STOP condition detection • Transmit end • NACK detection • Arbitration lost • Timeout detection • Wake-up condition detection
Error detection	• Non-recoverable internal error • NACK received • Receive overflow or transfer underflow error • Arbitration lost error • Timeout error
Event link output	• Normal Rx data buffer full event • Normal Tx data buffer empty event • START condition event • STOP condition event • Transmit end event • NACK event • Arbitration lost event • Timeout event
Wake-up source	Address detection of slave address

Table 33.1 I^2C specifications (2 of 2)

Parameter	Specifications
Module-stop function	Module-stop state can be set to reduce power consumption
Trust Zone Filter	Security and Privilege attribution can be set

Table 33.2 I3C specifications (1 of 2)

Parameter	Specifications
Operation mode	Master (main master/secondary master) mode and slave mode selectable
Data handler	• Master : – High priority FIFO buffer transfer – Normal FIFO buffer transfer • Slave : – Normal FIFO buffer transfer
Communication protocol	• SDR (I3C Single Data Rate) mode : up to 12.5 Mbps – Private message – Broadcast message (common command code) – Direct message (common command code) • HDR (I3C High Data Rate) mode – HDR-DDR (Dual Data Rate) mode : up to 25 Mbps – HDR-TSL (Ternary Symbol Legacy) mode : up to 27.5 Mbps – HDR-TSP (Ternary Symbol Pure-bus) mode : up to 39.5 Mbps • Legacy I^2C message – Fast-mode (Fm) : 0 to 400 kbps – Fast-mode Plus (Fm+) : 0 to 1 Mbps
In-band interrupt	• Slave interrupt request • Master ship request (secondary master only)
Address format	7-bit address
Address detection	• Slave address (static address or dynamic address) • Broadcast address (0x7E)
Clock stalling	Clock stalling capability
Timing control	• Synchronous timing control – Sync mode : Synchronous basic mode • Asynchronous timing control – Async mode 0 : Asynchronous basic mode – Async mode 1 : Asynchronous advanced mode
Interrupt source	• Non-recoverable internal error • Normal Transfer error • Normal Transfer abort • Normal Response queue full • Normal Command queue empty • Normal IBI Queue Empty/Full • Normal Rx Data buffer full • Normal Tx Data buffer empty • Normal Receive status queue full • High Priority Response queue full • High Priority Command queue empty • High Priority Rx Data buffer full • High Priority Tx Data buffer empty • High Priority Transfer Error • High Priority Transfer Abort • START condition detection • STOP condition detection • HDR exit pattern detection • Timeout detection • Synchronous Timing • MREF Counter Overflow • MREF Capture • Additional Master-initiated bus Event • Wake-up condition detection

Table 33.2 I3C specifications (2 of 2)

Parameter	Specifications
Error detection	• Non-recoverable internal error • CRC error • Parity error • Frame error • Address header error • Address NACKed or dynamic address assignment NACKed • Receive overflow or transfer underflow error • Aborted • NACK received for the I^2C write data transfer • Timeout error
Event link output	• Normal Response queue full event • Normal Command queue empty event • Normal IBI Queue Empty/Full • Normal Rx data buffer full event • Normal Tx data buffer empty event • Normal Receive Status queue full event • High Priority Response queue full event • High Priority Command queue empty event • High Priority Rx Data buffer full event • High Priority Tx Data buffer empty event • Non-recoverable internal error event • Normal Transfer Error event • Normal Transfer Abort event • High Priority Transfer Error event • High Priority Transfer Abort event • START condition event • STOP condition event • HDR exit pattern event • Timeout event • Synchronous timing event • MREF counter overflow event • MREF capture event • Additional master-initiated bus Event
Wake-up source	• Master : SDA assert of IBI (START condition detection) • Slave : Address detection of broadcast address (0x7E) and slave address
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security and Privilege attribution can be set

Table 33.3 I3C I/O pins

Channel	Pin name	I/O	Function
I3C	I3C_SCL0	I/O	I3C serial clock I/O pin
	I3C_SDA0	I/O	I3C serial data I/O pin

Note: In this section, I3C_SCL refers to I3C_SCL0, I3C_SDA refers to I3C_SDA0.

33.1.2 Block Diagram

Figure 33.1 shows the main components of this I3C.

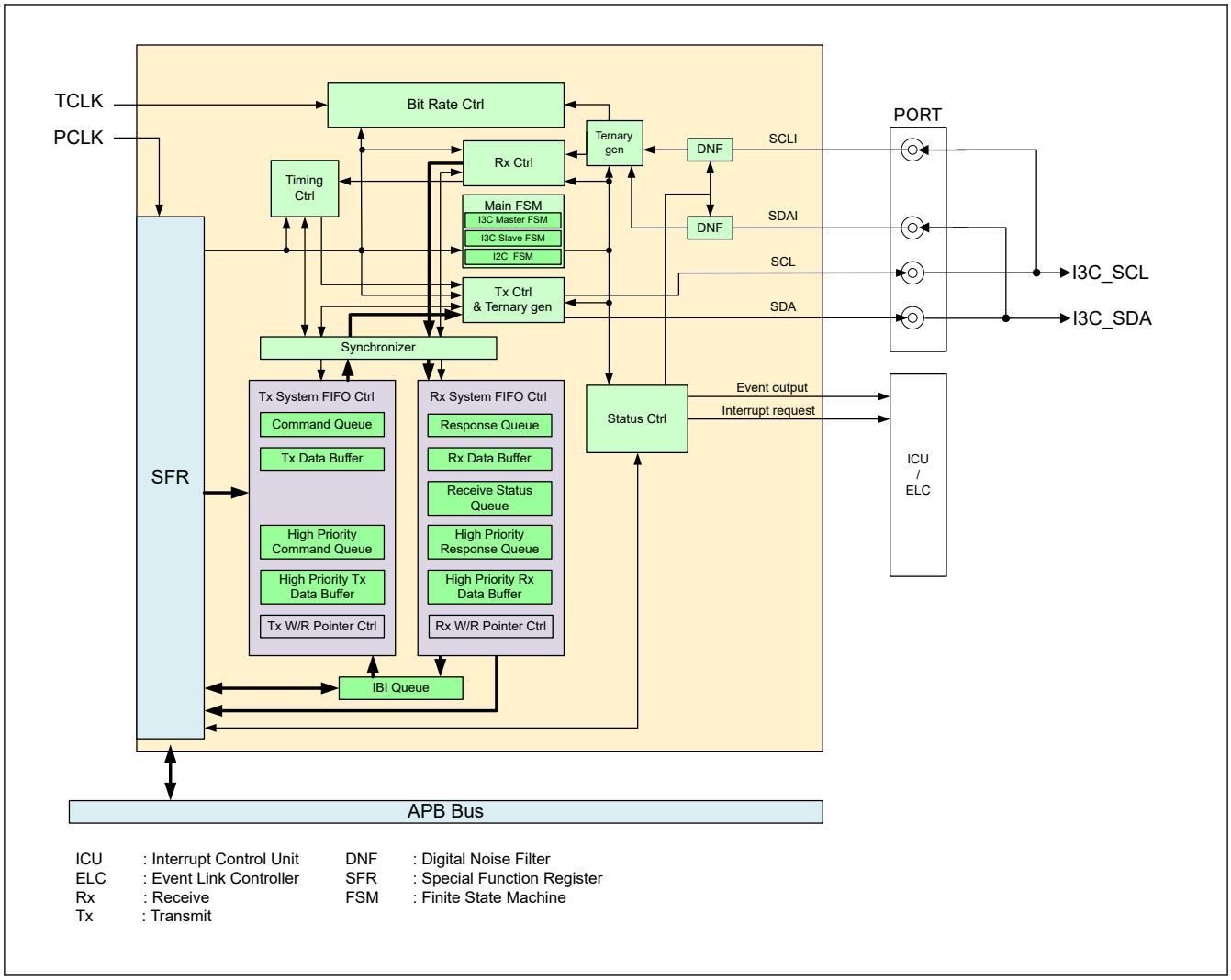


Figure 33.1 I3C block diagram

33.2 Registers

33.2.1 PRTS : Protocol Selection Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PRTMD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	PRTMD	Protocol Mode 0: I3C protocol mode 1: I^2C protocol mode	R/W

Bit	Symbol	Function	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

PRTMD bit (Protocol Mode)

PRTMD = 0 : I3C FIFO buffer transfer (Equivalent to HCI)

PRTMD = 1 : I^2C single buffer transfer

There are notes about rewriting the PRTS.PRTMD bit, see section 33.7.4. Notes on Rewriting the PRTS.PRTMD Bit.

33.2.2 BCTL : Bus Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BUSE	RSM	ABT	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	BMDS	—	—	—	—	—	—	INCBA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	INCBA	Include I3C Broadcast Address*1 0: Do not include I3C broadcast address for private transfers 1: Include I3C broadcast address for private transfers	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	BMDS	Bus Mode Selection*1 0: Legacy inclusive Bus mode disabled 1: Legacy inclusive (mix) Bus mode enabled	R/W
28:8	—	These bits are read as 0. The write value should be 0.	R/W
29	ABT	Abort*1 0: I3C is running. 1: I3C has aborted a transfer.	R/W
30	RSM	Resume*2 Values when read: 0: I3C is running. 1: I3C is suspended.	R/W
31	BUSE	Bus Enable 0: I3C bus operation is disabled. 1: I3C bus operation is enabled.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports I3C master mode and I3C secondary master mode.
Note 2. This bit supports all I3C mode.

INCBA bit (Include I3C Broadcast Address)

This bit controls whether the I3C broadcast address (0x7E) is included for private transfers.

If the I3C broadcast address is not included for private transfers, then IBIs driven from Slaves might not win the arbitration, potentially delaying acceptance of the IBIs.

BMDS bit (Bus Mode Selection)

Indicates whether or not any Legacy I^2C Devices are present on the I3C Bus. Whenever this bit is set, I3C shall use the HDR-TSL protocol (Ternary Symbol Legacy Inclusive Bus) for HDR-TS Transfers.

The specific mode of the transfer (SDR, DDR, HDRTS) is controlled on a per-Command basis.

I3C shall use this field to select either TSL (1) or TSP (0) for HDR-TS transfers.

ABT bit (Abort)

When set to 1, this bit allows I3C to relinquish control of the I3C Bus before completing the currently issued transfer.

In response to an ABORT request, I3C issues the STOP condition on the I3C Bus after the complete data byte is transferred or received.

The Driver shall clear the ABT bit to allow operation on the Bus.

If BCTL.ABT is set and ABORT processing is performed, please ignore ERR_STATUS of Response Descriptor.

RSM bit (Resume)

This bit is used to resume I3C operation following the Halt state.

I3C enters the Halt state (as indicated in register PRSTDBG) as a result of any type of error occurring in a transfer.

The error type is indicated by the field ERR_STATUS in register NRSPQP, HRSPQP, NRSQP and NIBIQP).

After I3C has entered the Halt state, the application must write the value 1 to the RSM bit to resume I3C operation. I3C shall auto-clear the RSM bit once it has resumed making transfers (it has initiated the next Command).

BUSE bit (Bus Enable)

Enables or disables the operation on the I3C Bus by I3C.

Set the BUSE bit to 1 when using I3C. The I3C_SCL and I3C_SDA pins are placed in the active state when the BUSE bit is set to 1. Set the BUSE bit to 0 when I3C is not to be used. The I3C_SCL and I3C_SDA pins are placed in the inactive state when the BUSE bit is set to 0.

If the software sets this bit, then it also confirms that initialization is done, and that I3C can use the programmed register values (for example, generation of SCL on IBI detection, etc.). If this bit is not set, then I3C shall not generate SCL for incoming IBI.

Software may disable I3C bus operation while it is active, However:

- If a disable request occurs while receiving IBI, the actual disabling will not occur until reception of the IBI is complete.
- When the software reads the value 0 from this field, this indicates that I3C bus operation disable operation has completed.
  If commands remain in the command queue, do not set BUSE = 0.

33.2.3 MSDVAD : Master Device Address Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x018

Bit position:	31	30	29	28	27	26	25	24	23	22						16
Bit field:	MDYADV	—	—	—	—	—	—	—	—	MDYAD[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
22:16	MDYAD[6:0]	Master Dynamic Address	R/W
30:23	—	These bits are read as 0. The write value should be 0.	R/W
31	MDYADV	Master Dynamic Address Valid 0: The master dynamic address field is not valid. 1: The master dynamic address field is valid.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode.

MDYAD[6:0] bits (Master Dynamic Address)

This field is used to program I3C master dynamic address. I3C uses this address to respond to master transactions in I3C interface mode (slave or secondary master role).

In I3C main master mode, the software shall program the dynamic address as it self-assigns its dynamic address.

MDYADV bit (Master Dynamic Address Valid)

This bit indicates whether or not the value in the MDYAD field is valid.

In I3C main master mode, the user sets this bit to 1 as it self-assigns its dynamic address.

Note: After setting MSDVAD, and setting BCTL. BUSE = 1, the device will act as main master.

Without setting MSDVAD, setting SVDCT.TBCR76[1:0] = 00b (Device Role Slave), and setting BCTL.BUSE = 1, the device will act as slave.

Without setting MSDVAD, setting MSDCTm.RBCR76[1:0] = 01b (Device Role Master), and setting BCTL.BUSE = 1, the device will act as slave.

33.2.4 RSTCTL : Reset Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INTLRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	HRDBRST	HTDBRST	HRSPQRST	HCMDQRST	—	—	RSQRST	IBIQRST	RDBRST	TDBRST	RSPQRST	CMDQRST	RI3CRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RI3CRST	I3C Software Reset 0: Release I3C reset. 1: Initiate I3C reset.	R/W
1	CMDQRST	Normal Command Queue Software Reset*1 0: The Normal Command Queues in I3C is not flushed. 1: The Normal Command Queues in I3C is flushed.	R/W
2	RSPQRST	Normal Response Queue Software Reset*1 0: The Normal Response Queues in I3C is not flushed. 1: The Normal Response Queues in I3C is flushed.	R/W
3	TDBRST	Normal Tx Data Buffer Software Reset*1 0: The Normal Tx Data buffers in I3C is not flushed. 1: The Normal Tx Data buffers in I3C is flushed.	R/W

Bit	Symbol	Function	R/W
4	RDBRST	Normal Rx Data Buffer Software Reset*1 0: The Normal Rx Data buffers in I3C is not flushed. 1: The Normal Rx Data buffers in I3C is flushed.	R/W
5	IBIQRST	Normal IBI Queue Software Reset*1 0: The Normal IBI Queues in I3C is not flushed. 1: The Normal IBI Queues in I3C is flushed.	R/W
6	RSQRST	Normal Receive Status Queue Software Reset*2 0: The Normal Receive Status Queue in I3C is not flushed. 1: The Normal Receive Status Queue in I3C is flushed.	R/W
8:7	—	These bits are read as 0. The write value should be 0.	R/W
9	HCMDQRST	High Priority Command Queue Software Reset*3 0: The High Priority Command Queues in I3C is not flushed. 1: The High Priority Command Queues in I3C is flushed.	R/W
10	HRSPQRST	High Priority Response Queue Software Reset*3 0: The High Priority Response Queues in I3C is not flushed. 1: The High Priority Response Queues in I3C is flushed.	R/W
11	HTDBRST	High Priority Tx Data Buffer Software Reset*3 0: The High Priority Tx Data buffers in I3C is not flushed. 1: The High Priority Tx Data buffers in I3C is flushed.	R/W
12	HRDBRST	High Priority Rx Data Buffer Software Reset*3 0: The High Priority Rx Data buffers in I3C is not flushed. 1: The High Priority Rx Data buffers in I3C is flushed.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	INTLRST	Internal Software Reset 0: Releases of some registers and internal state. 1: Resets of some registers and internal state.	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.
Note 2. This bit supports I3C secondary master mode and I3C slave mode.
Note 3. This bit supports I3C master mode and I3C secondary master mode.

For details on reset for each register, see section 33.6. Reset Descriptions.

RI3CRST bit (I3C Software Reset)

On Driver setting this bit to 1, I3C shall be reset and disabled.

All registers shall return to their reset values, and the software shall re-initialize I3C.

This field is cleared automatically upon I3C reset completion. This field also resets all Queues in I3C.

Note: Programming this field while it contains a value of 1 may result in undefined behavior.

CMDQRST bit (Normal Command Queue Software Reset)

On software setting this bit to 1, the Normal Command Queues in I3C shall be flushed.

This field shall be cleared automatically upon Normal Command Queue reset completion.

RSPQRST bit (Normal Response Queue Software Reset)

On software setting this bit to 1, the Normal Response Queues in I3C shall be flushed.

This field shall be cleared automatically upon Normal Response Queue reset completion.

TDBRST bit (Normal Tx Data Buffer Software Reset)

On software setting this bit to 1, the Normal Tx Data Buffers in I3C shall be flushed.

This field shall be cleared automatically upon Normal Tx Data Buffer reset completion.

RDBRST bit (Normal Rx Data Buffer Software Reset)

On software setting this bit to 1, the Normal Rx Data Buffers in I3C shall be flushed.

This field shall be cleared automatically upon completion of Normal Rx Data Buffer reset.

IBIQRST bit (Normal IBI Queue Software Reset)

On software setting this bit to 1, the Normal IBI Queues in I3C shall be flushed.

This field shall be cleared automatically upon completion of Normal IBI Queue reset.

RSQRST bit (Normal Receive Status Queue Software Reset)

On software setting this bit to 1, the Normal Receive Status Queues in I3C shall be flushed.

This field shall be cleared automatically upon Normal Receive Status Queue reset completion.

HCMDQRST bit (High Priority Command Queue Software Reset)

On software setting this bit to 1, the High Priority Command Queues in I3C shall be flushed.

This field shall be cleared automatically upon High Priority Command Queue reset completion.

HRSPQRST bit (High Priority Response Queue Software Reset)

On software setting this bit to 1, the High Priority Response Queues in I3C shall be flushed.

This field shall be cleared automatically upon High Priority Response Queue reset completion.

HTDBRST bit (High Priority Tx Data Buffer Software Reset)

On software setting this bit to 1, the High Priority Tx Data Buffers in I3C shall be flushed.

This field shall be cleared automatically upon High Priority Tx Data Buffer reset completion.

HRDBRST bit (High Priority Rx Data Buffer Software Reset)

On software setting this bit to 1, the High Priority Rx Data Buffers in I3C shall be flushed.

This field shall be cleared automatically upon completion of High Priority Rx Data Buffer reset.

INTLRST bit (Internal Software Reset)

When set to 1, some of registers is reset. For details on the registers to be reset, see section 33.6. Reset Descriptions.

Note: When set internal software reset during bus operation enable, use DISEC CCC in advance to disable IBI transmission to I3C Slave in order to avoid conflict with IBI from I3C Slave connected to I3C Bus.

33.2.5 PRSST : Present State Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x024

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	PRSS TWP	—	—	TRMD	—	CRMS	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
2	CRMS	Current Master*2 0: The Master is not the Current Master, and must request and acquire bus ownership before initiating any transfer. 1: The Master is the Current Master, and as a result can initiate transfers.	R/W*1
3	—	This bit is read as 0. The write value should be 0.	R/W
4	TRMD	Transmit/Receive Mode*3 0: Receive mode 1: Transmit mode	R
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	PRSSTWP	Present State Write Protect*2 0: CRMS bit is protected. 1: CRMS bit can be written when writing simultaneously with the value of the target bit.	W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. When the PRSSTWP bit is set to 1, the CRMS bit can be written to.
Note 2. This bit supports I2C, I3C master, and I3C secondary master mode.
Note 3. This bit supports I2C mode.

CRMS bit (Current Master)

Indicates the set condition and reset condition of each operation mode.

Operation Mode [I2C/I3C common]

[Clearing conditions]

- When 0 written to the PRSST.CRMS by the software.

[Setting conditions]

- When 1 written to the PRSST.CRMS by the software.

Operation Mode [I2C]

[Clearing conditions]

- When STOP is issued.
- When Master Arbitration-Lost.

[Setting conditions]

- When START is issued.

Operation Mode [I3C Main Master]

[Clearing conditions]

- When 0 written to the MSDVAD.MDYADV by the software.
- When GETACCMST transmission is successfully completed by issuing STOP, after responding ACK to the Mastership-Request received from the Secondary Master.

[Setting conditions]

- When 1 written to the MSDVAD.MDYADV by the software.
- When GETACCMST reception is successfully completed by issuing STOP, after the ACK is responded to the Mastership-Request transmitted to the Secondary Master.

Operation Mode [I3C Secondary Master]

[Clearing condition]

- When GETACCMST transmission is successfully completed by issuing STOP, after responding ACK to the Mastership-Request received from the Non-Current Master.

[Setting condition]

- When GETACCMST reception is successfully completed by issuing STOP, after the ACK is responded to the Mastership-Request transmitted to the Current Master.

The PRSST register returns I3C current state.

State has two parts: this register which is mandatory, and an additional optional PRSST_DEBUG register intended for debug purposes (see the Debug Capability registers in the Extended Capabilities list).

TRMD bit (Transmit/Receive Mode)

This bit indicates transmit or receive mode.

I3C is in receive mode when the TRMD bit is set to 0 and is in transmit mode when the bit is set to 1. Combination of this bit and the CRMS bit indicates the operating mode of I3C.

The value of TRMD bit is automatically changed to 1 for transmit mode or 0 for receive mode by issuing or detection of a START condition and setting of the R/W# bit.

[Setting conditions]

- When a START condition is issued normally according to the START condition issuance request (when a START condition is detected with the CNDCTL.STCND bit set to 1).
- When a Repeated START condition is issued normally according to the Repeated START condition issuance request (when a Repeated START condition is detected with the CNDCTL.SRCND bit set to 1).
- When the R/W# bit added to the slave address is set to 0 in master mode.
- When the address received in slave mode matches the address enabled in SVCTL, with the R/W# bit set to 1.

[Clearing conditions]

- When a STOP condition is detected.
- The ALF (arbitration-lost) flag in BST being set to 1.
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended.
- In slave mode, a match between the received address and the address enabled in SVCTL when the value of the received R/W# bit is 0 (including cases where the received address is the general call address).
- In slave mode, a Repeated START condition is detected (a Repeated START condition is detected with BCST.BFREF = 0 and CRMS = 0).

PRSSTWP bit (Present State Write Protect)

PRSSTWP is always 0 when reading.

When writing to PRSST, writing 1 to this bit at the same time enables writing to CRMS bit.

33.2.6 INST : Internal Status Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x030

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	INEF	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	—	These bits are read as 0. The write value should be 0.	R/W
10	INEF	Internal Error Flag 0: I3C Internal Error has not detected. 1: I3C Internal Error has detected.	R/W*1
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.
Note 1. Clearing (to 0) condition : Writing 0 after the 1 state is read.

The Interrupt Status register reflects the status of outstanding interrupts.

The status fields are either write 0 to clear, or else are cleared based on queue operations.

INEF bit (Internal Error Flag)

When this bit is 1, it indicates that I3C Internal Error has detected.

When this bit is 0, it indicates that I3C Internal Error has not detected.

[Setting conditions]

- The following 1 is satisfied and any of the following 2 to 9 are satisfied.
  1. The INSTE.INEE bit = 1
  2. When transmit data is written to the Tx Data Buffer that is completely full.
  3. When received data is read from the Rx Data Buffer that is completely empty.
  4. When Command Descriptor is written to the Command Queue that is completely full.
  5. When Response Descriptor is read from the Response Queue that is completely empty.
  6. When Receive Status Descriptor is read from the Receive Status Queue that is completely empty.
  7. When IBI Status Descriptor is read from the IBI Queue under the condition that the IBI Queue is completely empty and PRSST.CRMS = 1.
  8. When IBI Data is written to the IBI Queue under the condition that the IBI Queue is completely full and PRSST.CRMS = 0.
  9. When the Response Queue, IBI Queue or Receive Status Queue overflows.

[Clearing condition]

- When 0 is written to the INEF bit after reading INEF bit = 1.

33.2.7 INSTE : Internal Status Enable Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x034

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	INEE	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
10	INEE	Internal Error Enable 0: Disable INST.INEF 1: Enable INST.INEF	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

INEE bit (Internal Error Enable)

When this bit set to 1, it enables detection of I3C Internal Error.

When this bit set to 0, it disables detection of I3C Internal Error.

33.2.8 INIE : Internal Interrupt Enable Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x038

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	INEIE	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	—	These bits are read as 0. The write value should be 0.	R/W
10	INEIE	Internal Error Interrupt Enable 0: Disables Non-recoverable Internal Error Interrupt Signal. 1: Enables Non-recoverable Internal Error Interrupt Signal.	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

INEIE bit (Internal Error Interrupt Enable)

When set to 1 and register INEF is set, the hardware Controller asserts an interrupt to the Host.

33.2.9 INSTFC : Internal Status Force Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x03C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	INEFC	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	—	The write value should be 0.	W
10	INEFC	Internal Error Force 0: Not force a specific interrupt 1: Force a specific interrupt	W
31:11	—	The write value should be 0.	W

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

INEFC bit (Internal Error Force)

For debug, helps to force this interrupt.

33.2.10 DVCT : Device Characteristic Table Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x044

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	IDX[4:0]					—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
18:0	—	These bits are read as 0.	R
23:19	IDX[4:0]	DCT Table Index Current index of the DCT, which is used as the starting index for the I3C ENTDAA CCC.	R
31:24	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

IDX[4:0] bits (DCT Table Index)

Once the complete characteristics of device that won the arbitration are written to the DCT (during ENTDAA using Address Assignment Command) this index is incremented by 1.

Note: How to check the progress of ENTDAA using this bit:

1. Read the value of this bit before setting the Command Descriptor for issuing the ENTDAA command.
2. After starting the ENTDAA command, until the value of this bit is updated (that is, it changes from the value read in advance), it indicates that the Dynamic Address is being assigned to the device specified by the first index value (value set in DEV_INDEX[4:0] of Command Descriptor).
3. After the value of this bit is updated, it indicates that Dynamic Address is being assigned according to the value set in DEV_INDEX[4:0] and DEV_COUNT[3:0] of Command Descriptor to the device of the first index value or later.

33.2.11 IBINCTL : IBI Notify Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x058

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	NRSIRCTL	—	NRMRCTL	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	NRMRCTL	Notify Rejected Master Request Control 0: Do not pass rejected IBI Status to the Normal IBI Queue, if the incoming Master Request is NACKed and is auto-disabled based on DVMRRJ field in relevant DAT entry. 1: Pass rejected IBI Status to the Normal IBI Queue if the incoming Master Request is NACKed and is autodisabled based on DVMRRJ field in relevant DAT entry.	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	NRSIRCTL	Notify Rejected Slave Interrupt Request Control 0: Do not pass rejected IBI Status to the Normal IBI Queue, if the incoming SIR is NACKed and is auto-disabled based on DVSIRRJ field in relevant DAT entry. 1: Pass rejected IBI Status to the Normal IBI Queue, if the incoming SIR is NACKed and is auto-disabled based on DVSIRRJ field in relevant DAT entry.	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

NRMRCTL bit (Notify Rejected Master Request Control)

Enables or disables reporting rejection of individual Master Requests.

NRSIRCTL bit (Notify Rejected Slave Interrupt Request Control)

Enables or disables reporting rejection of individual Slave Interrupt Requests (SIR).

33.2.12 BFCTL : Bus Function Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x060

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	HSME	FMPE	—	SMBS	—	—	—	SCSYNE	—	—	—	—	—	SALE	NALE	MALE
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	MALE	Master Arbitration-Lost Detection Enable 0: Master arbitration-lost detection disables. Disables the arbitration-lost detection function and does not clear the CRMS and TRMD bits in PRSST automatically when arbitration is lost. 1: Master arbitration-lost detection enables. Enables the arbitration-lost detection function and clears the CRMS and TRMD bits in PRSST automatically when arbitration is lost.	R/W
1	NALE	NACK Transmission Arbitration-Lost Detection Enable 0: NACK transmission arbitration-lost detection disables. 1: NACK transmission arbitration-lost detection enables.	R/W
2	SALE	Slave Arbitration-Lost Detection Enable 0: Slave arbitration-lost detection disables. 1: Slave arbitration-lost detection enables.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	SCSYNE	SCL Synchronous Circuit Enable 0: No SCL synchronous circuit uses. 1: An SCL synchronous circuit uses.	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
12	SMBS	SMBus/I^2C Bus Selection 0: The I^2C bus select. 1: The SMBus select.	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
14	FMPE	Fast-mode Plus Enable 0: No Fm+ slope control circuit uses for the I3C_SCL pin and I3C_SDA pin. (n = 0) 1: An Fm+ slope control circuit uses for the I3C_SCL pin and I3C_SDA pin. (n = 0)	R/W
15	HSME	High Speed Mode Enable 0: Disable High Speed Mode. 1: Enable High Speed Mode.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports for I2C mode.

MALE bit (Master Arbitration-Lost Detection Enable)

This bit is used to specify whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

NALE bit (NACK Transmission Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode (such as when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with different number of receive bytes).

SALE bit (Slave Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode (such as when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs due to noise).

SCSYNE bit (SCL Synchronous Circuit Enable)

This bit is used to specify whether to synchronize the SCL clock with the SCL input clock. Normally, set this bit to 1.

When the SCSYNE bit set to 0 (no SCL synchronous circuit used), I3C does not synchronize the SCL clock with the SCL input clock. In this setting, I3C outputs the SCL clock with the transfer rate set in STDBR and EXTBR regardless of the I3C_SCL line state. For this reason, if the bus load of the I^2C bus line is much larger than the specification value or if the SCL clock output overlaps in multiple masters, the short-cycle SCL clock that does not meet the specification may be output. When no SCL synchronous circuit uses, it also affects the issuance of a START condition, Repeated START condition, and STOP condition, and the continuous output of extra SCL clock cycles.

This bit must not be set to 0 except for checking the output of the set transfer rate.

FMPE bit (Fast-mode Plus Enable)

This bit is used to specify whether to use a slope control circuit for Fast-mode Plus [Fm+].

When this bit is set to 1, a slope control circuit conforming to the Fast-mode Plus [Fm+] slope control specification (tof) of the I3C-bus is selected. When this bit is set to 0, a slope control circuit conforming to the Standard-mode [Sm] and Fast-mode [fm] slope control specification (tof) of the I3C-bus is selected.

Set this bit to 1 when using the transmission rate within a range up to 1 Mbps (Fast-mode Plus [Fm+]) of the I3C-bus specification. Set this bit to 0 when using the transmission rate at other rates (up to 100 kbps [Sm], up to 400 kbps [Fm]) or for SMBus (10 to 100 kbps).

Note: When communicating in Hs-mode, set as follows.
- Set FMPE to 0 when sending Hs-mode master code (0000 1XXXb) with Fast-mode.
- Set FMPE to 1 when sending Hs-mode master code (0000 1XXXb) with Fast-mode Plus.

HSME bit (High Speed Mode Enable)

This bit is used for communicating in Hs-mode.

When this bit is set to 1, the Hs-mode master code is recognized and Hs-mode communication is possible.

After the START condition is detected, if Hs-mode master code (0000 1XXXb) transmission is recognized, Hs-mode communication starts from Repeated START after receiving the NACK response.

It communicates at the bit rate set in STDBR until the NACK response, and automatically switches from Repeated START condition issuance after receiving the NACK response to the bit rate set in EXTBR.

Hs-mode continues until a STOP condition is detected.

When the STOP condition is detected, the bit rate is automatically switched to the bit rate set in STDBR.

Note: When this bit is set to 1, the BST.NACKDF bit will not be set even if a NACK response is received after sending the Hs-mode master code.

33.2.13 SVCTL : Slave Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x064

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18		16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SVAE[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	HOAE	—	—	—	—	—	—	—	—	DVIDE	HSMCE	—	—	—	—	GCAE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	GCAE	General Call Address Enable*1 0: General call address detection disables. 1: General call address detection enables.	R/W
4:1	—	These bits are read as 0. The write value should be 0.	R/W
5	HSMCE	Hs-mode Master Code Enable*1 0: Hs-mode Master Code Detection disables. 1: Hs-mode Master Code Detection enables.	R/W
6	DVIDE	Device-ID Address Enable*1 0: Device-ID address detection disables. 1: Device-ID address detection enables.	R/W
14:7	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15	HOAE	Host Address Enable*1 0: Host address detection disables. 1: Host address detection enables.	R/W
18:16	SVAE[2:0]	Slave Address Enable n (n = 0 to 2)*2 0: Slave n disables 1: Slave n enables	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports I2C mode.
Note 2. These bits support I2C, I3C secondary master, and I3C slave mode.

GCAE bit (General Call Address Enable)

This bit is used to specify whether to ignore the general call address (0000 000 + 0 (write): All 0) when it is received. When this bit is set to 1, if the received slave address matches the general call address, I3C recognizes the received slave address as the general call address independently of the slave addresses set in the SVDVADn.SVAD[9:0] bits (n = 0 to 2) and performs data receive operation.

When this bit is set to 0, the received slave address is ignored even if it matches the general call address.

HSMCE bit (Hs-mode Master Code Enable)

This bit is used to specify whether to recognize and execute the Hs-mode master code (00001xxxb) is received in the first byte after a START condition is detected.

When this bit is set to 1, if the received first byte matches the Hs-mode master code, I3C recognizes that the Hs-mode master code has been received.

The first byte after Repeated START after NACK response to Hs-mode master code is recognized as a slave address and compared with the slave address set by SVDVADn.SVAD[9:0]bits (n = 0 to 2).

If the addresses match, the transmission / reception operation continues according to the R/W# bit value.

Hs-mode continues until a STOP condition is detected.

When this bit is set to 0, I3C will ignore the pattern until a STOP condition is detected, even if it matches the Hs-mode master code.

Note: When this bit is set to 1, SCSTRCTL.ACKTWE bit must be set to 0 and SCSTRCTL.RWE bit must be set to 1.

DVIDE bit (Device-ID Address Enable)

This bit is used to specify whether to recognize and execute the Device-ID address when a device ID (1111 100) is received in the first byte after a START condition or Repeated START condition is detected.

When this bit is set to 1, if the received first byte matches the Device-ID, I3C recognizes that the Device-ID address has been received. When the following R/W# bit is 0 (write), I3C recognizes the second and the following bytes as slave addresses and continues the receive operation.

When this bit is set to 0, I3C ignores the received first byte even if it matches the Device ID address and recognizes the first byte as a normal slave address.

For details on the Device-ID address detection, see (3) Device-ID Address Detection [I2C mode].

HOAE bit (Host Address Enable)

This bit is used to specify whether to ignore received host address (0001 000) when the BFCTL.SMBS bit = 1.

When this bit is set to 1 while the SMBS bit = 1, if the received slave address matches the host address, I3C recognizes the received slave address as the host address independently of the slave addresses set in the SVDVADn.SVAD[9:0] bits (n = 0 to 2) and performs the receive operation.

When the SMBS bit or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

SVAE[2:0] bits (Slave Address Enable n (n = 0 to 2))

This bit is used to enable or disable the slave address set in the SVDVADn.SVAD[9:0] bits.

When this bit is set to 1, the slave address set in the SVAD[9:0] bits is enabled and is compared with the received slave address.

When this bit is set to 0, the slave address set in the SVAD[9:0] bits is disabled and is ignored even if it matches the received slave address.

33.2.14 REFCKCTL : Reference Clock Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x070

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	IREFCKS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	IREFCKS[2:0]*1	Internal Reference Clock Selection Selects the internal reference clock source (I3Cφ) for I3C. 0 0 0: TCLK/1 clock 0 0 1: TCLK/2 clock 0 1 0: TCLK/4 clock 0 1 1: TCLK/8 clock 1 0 0: TCLK/16 clock 1 0 1: TCLK/32 clock 1 1 0: TCLK/64 clock 1 1 1: TCLK/128 clock	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. Set the IREFCKS[2:0] bit to 000 in I3C mode.

33.2.15 STDBR : Standard Bit Rate Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x074

Bit position:	31	30	29					24	23	22	21					16
Bit field:	DSBRPO	—	SBRHP[5:0]						—	—	SBRLP[5:0]					
Value after reset:	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1

Bit position:	15							8	7							0
Bit field:	SBRHO[7:0]								SBRLO[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	SBRLO[7:0]	Standard Bit Rate Low-Level Period Open-Drain Count value of the low-level period of SCL clock*1	R/W
15:8	SBRHO[7:0]	Standard Bit Rate High-Level Period Open-Drain Count value of the high-level period of SCL clock*1	R/W

Bit	Symbol	Function	R/W
21:16	SBRLP[5:0]	Standard Bit Rate Low-level Period Push-Pull*2 Count value of the low-level period of SCL clock	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
29:24	SBRHP[5:0]	Standard Bit Rate High-Level Period Push-Pull*3 Count value of the high-level period of SCL clock	R/W
30	—	This bit is read as 0. The write value should be 0.	R/W
31	DSBRPO	Double the Standard Bit Rate Period for Open-Drain*4 0: The time period set for SBRHO[7:0] and SBRLO[7:0] is not doubled. 1: The time period set for SBRHO[7:0] and SBRLO[7:0] is doubled.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. These bits support I2C, I3C master, and I3C secondary master mode.
Note 2. These bits support I3C master mode and I3C secondary master mode.
Note 3. These bits support all I3C mode.
Note 4. This bit supports I2C, I3C master, and I3C secondary master mode.

The STDBR register sets the bit rate according to the operating speed.

- I2C mode: Bit rate setting when communicating with Standard-mode / Fast-mode / Fast-mode plus
- I3C master mode: Bit rate setting selected by MODE bit of command descriptor
- I3C slave mode: I3C bit rate setting

The I2C transfer rate and the SCL clock duty are calculated using the following expression.

Transfer rate = 1 / {[(High-Level Period + α*1) + (Low-Level Period + α)] / I3Cφ*2 + I3C_SCL line rising time [tr]*3 + I3C_SCL line falling time [tf]*3}

Duty cycle = {I3C_SCL line rising time [tr] + (High-Level Period + α) / I3Cφ} / {I3C_SCL line falling time [tf] + (Low- Level Period + α) / I3Cφ}

Note 1. α depend on the number of stages in the noise filter.
Note 2. I3Cφ = TCLK × Division ratio
Note 3. The I3C_SCL line rising time [tr] and I3C_SCL line falling time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, see the I2C-bus specification from NXP Semiconductors.

The I3C transfer rate and the SCL clock duty are calculated using the following expression.

Transfer rate = 1 / [(High-Level Period + Low-Level Period) / I3Cφ + I3C_SCL line rising time [tr] + I3C_SCL line falling time [tf]]

Duty cycle = [I3C_SCL line rising time [tr] + High-Level Period / I3Cφ] / [I3C_SCL line falling time [tf] + Low-Level Period / I3Cφ]

SBRLO[7:0] bits (Standard Bit Rate Low-Level Period Open-Drain)

The SBRLO[7:0] bits are used to set the low-level period of SCL clock in Open-Drain mode.

I3C counts the low-level period with the internal reference clock source (I3Cφ) specified by the REFCKCTL.IREFCKS[2:0] bits. It also works to generate the data setup time for automatic SCL low-hold operation (see section 33.3.2.3.6. Clock Stretching [I2C mode]); when I3C is used in I2C slave mode, these bits need to be set to a value longer than the data setup time*1.

If the digital noise filter is enabled (INCTL.DNFE = 1), set the SBRLO[7:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the INCTL.DNFS[3:0] bits.

Note 1. Data setup time (tSU: DAT)
250 ns (up to 100 kbps: Standard-mode [Sm])
100 ns (up to 400 kbps: Fast-mode [Fm])
50 ns (up to 1 Mbps: Fast-mode plus [Fm+])
10 ns (up to 3.4 Mbps: Hs-mode [HS])

SBRHO[7:0] bits (Standard Bit Rate High-Level Period Open-Drain)

The SBRHO[7:0] bits use to set the high-level period of SCL clock in Open-Drain mode. SBRHO[7:0] bits are valid in master mode. If I3C is used only in I^2C slave mode, these bits need not to set the high-level period.

I3C counts the high-level period with the internal reference clock source (I3Cφ) specified by the REFCKCTL.IREFCKS[2:0] bits.

If the digital noise filter is enabled (the INCTL.DNFE bit = 1), set the SBRHO[7:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the INCTL.DNFS[3:0] bits.

SBRLP[5:0] bits (Standard Bit Rate Low-level Period Push-Pull)

SBRLP[5:0] bits are used to set the low-level period of SCL clock in Push-Pull.

I3C counts the low-level period with the internal reference clock source (I3Cφ) specified by the REFCKCTL.IREFCKS[2:0] bits.

If the digital noise filter is enabled (the INCTL.DNFE bit = 1), set the SBRLP[5:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the INCTL.DNFS[3:0] bits.

SBRHP[5:0] bits (Standard Bit Rate High-Level Period Push-Pull)

SBRHP[5:0] bits is used to set the high-level period of SCL clock in Push-Pull mode.

SBRHP[5:0] bits are valid in master mode. If I3C is used only in I^2C slave mode, these bits need not to set the high-level period.

I3C counts the high-level period with the internal reference clock source (I3Cφ) specified by the REFCKCTL.IREFCKS[2:0] bits.

If the digital noise filter is enabled (the INCTL.DNFE bit = 1), set the SBRHP[5:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the INCTL.DNFS[3:0] bits.

DSBRPO bit (Double the Standard Bit Rate Period for Open-Drain)

When DSBRPO = 1, double the high-level period that is set in SBRHO[7:0] and double the low-level period that is set in SBRLO[7:0].

Table 33.4 Requirement and usage of setting in each mode

Bit name	Device mode				
	I^2C master	I^2C slave	I3C Master	I3C Secondary Master	I3C Slave
SBRHP[5:0]	do not use	do not use	Setting required*3	Setting required*4	Setting required*6
SBRLP[5:0]	do not use	do not use	Setting required*3	Setting required*5	do not use
SBRHO[7:0]	Setting required*1	do not use	Setting required*3	Setting required*5	do not use
SBRLO[7:0]	Setting required*1	Setting required*2	Setting required*3	Setting required*5	do not use

Note 1. The setting value is used for the data rate of ST, FM, and FM+ mode.
Note 2. The setting value is used for the data setup time of automatic SCL low-hold operation.
Note 3. The setting value is used for the data rate of each communication.
Note 4. When operating with I3C Master, the setting value is used for the data rate of each communication.
When operating with I3C Slave, the setting value is used for the data rate of HDR-TSP/TSL mode.
Note 5. When operating with I3C Master, the setting value is used for the data rate of each communication.
When operating with I3C Slave, do not use.
Note 6. The setting value is used for the data rate of HDR-TSP/TSL mode.

33.2.16 EXTBR : Extended Bit Rate Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x078

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	EBRHP[5:0]						—	—	EBRLP[5:0]						EBRHO[7:0]								EBRLO[7:0]							
Value after reset:	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	EBRLO[7:0]	Extended Bit Rate Low-Level Period Open-Drain*1 Count value of the low-level period of SCL clock	R/W
15:8	EBRHO[7:0]	Extended Bit Rate High-Level Period Open-Drain*1 Count value of the high-level period of SCL clock	R/W
21:16	EBRLP[5:0]	Extended Bit Rate Low-Level Period Push-Pull*2 Count value of the low-level period of SCL clock	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
29:24	EBRHP[5:0]	Extended Bit Rate High-Level Period Push-Pull*2 Count value of the high-level period of SCL clock	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. These bits support I2C, I3C master, and I3C secondary master mode.
Note 2. These bits support I3C master mode and I3C secondary master mode.

The EXTBR register sets the bit rate according to the operating speed.

- I2C mode: Bit rate setting for communicating in high-speed mode
- I3C master mode: Bit rate setting selected by MODE bit of command descriptor
- I3C slave mode: unused

EBRLO[7:0] bits (Extended Bit Rate Low-Level Period Open-Drain)

See SBRLO[7:0] bits of section 33.2.15. STDBR : Standard Bit Rate Register for details. Watch SBRHO, SBRLO as EBRHO[7:0], EBRLO[7:0].

EBRHO[7:0] bits (Extended Bit Rate High-Level Period Open-Drain)

See SBRHO[7:0] bits of section 33.2.15. STDBR : Standard Bit Rate Register for details. Watch SBRHO, SBRLO as EBRHO[7:0], EBRLO[7:0].

EBRLP[5:0] bits (Extended Bit Rate Low-Level Period Push-Pull)

See SBRLP[5:0] bits of section 33.2.15. STDBR : Standard Bit Rate Register for details. Watch SBRHP, SBRLP as EBRHP[5:0], EBRLP[5:0].

EBRHP[5:0] bits (Extended Bit Rate High-Level Period Push-Pull)

See SBRHP[5:0] bits of section 33.2.15. STDBR : Standard Bit Rate Register for details. Watch SBRHP, SBRLP as EBRHP[5:0], EBRLP[5:0].

Table 33.5 Requirement and usage of setting in each mode (1 of 2)

Bit name	Device mode				
	I2C master	I2C slave	I3C Master	I3C Secondary Master	I3C Slave
EBRHP[5:0]	do not use	do not use	Setting required*3	Setting required*4	do not use
EBRLP[5:0]	do not use	do not use	Setting required*3	Setting required*4	do not use

Table 33.5 Requirement and usage of setting in each mode (2 of 2)

Bit name	Device mode				
	I²C master	I²C slave	I3C Master	I3C Secondary Master	I3C Slave
EBRHO[7:0]	Setting required*1	do not use	Setting required*3	Setting required*4	do not use
EBRLO[7:0]	Setting required*1	Setting required*2	Setting required*3	Setting required*4	do not use

Note 1. The setting value is used for the data rate of High-Speed mode.
Note 2. The setting value is used for the data setup time of automatic SCL low-hold operation in Hs-mode.
Note 3. The setting value is used for the data rate of each communication.
Note 4. When operating with I3C Master, the setting value is used for the data rate of each communication.
When operating with I3C Slave, do not use.

33.2.17 BFRECDT : Bus Free Condition Detection Time Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x07C

Bit position:	31																					9	8									0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRECYC[8:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	FRECYC[8:0]	Bus Free Condition Detection Cycle The count value is a period for detecting the Bus free condition.	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

FRECYC[8:0] bits (Bus Free Condition Detection Cycle)

I3C counts the period for detecting the Bus free condition with the I3Cφ.

These bits set the Bus Free period. This Bus Free period is counted by the internal reference clock (I3Cφ) selected by the REFCKCTL.IREFCKS[2:0] bits. See the BCST.BFREF flag for Bus Free detection behavior.

33.2.18 BAVLCDT : Bus Available Condition Detection Time Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x080

Bit position:	31																					9	8									0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AVLCYC[8:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	AVLCYC[8:0]	Bus Available Condition Detection Cycle The count value is a period for detecting the Bus available condition.	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

AVLCYC[8:0] bits (Bus Available Condition Detection Cycle)

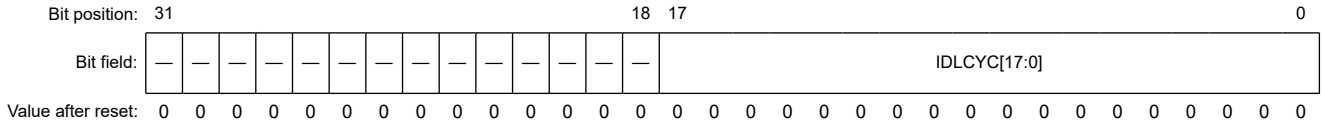
I3C counts the period for detecting the Bus available condition with the I3Cφ.

These bits set the Bus Available period. This Bus Available period is counted by the internal reference clock (I3Cφ) selected by the REFCKCTL.IREFCKS[2:0] bits. See the BCST.BAVLF flag for Bus Available detection behavior.

33.2.19 BIDLCDT : Bus Idle Condition Detection Time Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x084



Bit	Symbol	Function	R/W
17:0	IDLCYC[17:0]	Bus Idle Condition Detection Cycle The count value is a period for detecting the Bus idle condition.	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

IDLCYC[17:0] bits (Bus Idle Condition Detection Cycle)

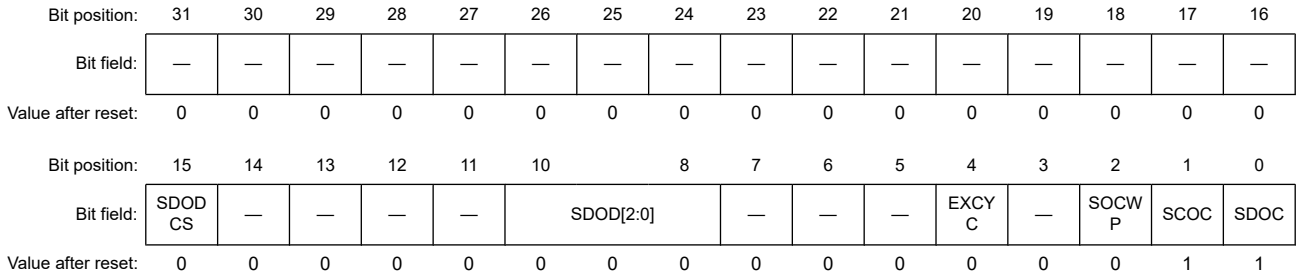
I3C counts the period for detecting the Bus idle condition with the I3Cφ.

These bits set the Bus Idle period. This Bus Idle period is counted by the internal reference clock (I3Cφ) selected by the REFCKCTL.IREFCKS[2:0] bits. See the BCST.BIDLF flag for Bus Available detection behavior.

33.2.20 OUTCTL : Output Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x088



Bit	Symbol	Function	R/W
0	SDOC	SDA Output Control*1 0: I3C drives the I3C_SDA pin low. 1: I3C releases the I3C_SDA pin.	R/W
1	SCOC	SCL Output Control*1 High level output is achieved through an external pull-up resistor. 0: I3C drives the I3C_SCL pin low. 1: I3C releases the I3C_SCL pin.	R/W
2	SOCWP	SCL/SDA Output Control Write Protect*1 0: Bits SCOC and SDOC are protected. 1: Bits SCOC and SDOC can be written (When writing simultaneously with the value of the target bit). This bit is read as 0.	W
3	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
4	EXCYC	Extra SCL Clock Cycle Output*3 The EXCYC bit is cleared automatically after one clock cycle is output. 0: Does not output an extra SCL clock cycle (default). 1: Outputs an extra SCL clock cycle.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
10:8	SDOD[2:0]	SDA Output Delay*2 0 0 0: No output delay 0 0 1: 1 I3Cφ cycle (When OUTCTL.SDODCS = 0 (I3Cφ)) 1 or 2 I3Cφ cycles (When OUTCTL.SDODCS = 1 (I3Cφ/2)) 0 1 0: 2 I3Cφ cycles (When OUTCTL.SDODCS = 0 (I3Cφ)) 3 or 4 I3Cφ cycles (When OUTCTL.SDODCS = 1 (I3Cφ/2)) 0 1 1: 3 I3Cφ cycles (When OUTCTL.SDODCS = 0 (I3Cφ)) 5 or 6 I3Cφ cycles (When OUTCTL.SDODCS = 1 (I3Cφ/2)) 1 0 0: 4 I3Cφ cycles (When OUTCTL.SDODCS = 0 (I3Cφ)) 7 or 8 I3Cφ cycles (When OUTCTL.SDODCS = 1 (I3Cφ/2)) 1 0 1: 5 I3Cφ cycles (When OUTCTL.SDODCS = 0 (I3Cφ)) 9 or 10 I3Cφ cycles (When OUTCTL.SDODCS = 1 (I3Cφ/2)) 1 1 0: 6 I3Cφ cycles (When OUTCTL.SDODCS = 0 (I3Cφ)) 11 or 12 I3Cφ cycles (When OUTCTL.SDODCS = 1 (I3Cφ/2)) 1 1 1: 7 I3Cφ cycles (When OUTCTL.SDODCS = 0 (I3Cφ)) 13 or 14 I3Cφ cycles (When OUTCTL.SDODCS = 1 (I3Cφ/2))	R/W
14:11	—	These bits are read as 0. The write value should be 0.	R/W
15	SDODCS	SDA Output Delay Clock Source Selection*3 0: The internal reference clock (I3Cφ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (I3Cφ/2) is selected as the clock source of the SDA output delay counter.*4	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports I^2C, I3C master, and I3C secondary master mode.
Note 2. These bits support I^2C mode.
Note 3. This bit supports I^2C mode.
Note 4. The setting SDODCS = 1 (I3Cφ/2) only becomes valid when SCL is at the low level. When SCL is at the high level, the setting SDODCS = 1 becomes invalid and the clock source becomes the internal reference clock (I3Cφ).

SDOC bit (SDA Output Control) and SCOC bit (SCL Output Control)

These bits are used to directly control the I3C_SDA and I3C_SCL signals output from this I3C.

When writing to these bits, also write 1 to the SOCWP bit at the same time.

The result of setting these bits is input to I3C through the input buffer. When slave mode is selected, a START condition may be detected and the bus may be released depending on the bit settings.

Do not rewrite these bits during a START condition, STOP condition, Repeated START condition, or during transmission or reception. Operation after rewriting under the above conditions is not guaranteed.

EXCYC bit (Extra SCL Clock Cycle Output)

This bit is used to output an extra SCL clock cycle for debugging or error processing.

Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.

For details on this function, see section 33.3.2.3.11. Port Control , (1) Extra SCL Clock Cycle Output Function.

33.2.21 INCTL : Input Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x08C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	DNFE	DNFS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0

Bit	Symbol	Function	R/W
3:0	DNFS[3:0]	Digital Noise Filter Stage Selection 0x0: Noise of up to one I3Cφ cycle is filtered out (single-stage filter). 0x1: Noise of up to two I3Cφ cycles is filtered out (2-stage filter). 0x2: Noise of up to three I3Cφ cycles is filtered out (3-stage filter). 0x3: Noise of up to four I3Cφ cycles is filtered out (4-stage filter). 0x4: Noise of up to five I3Cφ cycles is filtered out (5-stage filter). ⋮ 0xF: Noise of up to sixteen I3Cφ cycles is filtered out (16-stage filter).	R/W
4	DNFE	Digital Noise Filter Circuit Enable 0: No digital noise filter circuit is used. 1: A digital noise filter circuit is used.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
7:6	—	These bits are read as 1. The write value should be 1.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I^2C mode.

DNFS[3:0] bits (Digital Noise Filter Stage Selection)

These bits are used to select the number of stages in the digital noise filter.

For details on the digital noise filter function, see section 33.3.2.6.3. Digital Noise-Filter Circuits [I^2C mode].

In I^2C High Speed mode, I3C changes the number of noise filter stage to a quarter of the number of noise filter stage automatically.

Note:
- Set the noise range to be filtered out by the noise filter within a range less than the I3C_SCL line high-level period or low-level period. If the noise range is set to a value of (SCL clock width: high-level period or lowlevel period, whichever is shorter) - [1.5 internal reference clock (I3Cφ) cycles] or more, the SCL clock is regarded as noise by the noise filter function of I3C, which may prevent I3C from operating normally.
- In I^2C High Speed mode, the lower 2 bits of the DNFS [3:0] bits are ignored, and the number of filter stages for 1 to 4 stages is selected by the upper 2 bits.

33.2.22 TMOCTL : Timeout Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x090

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	TOMDS[1:0]		TOHCTL	TOLCTL	—	—	TODTS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TODTS[1:0]	Timeout Detection Time Selection 0 0: 16bit-timeout 0 1: 14bit-timeout 1 0: 8bit-timeout 1 1: 6bit-timeout	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	TOLCTL	Timeout L Count Control 0: Count is disabled while the I3C_SCL line is at a low level. 1: Count is enabled while the I3C_SCL line is at a low level.	R/W
5	TOHCTL	Timeout H Count Control 0: Count is disabled while the I3C_SCL line is at a high level. 1: Count is enabled while the I3C_SCL line is at a high level.	R/W
7:6	TOMDS[1:0]	Timeout Operation Mode Selection 0 0: Timeout is detected during the following conditions: • The bus is busy (BCST.BFREF = 0) in master mode. • I3C's own slave address is detected and the bus is busy in slave mode. • The bus is free (BCST.BFREF = 1) while generation of a START condition is requested (CNDCTL.STCND = 1). 0 1: Timeout is detected while the bus is busy. 1 0: Timeout is detected while the bus is free. 1 1: Setting prohibited	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

TODTS[1:0] bits (Timeout Detection Time Selection)

These bits are used to select for the timeout detection time when the timeout function is enabled (BSTE.TODE bit = 1).

When these bits are set to 00b, the timeout detection internal counter functions as a 16-bit counter.

When these bits are set to 01b, the counter functions as a 14-bit counter.

When these bits are set to 10b, the counter functions as a 8-bit counter.

When these bits are set to 11b, the counter functions as a 6-bit counter.

While the I3C_SCL line is in the state that enables this counter as specified by bits TOHCTL and TOLCTL, the counter counts up in synchronization with the internal reference clock (I3Cφ) as a count source.

For details on the timeout function, see section 33.3.2.4.5. Timeout Error Detection .

TOLCTL bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the I3C_SCL line is held low when the timeout function is enabled (BSTE.TODE = 1).

TOHCTL bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the I3C_SCL line is held high when the timeout function is enabled (BSTE.TODE = 1).

TOMDS[1:0] bits (Timeout Operation Mode Selection)

These bits are used to select the detection condition for timeout when the timeout function is enabled.

Note: When working with I²C Slave, during 10-bit address communication, the timeout count starts when the upper address match is detected.

33.2.23 WUCTL : Wake Up Unit Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x098

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	WUFE	WUFSYNE	—	WUANFS	—	—	—	WUACKS
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	WUACKS	Wake-Up Acknowledge Selection*1 Choice of four response mode with a combination of RSTCTL.INTLRST bit and WUACKS bit. Shown in Table 33.6.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	WUANFS	Wake-Up Analog Noise Filter Selection*1 0: Do not add the Wake Up analog filter. 1: Add the Wake Up analog filter.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	WUFSYNE	Wake-Up function PCLK Synchronous Enable 0: I3C asynchronous circuit enable 1: I3C synchronous circuit enable	R/W
7	WUFE	Wake-Up function Enable Do not set WUFE = 0 during Wake-Up operation. 0: Wake-up function disables 1: Wake-up function enables	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports I²C mode.

Table 33.6 Wake-Up Mode

INTLRST	WUACKS	Operation mode	Description
0	0	Normal Wake-Up mode 1	ACK response at 9th SCL and SCL low hold after at 9th SCL.
0	1	Normal Wake-Up mode 2	No ACK response immediately and SCL low hold between 8th and 9th SCL. Release SCL low hold and ACK response at 9th SCL.
1	0	Command recovery mode	ACK response at 9th SCL and not SCL low hold.
1	1	EEP response mode	NACK response at 9th SCL and not SCL low hold.

Note: In WakeUp mode 2, HS mode cannot be used.

WUFSYNE bit (Wake-Up function PCLK Synchronous Enable)

This bit is used to switch between the PCLK synchronous operation and the PCLK asynchronous operation.

The bit is used in combination with the WUASYNF flag at Wake-Up effective function (WUCTL.WUFE bit = 1).

[When switching from the PCLK synchronous operation to the PCLK asynchronous operation]

I3C operation changes into the PCLK asynchronous operation during BCST.BFREF flag = 1, when the WUASYNF flag set to 1 during WUFSYNE = 0.

The reception can operate without depending on the state of operation of PCLK (With PCLK stopped) after it switches to the PCLK asynchronous operation (Wake-Up event detection operation).

[When switching from the PCLK asynchronous operation to the PCLK synchronous operation]

I3C operation changes into the PCLK synchronous operation at the following conditions. (At the same timing when WUFSYNE flag becomes 0)

In the case Wake-Up event detects : right after WUFSYNE bit is set to 1.

In the case Wake-Up event does not detect : when STOP condition is detected after WUFSYNE bit is set to 1.

[Setting condition]

- When 1 is written to the WUFSYNE bit.
- WUCTL.WUFE = 0

[Clearing conditions]

- When 0 is written to the WUFSYNE bit.

33.2.24 ACKCTL : Acknowledge Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x0A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	ACKTWP	ACKT	ACKR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ACKR	Acknowledge Reception 0: A 0 is received as the acknowledge bit (ACK reception). 1: A 1 is received as the acknowledge bit (NACK reception).	R
1	ACKT	Acknowledge Transmission 0: A 0 is sent as the acknowledge bit (ACK transmission). 1: A 1 is sent as the acknowledge bit (NACK transmission).	R/W
2	ACKTWP	ACKT Write Protect 0: The ACKT bit are protected. 1: The ACKT bit can be written (when writing simultaneously with the value of the target bit). This bit is read as 0.	W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I^2C mode.

ACKR bit (Acknowledge Reception)

This bit is used to store the acknowledge bit information received from the receive device in transmit mode.

[Setting condition]

- When 1 is received as the acknowledge bit with the PRSST.TRMD bit set to 1.

[Clearing condition]

- When 0 is received as the acknowledge bit with the PRSST.TRMD bit set to 1.

ACKT bit (Acknowledge Transmission)

[Setting condition]

- When 1 is written to the ACKT bit and 1 is written to the ACKTWP bit at the same time.

[Clearing conditions]

- When 0 is written to the ACKT bit and 1 is written to the ACKTWP bit at the same time.
- When a STOP condition is detected. (when a STOP condition is detected with the CNDCTL.SPCND bit set to 1.)

Note: Set the ACKT bit to 0 in I^2C Slave mode.

ACKTWP bit (ACKT Write Protect)

This bit is used to control the modification of the ACKT bit.

When changing the ACKT bit, setting this bit to 1 at the same time can change the ACKT bit.

When this bit is read, 0 is always read.

33.2.25 SCSTRCTL : SCL Stretch Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x0A4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RWE	ACKTWE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ACKTWE	Acknowledge Transmission Wait Enable 0: NTST.RDBFF0 is set at the rising edge of the ninth SCL clock cycle. (The I3C_SCL line is not held low at the falling edge of the eighth clock cycle.) 1: NTST.RDBFF0 is set at the rising edge of the eighth SCL clock cycle. (The I3C_SCL line is held low at the falling edge of the eighth clock cycle.) Low-hold is released by writing a value to the ACKCTL.ACKT bit.	R/W
1	RWE	Receive Wait Enable 0: No WAIT (The period between ninth clock cycle and first clock cycle is not held low.) 1: WAIT (The period between ninth clock cycle and first clock cycle is held low.) Low-hold is released by reading NTDTBP0.	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I^2C mode.

ACKTWE bit (Acknowledge Transmission Wait Enable)

This bit is used to select the NTST.RDBFF0 flag set timing in receive mode and also to select whether to hold the I3C_SCL line low at the falling edge of the eighth SCL clock cycle.

When ACKTWE = 0, the I3C_SCL line is not held low at the falling edge of the eighth SCL clock cycle, and the NTST.RDBFF0 flag is set to 1 at the rising edge of the ninth SCL clock cycle.

When ACKTWE = 1, the NTST.RDBFF0 flag is set to 1 at the rising edge of the eighth SCL clock cycle and the I3C_SCL line is held low at the falling edge of the eighth SCL clock cycle. The low-hold of the I3C_SCL line is released by writing a value to the ACKCTL.ACKT bit.

After data is received with this setting, the I3C_SCL line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKCTL.ACKT = 0) or NACK (ACKCTL.ACKT = 1) according to receive data.

RWE bit (Receive Wait Enable)

This bit is used to control whether to hold the period between the ninth SCL clock cycle and the first SCL clock cycle low until the receive data buffer (NTDTBP0) is completely read each time single-byte data is received in receive mode.

When RWE = 0, the receive operation is continued without holding the period between the ninth and the first SCL clock cycle low. When both the ACKTWE and RWE bits = 0, continuous receive operation is enabled with the double buffer.

When RWE = 1, the I3C_SCL line is held low from the falling edge of the ninth clock cycle until the NTDTBP0 value is read each time single-byte data is received.

This enables receive operation in byte units.

Note: When the value of the RWE bit is to be read, be sure to read the NTDTBP0 beforehand.

33.2.26 SCSTLCTL : SCL Stalling Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x0B0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ACKPE	PARPE	—	AAPE	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	0
Bit field:	STLCYC[15:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
15:0	STLCYC[15:0]	Stalling Cycle Counter setting of stall period (I3Cφ cycle). Common use for each phase.	R/W
27:16	—	These bits are read as 0. The write value should be 0.	R/W
28	AAPE	Assigned Address Phase Enable Enable bit that allows stall by the first bit at address assignment 0: Does not stall the SCL clock during the address assignment phase. 1: Stall the SCL clock during address assignment phase.	R/W
29	—	This bit is read as 0. The write value should be 0.	R/W
30	PARPE	Parity Phase Enable Stall enable bit in parity bit period 0: Does not stall the SCL clock during the parity bit period. 1: Stall the SCL clock during the parity bit period.	R/W
31	ACKPE	ACK phase Enable Stall enable bit during ACK/NACK phase 0: Does not stall the SCL clock during the ACK/NACK phase. 1: Stall the SCL clock during the ACK/NACK phase.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

When setting this register, follow Chapter 5.1.2.5 Master Clock Stalling of MIPI I3C Spec V1.0, and use it only when necessary because of its negative impacts on bus performance.

STLCYC[15:0] bits (Stalling Cycle)

These bits set the SCL stall period. The SCL stall period is counted by the internal reference clock (I3Cφ). This is a counter common to the enable bits of each phase.

AAPE bit (Assigned Address Phase Enable)

The master can stall SCL during the low period of the first bit of the assigned address phase of the Enter Dynamic Address Assignment CCC command. It can gain time in assigning dynamic address to the device based on the BCR and DCR of the slave. However, because the Dynamic Address Assignment procedure sends the dynamic address set in the DATBASm (m = 0 to 7) register in sequence, it is not necessary to set this bit and it is prohibited.

PARPE bit (Parity Phase Enable)

The parity bit of the transmission data of I3C write transfer can be used for SCL stalling to avoid underrun of the transmission data FIFO. However, when the transmission data FIFO of the I3C master becomes empty, SCL stalling is performed regardless of the setting of this bit, it is not necessary to set this bit and it is prohibited. It is necessary to set this bit when the I3C slave requires preparation time to receive data.

ACKPE bit (ACK phase Enable)

Determine the need to perform SCL stalling in the ACK/NACK phase based on the following criteria:

- It is necessary to set this bit when the I3C and I2C slaves connected to the bus require preparation time to receive or transmit data.
- In legacy I^2C communication, if there is a possibility that the data FIFO of the I3C master might underrun or overflow, it is not necessary to set this bit because SCL Stalling is performed by FIFO Empty or Full regardless of the setting of this bit.
- Other than legacy I^2C communication, the data FIFO of I3C master might underrun or overflow, and if SCL stalling is required in ACK phase, this bit can be set. However, it is necessary to build the software so that the FIFO does not underrun or overflow due to the interrupt generated according to the FIFO threshold setting (NQTHCTL, NTBTHCTL0, NRQTHCTL, HQTHCTL, HTBTHCTL).
- When I3C master responds ACK/NACK to IBI, it is not necessary to set this bit because ACK/NACK response can be set in advance by BCTL.HJACK, DATBASm.DVMRRJ and DATBASm.DVS IRRJ (m = 0 to 7).
- It is necessary to set this bit when the I3C slave connected to the bus requires preparation time to transmit data for Direct GET CCC.

33.2.27 SVTDLG0 : Slave Transfer Data Length Register 0

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x0C0

Bit position:	31															16	15													0
Bit field:	STDLG[15:0]																—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
31:16	STDLG[15:0]	Slave Transfer Data Length Indicates the number of bytes to be transferred.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

33.2.28 STCTL : Synchronous Timing Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x120

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STOE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STOE	Synchronous Timing output Enable 0: Disable 1: Enable	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

33.2.29 ATCTL : Asynchronous Timing Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x124

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CDIV[7:0]								—	—	—	—	—	AMEOE	MREFOE	ATTRGS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ATTRGS	Asynchronous Timing Trigger Select*1 0: Software trigger 1: Hardware trigger	R/W
1	MREFOE	MREF Output Enable (Capture Event / Counter Overflow)*2 0: Disable 1: Enable	R/W
2	AMEOE	Additional Master-initiated bus Event Output Enable*2 0: Disable 1: Enable	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
15:8	CDIV[7:0]	TCLK Counter Divide Setting*3	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports I3C secondary master mode and I3C slave mode.
Note 2. This bit supports I3C master mode and I3C secondary master mode.
Note 3. These bits support all I3C mode.

33.2.30 ATTRG : Asynchronous Timing Trigger Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x128

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ATSTRG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ATSTRG	Asynchronous Timing Software Trigger 0: Do nothing 1: Software trigger (one-shot pulse) output This bit is always read as 0.	W
31:1	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

33.2.31 ATCCNTE : Asynchronous Timing Control Counter Enable Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x12C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ATCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ATCE	Asynchronous Timing Counter Enable for MREF, MC2, SC1, SC2. 0: Disable 1: Enable	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

33.2.32 CNDCTL : Condition Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x140

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SPCND	SRCND	STCND
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STCND	START (S) Condition Issuance 0: Does not request to issue a START condition. 1: Requests to issue a START condition.	R/W
1	SRCND	Repeated START (Sr) Condition Issuance 0: Does not request to issue a Repeated START condition. 1: Requests to issue a Repeated START condition.	R/W
2	SPCND	STOP (P) Condition Issuance 0: Does not request to issue a STOP condition. 1: Requests to issue a STOP condition.	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I^2C mode.

STCND bit (START (S) Condition Issuance)

This bit is used to request transition to master mode and issuance of a START condition.

For details on the START condition issuance, see section 33.3.2.3.3. START Condition / Repeated START Condition / STOP Condition Issuing Function .

[Setting condition]

- When 1 is written to the STCND bit

[Clearing conditions]

- When 0 is written to the STCND bit
- When a START condition has been issued (A START condition is detected)
- When the BST.ALF (arbitration-lost) flag is set to 1

Note: Set the STCND bit to 1 (START condition issuance request) when the BCST.BFREF flag is set to 1 (bus free state).
Note that arbitration may be lost due to a START condition issuance error if the STCND bit is set to 1 (START condition issuance request) when the BFREF flag is set to 0 (bus busy state).

SRCND bit (Repeated START (Sr) Condition Issuance)

This bit is used to request that a Repeated START condition be issued in master mode.

When this bit is set to 1 to request to issue a Repeated START condition, a Repeated START condition is issued when the BFREF flag is set to 0 (bus busy state) and the PRSST.CRMS bit is set to 1 (master mode).

For details on the Repeated START condition issuance, see section 33.3.2.3.3. START Condition / Repeated START Condition / STOP Condition Issuing Function .

[Setting condition]

- When 1 is written to the SRCND bit with the BCST.BFREF flag set to 0

[Clearing conditions]

- When 0 is written to the SRCND bit
- When a Repeated START condition has been issued (A Repeated START condition is detected)
- When the BST.ALF (arbitration-lost) flag is set to 1

Note: Do not set the SRCND bit to 1 while issuing a STOP condition.

Note: If 1 (requests to issue a Repeated START condition) is written to the SRCND bit in slave mode, the Repeated START condition is not issued but the SRCND bit remains set to 1.

If the operating mode changes to master mode with the bit not being cleared, note that the Repeated START condition may be issued.

SPCND bit (STOP (P) Condition Issuance)

This bit is used to request that a STOP condition be issued in master mode.

When this bit is set to 1 to request to issue a STOP condition, a STOP condition is issued when the BCST.BFREF flag is set to 0 (bus busy state) and the PRSST.CRMS bit is set to 1 (master mode).

For details on the STOP condition issuance, see section 33.3.2.3.3. START Condition / Repeated START Condition / STOP Condition Issuing Function .

[Setting condition]

- When 1 is written to the SPCND bit with the BCST.BFREF flag set to 0 and the PRSST.CRMS bit set to 1

[Clearing conditions]

- When 0 is written to the SPCND bit
- When a STOP condition has been issued (A STOP condition is detected)
- When the BST.ALF (arbitration-lost) flag is set to 1
- When a START condition and a Repeated START condition are detected

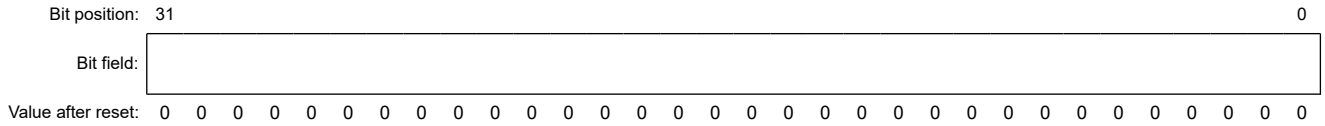
Note: Writing to the SPCND bit is not possible while the setting of the BCST.BFREF flag = 1 (bus free state).

Note: Do not set the SPCND bit to 1 while a Repeated START condition is being issued.

33.2.33 NCMDQP : Normal Command Queue Port Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x150



Bit	Symbol	Function	R/W
31:0	n/a	Normal Command Queue Port	W

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

32-bit mailbox register NCMDQP contains a command descriptor structure that depends on the requested transfer type:

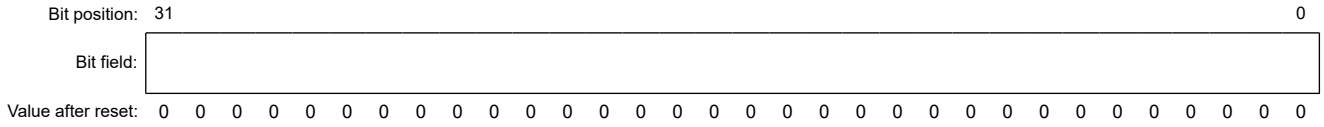
1. Address Assignment Command (see section 33.3.1.1.1. Address Assign Command)
2. Immediate Data Transfer (see section 33.3.1.1.2. Immediate Transfer Command)
3. Regular Data Transfer (see section 33.3.1.1.3. Regular Transfer Command)
4. Write + Write/Read Combo Transfer (see section 33.3.1.1.4. Combo Transfer Command)
5. Internal Control Command (see section 33.3.1.1.5. Internal Control Command)

Within the command descriptor, DWORDs appear starting with the Least Significant DWORD, in order until the Most Significant DWORD.

33.2.34 NRSPQP : Normal Response Queue Port Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x154



Bit	Symbol	Function	R/W
31:0	n/a	Normal Response Queue Port	R

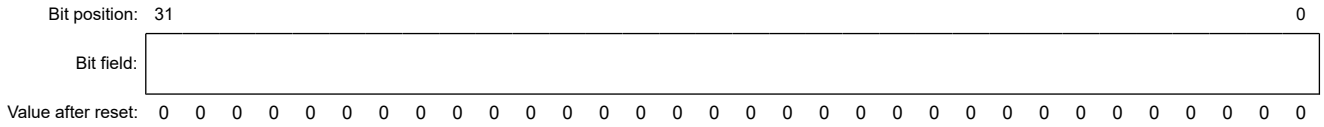
Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

32-bit mailbox register NRSPQP contains a response structure (see section 33.3.1.4. Receive Status Descriptor).

33.2.35 NTDTBP0/NTDTBP0_BY : Normal Transfer Data Buffer Port Register 0

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x158



Bit	Symbol	Function	R/W
31:0	n/a	Normal Transfer Data Buffer Port NTDTBP0 is a 32-bit read/write register. NTDTBP0_BY (NTDTBP0[7:0]) is a 8-bit read/write register.	R/W

Note: S-TYPE3, P-TYPE3

Note: NTDTBP0 is 32-bit access in I3C mode.
NTDTBP0_BY is 8-bit access in I^2C mode.

32-bit mailbox register NTDTBP0 is a 32-bit bi-directional data transfer register which is used both to read from the Normal Rx Data Buffer, and to write to the Normal Tx Data Buffer.

In other words, the Normal Rx Data Buffer and the Normal Tx Data Buffer have the same offset, forming a single bidirectional port for transmitting or receiving I3C data.

Read Operations:

[I3C protocol mode]

Data Read from the Normal Rx Data Buffer. Its should be read based on Normal Queue Status Level indications. The Rx data is always aligned to a 4-byte boundary, and stored in the Normal Rx Data Buffer. If the length of the data transfer is not aligned to a 4-byte boundary, then there will be extra (unused) bytes at the end of the transferred data. The valid data must be identified using the DATA_LENGTH field in the Response Descriptor.

[I2C protocol mode]

When 1 byte of data has been received, the received data is transferred from the internal shift register to NTDTBP0 to enable the next data to be received. The double-buffer structure of the internal shift register and NTDTBP0 allows continuous receive operation if the received data has been read from NTDTBP0 while the internal shift register is receiving

data. Read data from NTDTBP0 once when a Normal Rx Data buffer full interrupt (I3C_RX) request is generated. If NTDTBP0 receives the next Rx data before the current data is read from NTDTBP0 (while the RDBFF0 flag in NTST is 1), this module automatically holds the SCL clock low one cycle before the RDBFF0 flag is set to 1 next. The lower 8 bits of the read 32-bit data are valid as received data.

Write Operations:

[I3C protocol mode]

Data Written to the Normal Tx Data Buffer. Data DWORDs written to the Normal Tx Data Buffer are placed onto the I3C bus one byte at a time, with the DWORD LSB first. Within each byte, bits are placed onto the I3C bus in big-endian order, with bit 7 going out first on the bus. The Tx data should always start aligned to a 4-byte boundary, and written to the NTDTBP0 register. If the length of the transfer is not aligned to a 4-byte boundary, then there will be extra (unused) bytes at the end of the transferred data. I3C shall only send the valid number of bytes indicated in the DATA_LENGTH field of the Command Descriptor.

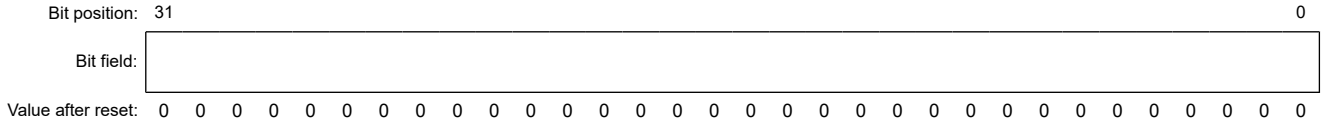
[I2C protocol mode]

When NTDTBP0 detects a space in the internal shift register, it transfers the Tx data that has been written to NTDTBP0 to the internal shift register and starts transmitting data in transmit mode. The double-buffer structure of NTDTBP0 and the internal shift register allows continuous transmit operation if the next Tx data has been written to NTDTBP0 while the internal shift register data is being transmitted. Write Tx data to NTDTBP0 once when a Normal Tx Data buffer empty interrupt (I3C_TX) request is generated. The lower 8 bits of the written 32-bit data are valid as transmission data.

33.2.36 NIBIQP : Normal IBI Queue Port Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x17C



Bit	Symbol	Function	R/W
31:0	n/a	Normal IBI Queue Port	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

When receiving an IBI, 32-bit mailbox register NIBIQP is used for both:

- Read the IBI status descriptor (see section 33.3.1.3. IBI Status Descriptor)
- Read the IBI data (which is raw/opaque data).

The IBI status descriptor is a read-only structure describing an IBI event received from a Slave device on the I3C bus.

Note: If the I3C HCI auto-read feature is used, then the IBI data includes the data received from the auto-generated private read operation.

Even if LAST_STATUS is set to 0, the driver software still evaluates the data payload length by examining the CHUNKS field.

33.2.37 NRSQP : Normal Receive Status Queue Port Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x180

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Normal Receive Status Queue Port	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

32-bit mailbox register NRSQP contains a receive status structure (see section 33.3.1.4. Receive Status Descriptor).

33.2.38 HCMDQP : High Priority Command Queue Port Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x184

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	High Priority Command Queue Port	W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

32-bit mailbox register HCMDQP contains a command descriptor structure that depends on the requested transfer type:

1. Address Assignment Command (see section 33.3.1.1.1. Address Assign Command)
2. Immediate Data Transfer (see section 33.3.1.1.2. Immediate Transfer Command)
3. Regular Data Transfer (see section 33.3.1.1.3. Regular Transfer Command)
4. Write + Write/Read Combo Transfer (see section 33.3.1.1.4. Combo Transfer Command)
5. Internal Control Command (see section 33.3.1.1.5. Internal Control Command)

Within the command descriptor, DWORDs appear starting with the least significant DWORD, in order until the most significant DWORD.

33.2.39 HRSPQP : High Priority Response Queue Port Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x188

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	High Priority Response Queue Port	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

32-bit mailbox register HRSPQP contains a response structure. (see section 33.3.1.2. Response Descriptor)

33.2.40 HTDTBP : High Priority Transfer Data Buffer Port Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x18C

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	High Priority Transfer Data Buffer Port	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

The HTDTBP register is a 32-bit bi-directional data transfer register which is used both to read from the high priority Rx data, and to write to the high priority Tx data.

For Read Operation:

To receive data from the High Priority Rx Data Buffer, read from the HTDTBP register. It should be read based on queue status indication.

The Rx data is always aligned to a 4-byte boundary, and stored in the High Priority Rx Data Buffer.

If the length of the data transfer is not aligned to a 4-byte boundary, then there will be extra (unused) bytes at the end of the transferred data.

The valid data must be identified using the DATA_LENGTH field in the response descriptor.

For Write Operation:

To send data to the High Priority Tx Data Buffer, write to the HTDTBP register. Data DWORDs written to the Data port are placed onto the I3C bus one byte at a time, with the DWORD's LSB first. Within each byte, bits are placed onto the I3C bus in big-endian order, with bit 7 going out first on the bus.

The High Priority Tx Data Buffer Port is mapped to the High Priority Tx Data Buffer.

The Tx data should always start aligned to a 4byte boundary, and written to the Tx data port register.

If the length of the transfer is not aligned to a 4-byte boundary, then there will be extra (unused) bytes at the end of the transferred data.

I3C shall only send the valid number of bytes indicated in the DATA_LENGTH field of the command descriptor.

33.2.41 NQTHCTL : Normal Queue Threshold Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x190

Bit position:	31							24	23							16	15							8	7							0
Bit field:	IBIQTH[7:0]								IBIDSSZ[7:0]								RSPQTH[7:0]								CMDQTH[7:0]							
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
7:0	CMDQTH[7:0]	Normal Command Queue Threshold*1 0x00: Interrupt is issued when Normal Command Queue is completely empty. Others: Interrupt is issued when Normal Command Queue contains N empties. (N = CMDQTH[7:0])	R/W
15:8	RSPQTH[7:0]	Normal Response Queue Threshold*1 0x00: Interrupt is issued when Normal Response Queue contains 1 entry (DWORD). Others: Interrupt is triggered when Normal Response Queue contains N+1 entries (DWORD). (N = CMDQTH[7:0])	R/W
23:16	IBIDSSZ[7:0]	Normal IBI Data Segment Size*2 Supported Values: Minimum: 1 (4 bytes) Maximum: 63 (252 bytes), provided that the configured IBI Queue depth is 64 or more. When ATCCNTE.ATCE = 1, restrict to the number of slices ≥ 2.	R/W
31:24	IBIQTH[7:0]	Normal IBI Queue Threshold*1 0x00: I3C Protocol mode (Master): Interrupt is generated when the Outstanding IBI Status count is 1 or more. I3C Protocol mode (Slave): Interrupt is issued when IBI Data Buffer is completely empty. Others: I3C Protocol mode (Master): Interrupt is generated when the Outstanding IBI Status count is N + 1 or more. (N = CMDQTH[7:0]) I3C Protocol mode (Slave): Interrupt is issued when IBI Data Buffer contains N empties.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. These bits support all I3C mode.
Note 2. These bits support I3C master mode and I3C secondary master mode.

The Queue Threshold Control register controls the interrupt trigger thresholds for the Command Queue, the Response Queue, and the IBI Queue.

The specific reset values are indicative, and could be hardware implementation specific.

CMDQTH[7:0] bits (Normal Command Queue Threshold)

Controls the minimum number of Normal Command Queue empties needed to trigger the I3C_CMD interrupt.

If this field is greater than (Normal Command Queue size*1 – 1), then only the number of bits required to address the full buffer depth will be considered.

RSPQTH[7:0] bits (Normal Response Queue Threshold)

Controls the minimum number of Normal Response Queue entries needed to trigger the I3C_RESP interrupt.

If this field is greater than (Normal Response Queue size*2 – 1), then only the number of bits required to address the full buffer depth will be considered.

IBIDSSZ[7:0] bits (Normal IBI Data Segment Size)

This is the IBI data segment size, in DWORDs (4 bytes).

In PIO mode, this field allows the incoming Normal IBI data to be sliced into multiple segments generating status individually, to support cutthrough readout of a long IBI payload data.

When Asynchronous Timing Control mode is supported, this field should be set to a value other than 1 or 3 to allow the single data segment to contain the entire Master time-stamp value (for example, both MREF and MC2).

IBIQTH[7:0] bits (Normal IBI Queue Threshold)

For I3C protocol mode (Master): PRTS.PRTMD = 0 and PRSST.CRMS = 1.

Controls generation of the I3C_IBI interrupt, based on the value of the Normal IBI Queue's Outstanding IBI status count.

Each IBI status entry can represent either the complete IBI payload (if the IBI payload byte size is 4 × IBIDSSZ or less), or a segment of the IBI payload (if the IBI payload byte size is more than 4 × IBIDSSZ).

For I3C protocol mode (Slave) : PRTS.PRTMD bit = 0, PRSST.CRMS bit = 0.

Controls the minimum number of IBI Data Buffer empties needed to trigger the I3C_IBI interrupt.

If this field is greater than (IBI Data Buffer size[*3] – 1), then only the number of bits required to address the full buffer depth will be considered.

Note 1. Normal Command Queue size is 4.
Note 2. Normal Response Queue size is 4.
Note 3. IBI Data Buffer size is 8.

Note: It is assumed that I3C has exactly one Command Queue, exactly one Response Queue, and exactly one IBI Queue.

33.2.42 NTBTHCTL0 : Normal Transfer Data Buffer Threshold Control Register 0

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x194

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	RXSTTH[2:0]			—	—	—	—	—	TXSTTH[2:0]		
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	RXDBTH[2:0]			—	—	—	—	—	TXDBTH[2:0]		
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
2:0	TXDBTH[2:0]	Normal Tx Data Buffer Threshold[*1] 0 0 0: Interrupt triggers at 2 Tx Buffer empties, DWORDs 0 0 1: Interrupt triggers at 4 Tx Buffer empties, DWORDs 0 1 0: Interrupt triggers at 8 Tx Buffer empties, DWORDs 0 1 1: Interrupt triggers at 16 Tx Buffer empties, DWORDs Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
10:8	RXDBTH[2:0]	Normal Rx Data Buffer Threshold[*1] 0 0 0: Interrupt triggers at 2 Rx Buffer entries, DWORDs 0 0 1: Interrupt triggers at 4 Rx Buffer entries, DWORDs 0 1 0: Interrupt triggers at 8 Rx Buffer entries, DWORDs 0 1 1: Interrupt triggers at 16 Rx Buffer entries, DWORDs Others: Setting prohibited	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
18:16	TXSTTH[2:0]	Normal Tx Start Threshold[*2] 0 0 0: Wait for 2 entry DWORDs 0 0 1: Wait for 4 entry DWORDs 0 1 0: Wait for 8 entry DWORDs 0 1 1: Wait for 16 entry DWORDs Others: Setting prohibited	R/W
23:19	—	These bits are read as 0. The write value should be 0.	R/W
26:24	RXSTTH[2:0]	Normal Rx Start Threshold[*2] 0 0 0: Wait for 2 empty DWORDs 0 0 1: Wait for 4 empty DWORDs 0 1 0: Wait for 8 empty DWORDs 0 1 1: Wait for 16 empty DWORDs Others: Setting prohibited	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. These bits support all I3C mode.
Note 2. These bits support I3C master mode and I3C secondary master mode.

The Data Buffer Control register controls the interrupt trigger thresholds for the Rx Data Buffer Queue and the Tx Data Buffer Queue.

TXDBTH[2:0] bits (Normal Tx Data Buffer Threshold)

Minimum number of Tx Data Buffer empties, in DWORDs, that will trigger the I3C_TX interrupt.

The software must program a value less than Tx Data Buffer size in this register.

RXDBTH[2:0] bits (Normal Rx Data Buffer Threshold)

Minimum number of Rx Data Buffer entries in DWORDs that will trigger the I3C_RX interrupt.

The software must program a value less than Rx Data Buffer size in this register.

TXSTTH[2:0] bits (Normal Tx Start Threshold)

When preparing to initiate a Write Transfer on the I3C Bus, I3C shall wait until the Transmit Buffer has at least the indicated number of locations available.

Two optional configurable Modes are available:

1. Store and Forward Mode
   If the TXSTTH[2:0] field is set to the Transmit Buffer size, then I3C shall delay initiation of the Write Command as follows:
   - If the data length to be transferred is more than the Transmit Buffer size, then this module shall wait until the Tx Data Buffer is completely full.
   - If the data length to be transferred is less than the Transmit Buffer size, then I3C shall wait until enough Tx Data Buffer locations are available to store the data to be transferred.
2. Threshold Mode
   If the TXSTTH[2:0] field value is less than the Transmit Buffer size, then I3C shall initiate the Write Command as soon as the indicated number of Tx Data Buffer locations are entries.

RXSTTH[2:0] bits (Normal Rx Start Threshold)

When preparing to initiate a Read Transfer on the I3C bus, I3C shall wait until the Receive Buffer has at least the indicated number of empty locations in DWORDs.

Two optional configurable Modes are available:

1. Store and Forward Mode
   If the RXSTTH[2:0] field is set to the Receive Buffer size, then I3C shall delay initiation of the Read Command as follows:
   - If the data length to be transferred is more than the Receive Buffer size, then this module shall wait until the Rx Data Buffer is completely empty.
   - If the data length to be transferred is less than the Receive Buffer size, then I3C shall wait until enough Rx Data Buffer locations are available to store the data to be transferred.
2. Threshold Mode
   If the RXSTTH[2:0] field value is less than the Receive Buffer size, then I3C shall initiate the Read Command as soon as the indicated number of Rx Data Buffer locations are empty.

33.2.43 NRQTHCTL : Normal Receive Status Queue Threshold Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RSQTH[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
7:0	RSQTH[7:0]	Normal Receive Status Queue Threshold 0x00: Interrupt is issued when Normal Receive Status Queue contains 1 entry (DWORD). Others: Interrupt is triggered when Normal Receive Status Queue contains N+1 entries (DWORD). (N = RSQTH[7:0])	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

RSQTH[7:0] bits (Normal Receive Status Queue Threshold)

Controls the minimum number of normal receive status queue entries needed to trigger the I3C_RCV interrupt.

If this field is greater than (Normal Receive Status Queue size[*1] – 1), then only the number of bits required to address the full buffer depth will be considered.

Note 1. Normal Receive Status Queue size is 2.

33.2.44 HQTHCTL : High Priority Queue Threshold Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1C4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RSPQTH[7:0]								CMDQTH[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
7:0	CMDQTH[7:0]	High Priority Command Queue Threshold 0x00: Interrupt is issued when High Priority Command Queue is completely empty. Others: Interrupt is issued when High Priority Command Queue contains N entries. (N = CMDQTH[7:0])	R/W
15:8	RSPQTH[7:0]	High Priority Response Queue Threshold 0x00: Interrupt is issued when High Priority Response Queue contains 1 entry (DWORD). Others: Interrupt is triggered when High Priority Response Queue contains N+1 entries (DWORD). (N = RSPQTH[7:0])	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

The High Priority Queue Threshold Control register controls the interrupt trigger thresholds for the High Priority Command Queue, the High Priority Response Queue, and the IBI Queue.

The specific reset values are indicative, and could be hardware implementation specific.

CMDQTH[7:0] bits (High Priority Command Queue Threshold)

Controls the minimum number of empty High Priority Command Queue entries needed to trigger the I3C_HCMD interrupt.

If this field is greater than (High Priority Command Queue size[*1] – 1), then only the number of bits required to address the full buffer depth will be considered.

RSPQTH[7:0] bits (High Priority Response Queue Threshold)

Controls the minimum number of High Priority Response Queue entries needed to trigger the I3C_HRESP interrupt.

If this field is greater than (High Priority Response Queue size[*2] – 1), then only the number of bits required to address the full buffer depth will be considered.

Note 1. High Priority Command Queue size is 2.
Note 2. High Priority Response Queue size is 2.

Note: It is assumed that I3C has exactly one High Priority Command Queue, exactly one High Priority Response Queue, and exactly one IBI Queue.

33.2.45 HTBTHCTL : High Priority Transfer Data Buffer Threshold Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1C8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	RXSTTH[2:0]			—	—	—	—	—	TXSTTH[2:0]		
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	RXDBTH[2:0]			—	—	—	—	—	TXDBTH[2:0]		
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
2:0	TXDBTH[2:0]	High Priority Tx Data Buffer Threshold 0 0 0: Interrupt triggers at 2 High Priority Tx Buffer empties, DWORDs 0 0 1: Reserved Others Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
10:8	RXDBTH[2:0]	High Priority Rx Data Buffer Threshold 0 0 0: Interrupt triggers at 2 High Priority Rx Buffer entries, DWORDs 0 0 1: Reserved Others Setting prohibited	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
18:16	TXSTTH[2:0]	High Priority Tx Start Threshold 0 0 0: Wait for 2 entry DWORDs 0 0 1: Reserved Others Setting prohibited	R/W
23:19	—	These bits are read as 0. The write value should be 0.	R/W
26:24	RXSTTH[2:0]	High Priority Rx Start Threshold 0 0 0: Wait for 2 empty DWORDs 0 0 1: Reserved Others Setting prohibited	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

TXDBTH[2:0] bits (High Priority Tx Data Buffer Threshold)

Minimum number of High Priority Tx Data Buffer empties, in DWORDs, that will trigger the I3C_HTX interrupt. The software must program a value less than High Priority Tx Data Buffer size in this register.

RXDBTH[2:0] bits (High Priority Rx Data Buffer Threshold)

Minimum number of High Priority Rx Data Buffer entries in DWORDs that will trigger the I3C_HRX interrupt. The software must program a value less than High Priority Rx Data Buffer size in this register.

TXSTTH[2:0] bits (High Priority Tx Start Threshold)

When preparing to initiate a Write Transfer on the I3C bus, I3C shall wait until the High Priority Transmit Buffer has at least the indicated number of locations available.

Two optional configurable modes are available:

1. Store and Forward Mode

If the TXSTTH[2:0] field is set to the High Priority Transmit Buffer size, then I3C shall delay initiation of the write command as follows:

- If the data length to be transferred is more than the High Priority Transmit Buffer size, then I3C shall wait until the High Priority Tx Data Buffer is completely full.
- If the data length to be transferred is less than the High Priority Transmit Buffer size, then I3C shall wait until enough High Priority Tx Data Buffer locations are available to store the data to be transferred.

2. Threshold mode
If the TXSTTH[2:0] field value is less than the High Priority Transmit Buffer size, then I3C shall initiate the write command as soon as the indicated number of High Priority Tx Data Buffer locations are empty.

RXSTTH[2:0] bits (High Priority Rx Start Threshold)

When preparing to initiate a Read Transfer on the I3C bus, I3C shall wait until the High Priority Receive Buffer has at least the indicated number of empty locations in DWORDs.

Two optional configurable modes are available:

1. Store and forward mode
If the RXSTTH[2:0] field is set to the High Priority Receive Buffer size, then I3C shall delay initiation of the read command as follows:

- If the data length to be transferred is more than the High Priority Receive Buffer size, then I3C shall wait until the High Priority Rx Data Buffer is completely empty.
- If the data length to be transferred is less than the High Priority Receive Buffer size, then I3C shall wait until enough High Priority Rx Data Buffer locations are available to store the data to be transferred.

2. Threshold mode
If the RXSTTH[2:0] field value is less than the High Priority Receive Buffer size, then I3C shall initiate the read command as soon as the indicated number of High Priority Rx Data Buffer locations are empty.

33.2.46 BST : Bus Status Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1D0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	WUCNDDF	—	—	—	TODF	—	—	—	ALF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TENDF	—	—	—	NACKDF	—	HDREXDF	SPCNDDF	STCNDDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STCNDDF	START Condition Detection Flag 0: START condition is not detected. 1: START condition is detected.	R/W*3
1	SPCNDDF	STOP Condition Detection Flag 0: STOP condition is not detected. 1: STOP condition is detected.	R/W*3
2	HDREXDF	HDR Exit Pattern Detection Flag*1 0: HDR Exit Pattern is not detected 1: HDR Exit Pattern is detected.	R/W*3
3	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
4	NACKDF	NACK Detection Flag*2 0: NACK is not detected. 1: NACK is detected.	R/W*3
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	TENDF	Transmit End Flag*2 0: Data is being transmitted. 1: Data has been transmitted.	R/W*3
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	ALF	Arbitration Lost Flag*2 0: Arbitration is not lost 1: Arbitration is lost.	R/W*3
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	TODF	Timeout Detection Flag 0: Timeout is not detected. 1: Timeout is detected.	R/W*3
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	WUCNDDF	Wake-Up Condition Detection Flag 0: Wake-Up is not detected. 1: Wake-Up is detected.	R/W*3
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.
Note 2. This bit supports I^2C mode.
Note 3. Clearing (to 0) condition : Writing 0 after 1 is read.

STCNDDF bit (START Condition Detection Flag)

[Setting conditions]

- All of the followings are satisfied:
  1. The BSTE.STCNDDE bit = 1.
  2. When a START condition (or a Repeated START condition) is detected.

[Clearing conditions]

- When 0 is written to the STCNDDF flag after reading STCNDDF flag = 1.
- When a STOP condition is detected.

SPCNDDF bit (STOP Condition Detection Flag)

[Setting conditions]

- All of the followings are satisfied:
  1. The BSTE.SPCNDDE bit = 1.
  2. When a STOP condition is detected.

[Clearing condition]

- When 0 is written to the SPCNDDF flag after reading SPCNDDF flag = 1.

HDREXDF bit (HDR Exit Pattern Detection Flag)

[Setting conditions]

- All of the followings are satisfied:
  1. The BSTE.HDREXDE bit = 1.
  2. When a HDR EXIT pattern is detected.

[Clearing condition]

- When 0 is written to the HDREXDF flag after reading HDREXDF flag = 1.

NACKDF bit (NACK Detection Flag)

[Setting conditions]

- All of the followings are satisfied:
  1. The PRTS.PRTMD bit = 1 (I^2C protocol mode).
  2. The BSTE.NACKDE bit = 1 (Enables NACK detection interrupt status logging).
  3. When acknowledge is not received (NACK is received) from the receive device in transmit mode.

[Clearing condition]

- When 0 is written to the NACKDF flag after reading NACKDF flag = 1.

TENDF bit (Transmit End Flag)

[Setting conditions]

- All of the followings are satisfied:
  1. The PRTS.PRTMD bit = 1 (I^2C protocol mode).
  2. The BSTE.TENDE bit = 1 (Enables Transmit End Interrupt Status logging).
  3. At the rising edge of the ninth SCL clock cycle while the NTST.TDBEF0 flag = 1.
     Excluding when sending an address.

[Clearing conditions]

- When 0 is written to the TENDF flag after reading TENDF flag = 1.
- When data is written to the NTDTBP0 register.
- When a STOP condition is detected.

ALF bit (Arbitration Lost Flag)

[Setting conditions]

When master arbitration-lost detection is enabled: BSTE.ALE bit = 1, BFCTL.MALE = 1.

- When the internal SDA output state does not match the I3C_SDA line level at the rising edge of SCL clock except for the ACK period during data (including slave address) transmission in master transmit mode (when the I3C_SDA line is driven low while the internal SDA output is at a high level (the I3C_SDA pin is in the highimpedance state)).
- All of the followings are satisfied.
  1. When the START condition is detected while the CNDCTL.STCND bit = 1.
  2. When the internal SDA output state does not match the I3C_SDA line level.
- When the CNDCTL.STCND bit is set to 1 (START condition issuance request) while the BCST.BFREF flag = 0.

When NACK arbitration-lost detection is enabled: BSTE.ALE bit = 1, BFCTL.NALE = 1.

- When the internal SDA output state does not match the I3C_SDA line level at the rising edge of SCL clock in the ACK period during NACK transmission in receive mode.

When slave arbitration-lost detection is enabled: BSTE.ALE bit = 1, BFCTL.SALE = 1.

- When the internal SDA output state does not match the I3C_SDA line level at the rising edge of SCL clock except for the ACK period during data transmission in slave transmit mode.

[Clearing condition]

- When 0 is written to the ALF flag after reading ALF flag = 1.

TODF bit (Timeout Detection Flag)

[Setting conditions]

- All of the followings are satisfied.
  1. The BSTE.TODE bit = 1 (Enables Timeout Detection Interrupt Status logging).
  2. When the master mode or the received slave address matches the slave address n (n = 0 to 2) in Slave mode.
  3. When the I3C_SCL line state remains unchanged for the period specified by TMOCTL register.

[Clearing condition]

- When 0 is written to the TODF flag after reading TODF flag = 1.

WUCNDDF bit (Wake-Up Condition Detection Flag)

[Setting condition]

For I^2C protocol mode: PRTS.PRTMD bit = 1

- When PCLK and TCLK are supplied after all of the followings are satisfied.
  1. The WUCTL.WUFE bit = 1 (Wake-up function is enabled).
  2. The BSTE.WUCNDDE bit = 1 (Enables Wake-up Condition Detection Status logging).
  3. The WUST.WUASYNF flag = 1.
  4. When the address received in slave mode matches the address of slave enabled in the SVCTL.SVAE[2:0] bit (except for the Device-ID address).

For I3C Protocol mode (Master): PRTS.PRTMD bit = 0, PRSST.CRMS bit = 1.

- When PCLK and TCLK are supplied after all of the followings are satisfied.
  1. The WUCTL.WUFE bit = 1 (Wake-up function is enabled).
  2. The BSTE.WUCNDDE bit = 1 (Enables Wake-up Condition Detection Status logging).
  3. The WUST.WUASYNF flag = 1.
  4. When low level of the I3C_SDA line is detected (When the START condition is detected).

For I3C Protocol mode (Slave) : PRTS.PRTMD bit = 0, PRSST.CRMS bit = 0.

- When PCLK and TCLK are supplied after all of the followings are satisfied.
  1. The WUCTL.WUFE bit = 1 (Wake-up function is enabled).
  2. The BSTE.WUCNDDE bit = 1 (Enables Wake-up Condition Detection Status logging).
  3. The WUST.WUASYNF flag = 1.
  4. When the broadcast address (0x7E) is detected after a START (or Repeated START) condition and the own dynamic address is detected after the Repeated START condition following the broadcast address.

[Clearing condition]

- When 0 is written to the WUCNDDF flag after reading WUCNDDF flag = 1 while the WUST.WUASYNF flag = 0.

33.2.47 BSTE : Bus Status Enable Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1D4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	WUCNDDE	—	—	—	TODE	—	—	—	ALE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TENDE	—	—	—	NACKDE	—	HDREXDE	SPCNDDE	STCNDDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STCNDDE	START Condition Detection Enable 0: Disables START condition Detection Interrupt Status logging. 1: Enables START condition Detection Interrupt Status logging.	R/W
1	SPCNDDE	STOP Condition Detection Enable 0: Disables STOP condition Detection Interrupt Status logging. 1: Enables STOP condition Detection Interrupt Status logging.	R/W
2	HDREXDE	HDR Exit Pattern Detection Enable*1 0: Disables HDR Exit Pattern Detection Interrupt Status logging. 1: Enables HDR Exit Pattern Detection Interrupt Status logging.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	NACKDE	NACK Detection Enable*2 0: Disables NACK Detection Interrupt Status logging. 1: Enables NACK Detection Interrupt Status logging.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	TENDE	Transmit End Enable*2 0: Disables Transmit End Interrupt Status logging. 1: Enables Transmit End Interrupt Status logging.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	ALE	Arbitration Lost Enable*2 0: Disables Arbitration Lost Interrupt Status logging. 1: Enables Arbitration Lost Interrupt Status logging.	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	TODE	Timeout Detection Enable 0: Disables Timeout Detection Interrupt Status logging. 1: Enables Timeout Detection Interrupt Status logging.	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	WUCNDDE	Wake-up Condition Detection Enable 0: Disables Wake-up Condition Detection Status logging. 1: Enables Wake-up Condition Detection Status logging.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.
Note 2. This bit supports I^2C mode.

STCNDDE bit (START Condition Detection Enable)

When this bit is 1, operation of BST.STCNDDF is enabled. For the setting conditions and clearing conditions of the BST.STCNDDF flag, see the details of BST.STCNDDF.

SPCNDDE bit (STOP Condition Detection Enable)

When this bit is 1, operation of BST.SPCNDDF is enabled. For the setting conditions and clearing conditions of the BST.SPCNDDF flag, see the details of BST.SPCNDDF.

HDREXDE bit (HDR Exit Pattern Detection Enable)

When this bit is 1, the operation of BST.HDREXDF is enabled. For the setting conditions and clearing conditions of the BST.HDREXDF flag, see the details of BST.HDREXDF.

NACKDE bit (NACK Detection Enable)

When this bit is 1, the operation of BST.NACKDF is enabled. This bit is used to specify whether to continue or discontinue the transfer operation when NACK is received from the slave device in transmit mode. Normally, set this bit to 1. For the setting conditions and clearing conditions of the BST.NACKDF flag, see the details of BST.NACKDF.

TENDE bit (Transmit End Enable)

When this bit is 1, the operation of BST.TENDF is enabled. For the setting conditions and clearing conditions of the BST.TENDF flag, see the details of BST.TENDF.

ALE bit (Arbitration Lost Enable)

When this bit is 1, the operation of BST.ALF is enabled. For the setting conditions and clearing conditions of the BST.ALF flag, see the details of BST.ALF.

TODE bit (Timeout Detection Enable)

When this bit is 1, the operation of BST.TODF is enabled. For the setting conditions and clearing conditions of the BST.TODF flag, see the details of BST.TODF.

WUCNDDE bit (Wake-up Condition Detection Enable)

When this bit is 1, the operation of BST.WUCNDDF is enabled. For the setting conditions and clearing conditions of the BST.WUCNDDF flag, see the details of BST.WUCNDDF.

33.2.48 BIE : Bus Interrupt Enable Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1D8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	WUCNDDIE	—	—	—	TODIE	—	—	—	ALIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TENDIE	—	—	—	NACKDIE	—	HDREXDIE	SPCNDDIE	STCNDDIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STCNDDIE	START Condition Detection Interrupt Enable 0: Disables START condition Detection Interrupt Signal. 1: Enables START condition Detection Interrupt Signal.	R/W
1	SPCNDDIE	STOP Condition Detection Interrupt Enable 0: Disables STOP condition Detection Interrupt Signal. 1: Enables STOP condition Detection Interrupt Signal.	R/W
2	HDREXDIE	HDR Exit Pattern Detection Interrupt Enable*1 0: Disables HDR Exit Pattern Detection Interrupt Signal. 1: Enables HDR Exit Pattern Detection Interrupt Signal.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
4	NACKDIE	NACK Detection Interrupt Enable*2 0: Disables NACK Detection Interrupt Signal. 1: Enables NACK Detection Interrupt Signal.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	TENDIE	Transmit End Interrupt Enable*2 0: Disables Transmit End Interrupt Signal. 1: Enables Transmit End Interrupt Signal.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	ALIE	Arbitration Lost Interrupt Enable*2 0: Disables Arbitration Lost Interrupt Signal. 1: Enables Arbitration Lost Interrupt Signal.	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	TODIE	Timeout Detection Interrupt Enable 0: Disables Timeout Detection Interrupt Signal. 1: Enables Timeout Detection Interrupt Signal.	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	WUCNDDIE	Wake-Up Condition Detection Interrupt Enable 0: Disables Wake-Up Condition Detection Interrupt Signal. 1: Enables Wake-Up Condition Detection Interrupt Signal.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.
Note 2. This bit supports I^2C mode.

The BIE register enables signaling of outstanding bus interrupts received by I3C.

STCNDDIE bit (START Condition Detection Interrupt Enable)

This bit enables or disables the START Condition Detection interrupt requests when the BST.STCNDDF flag is set to 1.

SPCNDDIE bit (STOP Condition Detection Interrupt Enable)

This bit enables or disables the STOP Condition Detection interrupt requests when the BST.SPCNDDF flag is set to 1.

HDREXDIE bit (HDR Exit Pattern Detection Interrupt Enable)

This bit enables or disables the HDR Exit Pattern Detection interrupt requests when the BST.HDREXDF flag is set to 1.

NACKDIE bit (NACK Detection Interrupt Enable)

This bit enables or disables the NACK Detection interrupt requests when the BST.NACKDF flag is set to 1.

TENDIE bit (Transmit End Interrupt Enable)

This bit enables or disables the Transmit End interrupt (I3C_TEND) requests when the BST.TENDF flag is set to 1.

ALIE bit (Arbitration Lost Interrupt Enable)

This bit enables or disables the Arbitration lost interrupt requests when the BST.ALF flag is set to 1.

TODIE bit (Timeout Detection Interrupt Enable)

This bit enables or disables the Timeout Detection interrupt requests when the BST.TODF flag is set to 1.

WUCNDDIE bit (Wake-Up Condition Detection Interrupt Enable)

This bit enables or disables the Wake-up Condition Detection interrupt (I3C_WU) requests when the BST.WUCNDDF flag is set to 1.

33.2.49 BSTFC : Bus Status Force Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1DC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	WUCNDDFC	—	—	—	TODFC	—	—	—	ALFC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TENDFC	—	—	—	NACKDFC	—	HDREXDFC	SPCNDDFC	STCNDDFC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STCNDDFC	START condition Detection Force 0: Not Force START condition Detection Interrupt for software testing. 1: Force START condition Detection Interrupt for software testing.	W
1	SPCNDDFC	STOP condition Detection Force 0: Not Force STOP condition Detection Interrupt for software testing. 1: Force STOP condition Detection Interrupt for software testing.	W
2	HDREXDFC	HDR Exit Pattern Detection Force*1 0: Not Force HDR Exit Pattern Detection Interrupt for software testing. 1: Force HDR Exit Pattern Detection Interrupt for software testing.	W
3	—	This bit is read as 0.	R
4	NACKDFC	NACK Detection Force*2 0: Not Force NACK Detection Interrupt for software testing. 1: Force NACK Detection Interrupt for software testing.	W
7:5	—	These bits are read as 0.	R
8	TENDFC*3	Transmit End Force*2 0: Not Force Transmit End Interrupt for software testing. 1: Force Transmit End Interrupt for software testing.	W
15:9	—	These bits are read as 0.	R
16	ALFC	Arbitration Lost Force*2 0: Not Force Arbitration Lost Interrupt for software testing. 1: Force Arbitration Lost Interrupt for software testing.	W
19:17	—	These bits are read as 0.	R
20	TODFC	Timeout Detection Force 0: Not Force Timeout Detection Interrupt for software testing. 1: Force Timeout Detection Interrupt for software testing.	W
23:21	—	These bits are read as 0.	R
24	WUCNDDFC	Wake-Up Condition Detection Force 0: Not Force Wake-Up Condition Detection Interrupt for software testing. 1: Force Wake-Up Condition Detection Interrupt for software testing.	W
31:25	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.
Note 2. This bit supports I^2C mode.
Note 3. TENDFC does not work unless TDBEF0 = 1.

33.2.50 NTST : Normal Transfer Status Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1E0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	RSQFF	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TEF	—	—	—	TABTF	RSPQFF	CMDQEF	IBIQEFF	RDBFF0	TDBEF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEF0	Normal Tx Data Buffer Empty Flag 0 0: For I2C protocol mode: PRTS.PRTMD bit = 1. Normal Tx Data Buffer contains transmit data. For I3C protocol mode: PRTS.PRTMD bit = 0. The number of empties in the Normal Tx Data Buffer is less than the NTBTHCTL0.TXDBTH[2:0] threshold. 1: For I2C protocol mode: PRTS.PRTMD bit = 1. Normal Tx Data Buffer contains no transmit data. For I3C protocol mode: PRTS.PRTMD bit = 0. The number of empties in the Normal Tx Data Buffer is the NTBTHCTL0.TXDBTH[2:0] threshold or more.	R/W*3
1	RDBFF0	Normal Rx Data Buffer Full Flag 0 0: For I2C protocol mode: PRTS.PRTMD bit = 1. Normal Rx Data Buffer contains no receive data. For I3C Protocol mode: PRTS.PRTMD bit = 0. The number of entries in the Normal Rx Data Buffer is less than the NTBTHCTL0.RXDBTH[2:0] threshold. 1: For I2C protocol mode: PRTS.PRTMD bit = 1. Normal Rx Data Buffer contains receive data. For I3C Protocol mode: PRTS.PRTMD bit = 0. The number of entries in the Normal Rx Data Buffer is the NTBTHCTL0.RXDBTH[2:0] threshold or more.	R/W*3
2	IBIQEFF	Normal IBI Queue Empty/Full Flag*1 0: For I3C protocol mode (Master): PRTS.PRTMD bit = 0, PRSST.CRMS bit = 1. The number of Normal IBI Queue entries is the NQTHCTL.IBIQTH threshold or less. For I3C protocol mode (Slave) : PRTS.PRTMD bit = 0, PRSST.CRMS bit = 0. If the NQTHCTL.IBIQTH = 0: The number of IBI Data Buffer empties is less than the IBI Data Buffer size. If the NQTHCTL.IBIQTH is other than 0: The number of IBI Data Buffer empties is less than the NQTHCTL.IBIQTH threshold. 1: For I3C protocol mode (Master): PRTS.PRTMD bit = 0, PRSST.CRMS bit = 1. The number of Normal IBI Queue entries is more than the NQTHCTL.IBIQTH threshold. For I3C protocol mode (Slave) : PRTS.PRTMD bit = 0, PRSST.CRMS bit = 0. If the NQTHCTL.IBIQTH = 0: The number of IBI Data Buffer empties is the IBI Data Buffer size. If the NQTHCTL.IBIQTH is other than 0: The number of IBI Data Buffer empties is the NQTHCTL.IBIQTH threshold or more.	R/W*3

Bit	Symbol	Function	R/W
3	CMDQEF	Normal Command Queue Empty Flag*1 0: If the NQTHCTL.CMDQTH = 0: The number of Normal Command Queue empties is less than the Normal Command Queue size. If the NQTHCTL.CMDQTH is other than 0: The number of Normal Command Queue empties is less than the NQTHCTL.CMDQTH threshold. 1: If the NQTHCTL.CMDQTH = 0: The number of Normal Command Queue empties is the Normal Command Queue size. If the NQTHCTL.CMDQTH is other than 0: 1: The number of Normal Command Queue empties is the NQTHCTL.CMDQTH threshold or more.	R/W*3
4	RSPQFF	Normal Response Queue Full Flag*1 0: The number of Normal Response Queue entries is the NQTHCTL.RSPQTH threshold or less. 1: The number of Normal Response Queue entries is more than the NQTHCTL.RSPQTH threshold.	R/W*3
5	TABTF	Normal Transfer Abort Flag*1 0: Normal Transfer Abort does not occur. 1: Normal Transfer Abort occur. To clear, write 0 to this bit after 1 state is read.	R/W*3
8:6	—	These bits are read as 0. The write value should be 0.	R/W
9	TEF	Normal Transfer Error Flag*1 0: Normal Transfer Error does not occur. 1: Normal Transfer Error occurs. To clear, write 0 to this bit after 1 state is read.	R/W*3
19:10	—	These bits are read as 0. The write value should be 0.	R/W
20	RSQFF	Normal Receive Status Queue Full Flag*2 0: The number of Normal Receive Status Queue entries is the NRQTHCTL.RSQTH threshold or less. 1: The number of Normal Receive Status Queue entries is more than the NRQTHCTL.RSQTH threshold.	R/W*3
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.
Note 2. This bit supports I3C secondary master mode and I3C slave mode.
Note 3. Clearing (to 0) condition : Writing 0 after the 1 state is read.

TDBEF0 bit (Normal Tx Data Buffer Empty Flag 0)

[Setting conditions]

For I^2C Protocol mode: PRTS.PRTMD bit = 1.

The following condition 1 is satisfied and any of the following conditions 2 to 4 are satisfied:

1. The NTSTE.TDBEE0 bit = 1 (enables Normal Tx Data Buffer Empty Interrupt Status logging).
2. When data has been transferred from the Normal Tx Data Buffer to the Shift Register and the Normal Tx Data Buffer becomes empty*1.
3. When the PRSST.TRMD bit is set to 1.
4. When the received slave address matches while the TRMD bit = 1.

For I3C Protocol mode: PRTS.PRTMD bit = 0.

The following conditions 1 and 2 are satisfied:

1. The NTSTE.TDBEE0 bit = 1 (enables Normal Tx Data Buffer Empty Interrupt Status logging).
2. When the number of empties in the Normal Tx Data Buffer is the NTBTHCTL0.TXDBTH[2:0] threshold or more (see NTBTHCTL0 register).

[Clearing conditions]

For I2C Protocol mode: PRTS.PRTMD bit = 1.

- When data is written to NTDTBP0.
- When the TRMD bit in PRSST is set to 0.

For I3C protocol mode: PRTS.PRTMD bit = 0.

- Write 0 to this bit after 1 is read.
- On completion of the last write access to Normal Tx Data by DMAC/DTC.

Note 1. When the BST.NACKDF flag is set to 1 while the BSTE.NACKDE bit = 1, I3C aborts data transmission/reception. If the TDBEF0 flag = 0 (next transmit data has been written), data is transferred to the Shift Register and the Normal Tx Data Buffer register becomes empty at the rising edge of the 9th clock cycle, but the TDBEF0 flag is not set to 1.

RDBFF0 bit (Normal Rx Data Buffer Full Flag 0)

[Setting conditions]

For I2C Protocol mode: PRTS.PRTMD bit = 1.

The following condition 1 is satisfied and any of the following condition 2 or 3 is satisfied:

1. The NTSTE.RDBFE0 bit = 1 (enables Normal Rx Data Buffer Full Interrupt Status logging).
2. When Rx data is transferred from Shift Register to Normal Rx Data Buffer.
   The RDBFF0 flag is set to 1 on the rising edge of the 8th or 9th SCL clock cycle (selected in the ACKTWE bit in SCSTRCTL).
3. When the received slave address matches after a START (or Repeated START) condition is detected with the TRMD bit in PRSST set to 0.

For I3C Protocol mode: PRTS.PRTMD bit = 0.

The following conditions 1 and 2 are satisfied:

1. The NTSTE.RDBFE0 bit = 1 (enables Normal Rx Data Buffer Full Interrupt Status logging).
2. When the number of Normal Rx Data Buffer entries is the NTBTHCTL0.RXDBTH[2:0] threshold or more (see NTBTHCTL0 register).

[Clearing conditions]

For I2C Protocol mode: PRTS.PRTMD bit = 1.

- When data is read from NTDTBP0.

For I3C Protocol mode: PRTS.PRTMD bit = 0.

- Write 0 to this bit after 1 is read.
- On completion of the last read access to Normal Rx Data by DMAC/DTC.

IBIQEFF bit (Normal IBI Queue Empty/Full Flag)

[Setting conditions]

The following 2 conditions are satisfied:

1. The NTSTE.IBIQEFE bit = 1 (enables Normal IBI Queue Empty/Full Interrupt Status logging)
2. For I3C protocol mode (master): PRTS.PRTMD bit = 0, PRSST.CRMS bit = 1.
   - When the number of Normal IBI Queue entries is more than the NQTHCTL.IBIQTH threshold (see NQTHCTL register).

   For I3C protocol mode (slave) : PRTS.PRTMD bit = 0, PRSST.CRMS bit = 0.
   If the NQTHCTL.IBIQTH = 0:
   - When IBI Data Buffer is completely empty.

If the NQTHCTL.IBIQTH is other than 0:

- When the number of IBI Data Buffer empties is the NQTHCTL.IBIQTH threshold or more (see NQTHCTL register).

[Clearing conditions]

For I3C protocol mode (master): PRTS.PRTMD bit = 0, PRSST.CRMS bit = 1.

- Write 0 to this bit after 1 is read.
- On completion of the last read access to IBI Status by DMAC/DTC.

For I3C protocol mode (slave): PRTS.PRTMD bit = 0, PRSST.CRMS bit = 0.

- Write 0 to this bit after 1 is read.
- On completion of the last write access to IBI Status by DMAC/DTC.

CMDQEF bit (Normal Command Queue Empty Flag)

[Setting conditions]

The following 2 conditions are satisfied:

1. The NTSTE.CMDQEE bit = 1 (enables Normal Command Queue Empty Interrupt Status logging).
2. If the NQTHCTL.CMDQTH = 0:
   - When Normal Command Queue is completely empty.

   If the NQTHCTL.CMDQTH is other than 0:
   - When the number of Normal Command Queue empties is the NQTHCTL.CMDQTH threshold or more (see NQTHCTL register).

[Clearing conditions]

- Write 0 to this bit after 1 is read.
- On completion of the last write access to Normal Command by DMAC/DTC.

RSPQFF bit (Normal Response Queue Full Flag)

[Setting conditions]

The following 2 conditions are satisfied:

1. The NTSTE.RSPQFE bit = 1 (enables Normal Response Queue Full Interrupt Status logging).
2. When the number of Normal Response Queue entries is more than the NQTHCTL.RSPQTH threshold (see NQTHCTL register).

[Clearing conditions]

- Write 0 to this bit after 1 is read.
- On completion of the last read access to Normal Receive Status by DMAC/DTC.

TABTF bit (Normal Transfer Abort Flag)

[Setting conditions]

The following 2 conditions are satisfied:

1. The NTSTE.TABTE bit = 1 (enables Normal Transfer Abort Interrupt Status logging).
2. When any transfer is aborted.

[Clearing condition]

- Write 0 to this bit after 1 is read.

TEF bit (Normal Transfer Error Flag)

[Setting conditions]

The following 2 conditions are satisfied:

1. The NTSTE.TEE bit = 1 (enables Normal Transfer Error Interrupt Status logging).
2. When any transfer error occurs on the I3C bus. The Error type for this error is available in the Response or Receive Status structure corresponding to the Transfer command.

[Clearing condition]

- Write 0 to this bit after 1 is read.

RSQFF bit (Normal Receive Status Queue Full Flag)

[Setting conditions]

The following 2 conditions are satisfied:

1. The NTSTE.RSQFE bit = 1 (Normal Receive Status Queue Full Enable).
2. When the number of Normal Receive Status Queue entries is more than the NRQTHCTL.RSQTH threshold (see NRQTHCTL register).

[Clearing conditions]

- Write 0 to this bit after 1 is read.
- On completion of the last read access to Normal Receive Status by DMAC/DTC.

33.2.51 NTSTE : Normal Transfer Status Enable Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1E4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	RSQFE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TEE	—	—	—	TABTE	RSPQFE	CMDQEE	IBIQEFE	RDBFE0	TDBEE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEE0	Normal Tx Data Buffer Empty Enable 0 0: Disables Normal Tx Data Buffer Empty Interrupt Status logging. 1: Enables Normal Tx Data Buffer Empty Interrupt Status logging.	R/W
1	RDBFE0	Normal Rx Data Buffer Full Enable 0 0: Disables Normal Rx Data Buffer Full Interrupt Status logging. 1: Enables Normal Rx Data Buffer Full Interrupt Status logging.	R/W
2	IBIQEFE	Normal IBI Queue Empty/Full Enable*1 0: Disables Normal IBI Queue Empty/Full Interrupt Status logging. 1: Enables Normal IBI Queue Empty/Full Interrupt Status logging.	R/W
3	CMDQEE	Normal Command Queue Empty Enable*1 0: Disables Normal Command Queue Empty Interrupt Status logging. 1: Enables Normal Command Queue Empty Interrupt Status logging.	R/W
4	RSPQFE	Normal Response Queue Full Enable*1 0: Disables Normal Response queue Full Interrupt Status logging. 1: Enables Normal Response queue Full Interrupt Status logging.	R/W

Bit	Symbol	Function	R/W
5	TABTE	Normal Transfer Abort Enable*1 0: Disables Normal Transfer Abort Interrupt Status logging. 1: Enables Normal Transfer Abort Interrupt Status logging.	R/W
8:6	—	These bits are read as 0. The write value should be 0.	R/W
9	TEE	Normal Transfer Error Enable*1 0: Disables Normal Transfer Error Interrupt Status logging. 1: Enables Normal Transfer Error Interrupt Status logging.	R/W
19:10	—	These bits are read as 0. The write value should be 0.	R/W
20	RSQFE	Normal Receive Status Queue Full Enable*2 0: Disables Normal Receive Status Queue Full Interrupt Status logging. 1: Enables Normal Receive Status Queue Full Interrupt Status logging.	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.
Note 2. This bit supports I3C secondary master mode and I3C slave mode.

TDBEE0 bit (Normal Tx Data Buffer Empty Enable 0)

When this bit is 1, the operation of NTST.TDBEF0 is enabled.

For the setting conditions and clearing conditions of the NTST.TDBEF0 flag, see the details of NTST.TDBEF0.

RDBFE0 bit (Normal Rx Data Buffer Full Enable 0)

When this bit is 1, the operation of NTST.RDBFF0 is enabled.

For the setting conditions and clearing conditions of the NTST.RDBFF0 flag, see the details of NTST.RDBFF0.

IBIQEFE bit (Normal IBI Queue Empty/Full Enable)

When this bit is 1, the operation of NTST.IBIQEFF is enabled.

For the setting conditions and clearing conditions of the NTST.IBIQEFF flag, see the details of NTST.IBIQEFF.

CMDQEE bit (Normal Command Queue Empty Enable)

When this bit is 1, the operation of NTST.CMDQEF is enabled.

For the setting conditions and clearing conditions of the NTST.CMDQEF flag, see the details of NTST.CMDQEF.

RSPQFE bit (Normal Response Queue Full Enable)

When this bit is 1, the operation of NTST.RSPQFF is enabled.

For the setting conditions and clearing conditions of the NTST.RSPQFF flag, see the details of NTST.RSPQFF.

TABTE bit (Normal Transfer Abort Enable)

When this bit is 1, the operation of NTST.TABTF is enabled.

For the setting conditions and clearing conditions of the NTST.TABTF flag, see the details of NTST.TABTF.

TEE bit (Normal Transfer Error Enable)

When this bit is 1, the operation of NTST.TEF is enabled.

For the setting conditions and clearing conditions of the NTST.TEF flag, see the details of NTST.TEF.

RSQFE bit (Normal Receive Status Queue Full Enable)

When this bit is 1, the operation of NTST.RSQFF is enabled.

For the setting conditions and clearing conditions of the NTST.RSQFF flag, see the details of NTST.RSQFF.

33.2.52 NTIE : Normal Transfer Interrupt Enable Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1E8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	RSQFIE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TEIE	—	—	—	TABTIE	RSPQFIE	CMDQEIE	IBIQEFIE	RDBFIE0	TDBEIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEIE0	Normal Tx Data Buffer Empty Interrupt Enable 0 0: Disables Normal Tx Data Buffer Empty Interrupt Signal. 1: Enables Normal Tx Data Buffer Empty Interrupt Signal.	R/W
1	RDBFIE0	Normal Rx Data Buffer Full Interrupt Enable 0 0: Disables Normal Rx Data Buffer Full Interrupt Signal. 1: Enables Normal Rx Data Buffer Full Interrupt Signal.	R/W
2	IBIQEFIE	Normal IBI Queue Empty/Full Interrupt Enable*1 0: Disables Normal IBI Queue Empty/Full Interrupt Signal. 1: Enables Normal IBI Queue Empty/Full Interrupt Signal.	R/W
3	CMDQEIE	Normal Command Queue Empty Interrupt Enable*1 0: Disables Normal Command Queue Empty Interrupt Signal. 1: Enables Normal Command Queue Empty Interrupt Signal.	R/W
4	RSPQFIE	Normal Response Queue Full Interrupt Enable*1 0: Disables Normal Response Queue Full Interrupt Signal. 1: Enables Normal Response Queue Full Interrupt Signal.	R/W
5	TABTIE	Normal Transfer Abort Interrupt Enable*1 0: Disables Normal Transfer Abort Interrupt Signal. 1: Enables Normal Transfer Abort Interrupt Signal.	R/W
8:6	—	These bits are read as 0. The write value should be 0.	R/W
9	TEIE	Normal Transfer Error Interrupt Enable*1 0: Disables Normal Transfer Error Interrupt Signal. 1: Enables Normal Transfer Error Interrupt Signal.	R/W
19:10	—	These bits are read as 0. The write value should be 0.	R/W
20	RSQFIE	Normal Receive Status Queue Full Interrupt Enable*2 0: Disables Normal Receive Status Queue Full Interrupt Signal. 1: Enables Normal Receive Status Queue Full Interrupt Signal.	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.
Note 2. This bit supports I3C secondary master mode and I3C slave mode.

The PIO Interrupt Signal Enable register enables signaling of outstanding interrupts received by I3C.

TDBEIE0 bit (Normal Tx Data Buffer Empty Interrupt Enable 0)

This bit is used to enable or disable the Normal Tx Data buffer empty interrupt (I3C_TX) requests when the NTST.TDBEF0 flag is set to 1.

RDBFIE0 bit (Normal Rx Data Buffer Full Interrupt Enable 0)

This bit is used to enable or disable the Normal Rx Data buffer full interrupt (I3C_RX) requests when the NTST.RDBFF0 flag is set to 1.

IBIQEFIE bit (Normal IBI Queue Empty/Full Interrupt Enable)

This bit is used to enable or disable the Normal IBI Queue Empty/Full interrupt (I3C_IBI) requests when the NTST.IBIQEFF flag is set to 1.

CMDQEIE bit (Normal Command Queue Empty Interrupt Enable)

This bit is used to enable or disable the Normal Command queue empty interrupt (I3C_CMD) requests when the NTST.CMDQEF flag is set to 1.

RSPQFIE bit (Normal Response Queue Full Interrupt Enable)

This bit is used to enable or disable the Normal Response queue full interrupt (I3C_RESP) requests when the NTST.RSPQFF flag is set to 1.

TABTIE bit (Normal Transfer Abort Interrupt Enable)

This bit is used to enable or disable the Normal Transfer Abort interrupt (I3C_EEI) requests when the NTST.TABTF flag is set to 1.

TEIE bit (Normal Transfer Error Interrupt Enable)

This bit is used to enable or disable the Normal Transfer Error interrupt (I3C_EEI) requests when the NTST.TEF flag is set to 1.

RSQFIE bit (Normal Receive Status Queue Full Interrupt Enable)

This bit is used to enable or disable the Normal Receive Status queue full interrupt (I3C_RCV) requests when the NTST.RSQFF flag is set to 1.

33.2.53 NTSTFC : Normal Transfer Status Force Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1EC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	RSQFFC	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TEFC	—	—	—	TABTFC	RSPQFFC	CMDQEFC	IBIQEFFC	RDBFFC0	TDBEFC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEFC0	Normal Tx Data Buffer Empty Force 0 0: Not Force Normal Tx Data Buffer Empty Interrupt for software testing. 1: Force Normal Tx Data Buffer Empty Interrupt for software testing.	W
1	RDBFFC0	Normal Rx Data Buffer Full Force 0 0: Not Force Normal Rx Data Buffer Full Interrupt for software testing. 1: Force Normal Rx Data Buffer Full Interrupt for software testing.	W
2	IBIQEFFC	Normal IBI Queue Empty/Full Force*1 0: Not Force Normal IBI Queue Empty/Full Interrupt for software testing. 1: Force Normal IBI Queue Empty/Full Interrupt for software testing.	W
3	CMDQEFC	Normal Command Queue Empty Force*1 0: Not Force Normal Command Queue Empty Interrupt for software testing. 1: Force Normal Command Queue Empty Interrupt for software testing.	W
4	RSPQFFC	Normal Response Queue Full Force*1 0: Not Force Normal Response Queue Full Interrupt for software testing. 1: Force Normal Response Queue Full Interrupt for software testing.	W

Bit	Symbol	Function	R/W
5	TABTFC	Normal Transfer Abort Force*1 0: Not Force Normal Transfer Abort Interrupt for software testing. 1: Force Normal Transfer Abort Interrupt for software testing.	W
8:6	—	These bits are read as 0.	R
9	TEFC	Normal Transfer Error Force*1 0: Not Force Normal Transfer Error Interrupt for software testing. 1: Force Normal Transfer Error Interrupt for software testing.	W
19:10	—	The write value should be 0.	W
20	RSQFFC	Normal Receive Status Queue Full Force*2 0: Not Force Normal Receive Status Queue Full Interrupt for software testing. 1: Force Normal Receive Status Queue Full Interrupt for software testing.	W
31:21	—	The write value should be 0.	W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.
Note 2. This bit supports I3C secondary master mode and I3C slave mode.

The PIO Interrupt Force register is used to force specific interrupt. It can be used for debug purposes.

TDBEFC0 bit (Normal Tx Data Buffer Empty Force 0)

For software testing, when set to 1, forces the corresponding interrupt, subject to TDBEE0 and TDBEIE0 configuration.

RDBFFC0 bit (Normal Rx Data Buffer Full Force 0)

For software testing, when set to 1, forces the corresponding interrupt, subject to RDBFE0 and RDBFIE0 configuration.

IBIQEFFC bit (Normal IBI Queue Empty/Full Force)

For software testing, when set to 1, forces the corresponding interrupt, subject to IBIQEFE and IBIQEFIE configuration.

CMDQEFC bit (Normal Command Queue Empty Force)

For software testing, when set to 1, forces the corresponding interrupt, subject to CMDQEE and CMDQEIE configuration.

RSPQFFC bit (Normal Response Queue Full Force)

For software testing, when set to 1, forces the corresponding interrupt, subject to RSPQFE and RSPQFIE configuration.

TABTFC bit (Normal Transfer Abort Force)

For software testing, forces the corresponding interrupt, subject to TABTE and TABTIE configuration.

TEFC bit (Normal Transfer Error Force)

For software testing, when set to 1, forces the corresponding interrupt, subject to TEE and TEIE configuration.

RSQFFC bit (Normal Receive Status Queue Full Force)

For software testing, when set to 1, forces the corresponding interrupt, subject to RSQFE and RSQFIE configuration.

33.2.54 HTST : High Priority Transfer Status Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x200

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TEF	—	—	—	TABTF	RSPQFF	CMDQEF	—	RDBFF	TDBEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEF	High Priority Tx Data Buffer Empty Flag 0: The number of empties in the High Priority Tx Data Buffer is less than the HTBTHCTL.TXDBTH[2:0] threshold. 1: The number of empties in the High Priority Tx Data Buffer is or more than the HTBTHCTL.TXDBTH[2:0] threshold.	R/W*1
1	RDBFF	High Priority Rx Data Buffer Full Flag 0: The number of entries in the High Priority Rx Data Buffer is less than the HTBTHCTL.RXDBTH[2:0] threshold. 1: The number of entries in the High Priority Rx Data Buffer is or more than the HTBTHCTL.RXDBTH[2:0] threshold.	R/W*1
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CMDQEF	High Priority Command Queue Empty Flag 0: If HQTHCTL.CMDQTH is 0, the number of High Priority Command Queue empties is less than the Command Queue size. If HQTHCTL.CMDQTH is other than 0, the number of High Priority Command Queue empties is less than the HQTHCTL.CMDQTH threshold. 1: If HQTHCTL.CMDQTH is 0, the number of High Priority Command Queue empties is the Command Queue size. If HQTHCTL.CMDQTH is other than 0, the number of High Priority Command Queue empties is or more than the HQTHCTL.CMDQTH threshold .	R/W*1
4	RSPQFF	High Priority Response Queue Full Flag 0: The number of High Priority Response Queue entries is less than the HQTHCTL .RSPQTH threshold. 1: The number of High Priority Response Queue entries is or more than the HQTHCTL .RSPQTH threshold.	R/W*1
5	TABTF	High Priority Transfer Abort Flag 0: High Priority Transfer Abort does not occur. 1: High Priority Transfer Abort occurs. To clear, write 0 to this bit after 1 is read.	R/W*1
8:6	—	These bits are read as 0. The write value should be 0.	R/W
9	TEF	High Priority Transfer Error Flag 0: High Priority Transfer Error does not occur. 1: High Priority Transfer Error occurs. To clear, write 0 to this bit after 1 is read.	R/W*1
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.
Note 1. Clearing (to 0) condition : Writing 0 after the 1 state is read.

TDBEF bit (High Priority Tx Data Buffer Empty Flag)

[Setting condition]

The following 2 conditions are satisfied:

1. The HTSTE.TDBEE bit = 1 (Enables High Priority Tx Data Buffer Empty Interrupt Status Logging).
2. When the number of empties in the High Priority Tx Data Buffer is the HTBTHCTL.TXDBTH[2:0] threshold (see register HTBTHCTL) or more.

[Clearing condition]

- Write 0 to this bit after 1 is read.
- On completion of the last write access to “High Priority Tx Data” by DMAC/DTC.

RDBFF bit (High Priority Rx Data Buffer Full Flag)

[Setting condition]

The following 2 conditions are satisfied:

1. The HTSTE.RDBFE bit = 1 (Enables High Priority Rx Data Buffer Full Interrupt Status Logging).
2. When the number of High Priority Rx Data Buffer entries is >= the HTBTHCTL.RXDBTH[2:0] threshold (see register HTBTHCTL).

[Clearing condition]

- Write 0 to this bit after 1 is read.
- On completion of the last read access to “High Priority Rx Data” by DMAC/DTC.

CMDQEF bit (High Priority Command Queue Empty Flag)

[Setting condition]

The following 2 conditions are satisfied:

1. The HTSTE.CMDQEE bit = 1 (Enables High Priority Command Queue Empty Interrupt Status Logging).
2. If the HQTHCTL.CMDQTH = 0:
   - When High Priority Command Queue is completely empty

   If the HQTHCTL.CMDQTH is other than 0:
   - When the number of High Priority Command Queue empties is the HQTHCTL.CMDQTH threshold (see register HQTHCTL) or more.

[Clearing condition]

- Write 0 to this bit after 1 is read.
- On completion of the last write access to “High Priority Command” by DMAC/DTC.

RSPQFF bit (High Priority Response Queue Full Flag)

[Setting condition]

The following 2 conditions are satisfied:

1. The HTSTE.RSPQFE bit = 1 (Enables High Priority Response Queue Full Interrupt Status Logging).
2. When the number of High Priority Response Queue entries is > the HQTHCTL.RSPQTH threshold (see register HQTHCTL).

[Clearing condition]

- Write 0 to this bit after 1 is read.
- On completion of the last read access to “High Priority Receive Status” by DMAC/DTC.

See the description of the RSPQFF bit for the elements used.

TABTF bit (High Priority Transfer Abort Flag)

[Setting condition]

The following 2 conditions are satisfied:

1. The HTSTE.TABTE bit = 1 (Enables High Priority Transfer Abort Interrupt Status Logging).
2. When any transfer is aborted.

[Clearing condition]

- Write 0 to this bit after 1 is read.

TEF bit (High Priority Transfer Error Flag)

[Setting condition]

The following 2 conditions are satisfied:

1. The HTSTE.TEE bit = 1 (Enables High Priority Transfer Error Interrupt Status Logging).
2. When any transfer error occurs on the I3C Bus. The error type for this error is available in the Response structure corresponding to this transfer/command.

[Clearing condition]

- Write 0 to this bit after 1 is read.

33.2.55 HTSTE : High Priority Transfer Status Enable Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x204

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TEE	—	—	—	TABTE	RSPQFE	CMDQEE	—	RDBFE	TDBEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEE	High Priority Tx Data Buffer Empty Enable 0: Disables High Priority Tx Data Buffer Empty Interrupt Status logging. 1: Enables High Priority Tx Data Buffer Empty Interrupt Status logging.	R/W
1	RDBFE	High Priority Rx Data Buffer Full Enable 0: Disables High Priority Rx Data Buffer Full Interrupt Status logging. 1: Enables High Priority Rx Data Buffer Full Interrupt Status logging.	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CMDQEE	High Priority Command Queue Empty Enable 0: Disables High Priority Command Queue Empty Interrupt Status logging. 1: Enables High Priority Command Queue Empty Interrupt Status logging.	R/W
4	RSPQFE	High Priority Response Queue Full Enable 0: Disables High Priority Response Queue Full Interrupt Status logging. 1: Enables High Priority Response Queue Full Interrupt Status logging.	R/W
5	TABTE	High Priority Transfer Abort Enable 0: Disables High Priority Transfer Abort Interrupt Status logging. 1: Enables High Priority Transfer Abort Interrupt Status logging.	R/W
8:6	—	These bits are read as 0. The write value should be 0.	R/W
9	TEE	High Priority Transfer Error Enable 0: Disables High Priority Transfer Error interrupt Stats logging. 1: Enables High Priority Transfer Error interrupt Stats logging.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

Note: This register supports I3C master mode and I3C secondary master mode.

TDBEE bit (High Priority Tx Data Buffer Empty Enable)

When TDBEE is 1, the operation of HTST.TDBEF is enabled.

For the setting conditions and clearing conditions of the HTST.TDBEF flag, see the details of HTST.TDBEF.

RDBFE bit (High Priority Rx Data Buffer Full Enable)

When RDBFE is 1, the operation of HTST.RDBFF is enabled.

For the setting conditions and clearing conditions of the HTST.RDBFF flag, see the details of HTST.RDBFF.

CMDQEE bit (High Priority Command Queue Empty Enable)

When CMDQEE is 1, the operation of HTST.CMDQEF is enabled.

For the setting conditions and clearing conditions of the HTST.CMDQEF flag, see the details of HTST.CMDQEF.

RSPQFE bit (High Priority Response Queue Full Enable)

When RSPQFE is 1, the operation of HTST.RSPQFF is enabled.

For the setting conditions and clearing conditions of the HTST.RSPQFF flag, see the details of HTST.RSPQFF.

TABTE bit (High Priority Transfer Abort Enable)

When TABTE is 1, the operation of HTST.TABTF is enabled.

For the setting conditions and clearing conditions of the HTST.TABTF flag, see the details of HTST.TABTF.

TEE bit (High Priority Transfer Error Enable)

When TEE is 1, the operation of HTST.TEF is enabled.

For the setting conditions and clearing conditions of the HTST.TEF flag, see the details of HTST.TEF.

33.2.56 HTIE : High Priority Transfer Interrupt Enable Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x208

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TEIE	—	—	—	TABTIE	RSPQFIE	CMDQEIE	—	RDBFIE	TDBEIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEIE	High Priority Tx Data Buffer Empty Interrupt Enable 0: Disables High Priority Tx Data Buffer Empty Interrupt Signal. 1: Enables High Priority Tx Data Buffer Empty Interrupt Signal.	R/W
1	RDBFIE	High Priority Rx Data Buffer Full Interrupt Enable 0: Disables High Priority Rx Data Buffer Full Interrupt Signal. 1: Enables High Priority Rx Data Buffer Full Interrupt Signal.	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CMDQEIE	High Priority Command Queue Empty Interrupt Enable 0: Disables High Priority Command Queue Empty Interrupt Signal. 1: Enables High Priority Command Queue Empty Interrupt Signal.	R/W

Bit	Symbol	Function	R/W
4	RSPQFIE	High Priority Response Queue Full Interrupt Enable 0: Disables High Priority Response Queue Full Interrupt Signal. 1: Enables High Priority Response Queue Full Interrupt Signal.	R/W
5	TABTIE	High Priority Transfer Abort Interrupt Enable 0: Disables High Priority Transfer Abort interrupt Signal. 1: Enables High Priority Transfer Abort interrupt Signal.	R/W
8:6	—	These bits are read as 0. The write value should be 0.	R/W
9	TEIE	High Priority Transfer Error Interrupt Enable 0: Disables High Priority Transfer Error Interrupt Signal. 1: Enables High Priority Transfer Error Interrupt Signal.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

The high priority interrupt signal enable register enables signaling of outstanding high priority interrupts received by I3C.

TDBEIE bit (High Priority Tx Data Buffer Empty Interrupt Enable)

TDBEIE is used to enable or disable the High Priority Tx Data buffer empty interrupt (I3C_HTX) requests when the HTST.TDBEF flag is set to 1.

RDBFIE bit (High Priority Rx Data Buffer Full Interrupt Enable)

RDBFIE is used to enable or disable the High Priority Rx Data buffer full interrupt (I3C_HRX) requests when the HTST.RDBFF flag is set to 1.

CMDQEIE bit (High Priority Command Queue Empty Interrupt Enable)

CMDQEIE is used to enable or disable the High Priority Command Queue empty interrupt (I3C_HCMD) requests when the HTST.CMDQEF flag is set to 1.

RSPQFIE bit (High Priority Response Queue Full Interrupt Enable)

RSPQFIE is used to enable or disable the High Priority Response Queue full interrupt (I3C_HRESP) requests when the HTST.RSPQFF flag is set to 1.

TABTIE bit (High Priority Transfer Abort Interrupt Enable)

TABTIE is used to enable or disable the High Priority Transfer Abort interrupt (I3C_EEI) requests when the HTST.TABTF flag is set to 1.

TEIE bit (High Priority Transfer Error Interrupt Enable)

TEIE is used to enable or disable the High Priority Transfer Error interrupt (I3C_EEI) requests when the HTST.TEF flag is set to 1.

33.2.57 HTSTFC : High Priority Transfer Status Force Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x20C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TEFC	—	—	—	TABTFC	RSPQFFC	CMDQEFC	—	RDBFFC	TDBEFC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEFC	High Priority Tx Data Buffer Empty Force 0: Not Force High Priority Tx Data Buffer Empty Interrupt for software testing. 1: Force High Priority Tx Data Buffer Empty Interrupt for software testing.	W
1	RDBFFC	High Priority Rx Data Buffer Full Force 0: Not Force High Priority Rx Data Buffer Full Interrupt for software testing. 1: Force High Priority Rx Data Buffer Full Interrupt for software testing.	W
2	—	This bit is read as 0.	R
3	CMDQEFC	High Priority Command Queue Empty Force 0: Not Force High Priority Command Queue Empty Interrupt for software testing. 1: Force High Priority Command Queue Empty Interrupt for software testing.	W
4	RSPQFFC	High Priority Response Queue Full Force 0: Not Force High Priority Response Queue Full Interrupt for software testing. 1: Force High Priority Response Queue Full Interrupt for software testing.	W
5	TABTFC	High Priority Transfer Abort Force 0: Not Force High Priority Transfer Abort Interrupt for software testing. 1: Force High Priority Transfer Abort Interrupt for software testing.	W
8:6	—	These bits are read as 0.	R
9	TEFC	High Priority Transfer Error Force 0: Not Force High Priority Transfer Error Interrupt for software testing. 1: Force High Priority Transfer Error Interrupt for software testing.	W
31:10	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

33.2.58 BCST : Bus Condition Status Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x210

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	BIDLF	BAVLF	BFREF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BFREF	Bus Free Detection Flag 0: Have not Detected Bus Free 1: Have Detected Bus Free	R
1	BAVLF	Bus Available Detection Flag*1 0: Have not Detected Bus Available 1: Have Detected Bus Available	R
2	BIDLF	Bus Idle Detection Flag*1 0: Have not Detected Bus Idle 1: Have Detected Bus Idle	R
31:3	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.

BFREF bit (Bus Free Detection Flag)

The Bus Free Condition is a period occurring after a STOP and before a START, and with the following duration:

- For Pure Bus: A duration of at least tCAS (see section 60, Electrical Characteristics)
- For Mixed Bus (at least one Legacy I^2C is present on the I3C Bus): A duration of at least tBUF (see section 60, Electrical Characteristics)

[Setting conditions]

- After a STOP condition is detected, when the number of cycles (I3Cφ) that are set by BFRECDT.FRECYC[8:0] has passed in the state of SCL = SDA = 1.
- After setting BCTL.BUSE to 1, when the number of cycles (I3Cφ) that are set by BFRECDT.FRECYC[8:0] has passed in the state of SCL = SDA = 1.

[Clearing conditions]

- When SCL and SDA are other than high.
- When the BCTL.BUSE bit is set to 0.

BAVLF bit (Bus Available Detection Flag)

The Bus Available Condition is a period during which the Bus Free Condition is sustained continuously for a duration of at least tAVAL (see section 60, Electrical Characteristics). A Slave can only issue a START Request (for an In-Band Interrupt, or for a Master Handoff Request) after a Bus Available Condition.

[Setting conditions]

- After a STOP condition is detected, when the number of cycles (I3Cφ) that are set by BAVLCDT.AVLCYC[8:0] has passed in the state of SCL = SDA = 1.
- After setting BCTL.BUSE to 1, when the number of cycles (I3Cφ) that are set by BAVLCDT.AVLCYC[8:0] has passed in the state of SCL = SDA = 1.

[Clearing conditions]

- When SCL and SDA are other than high.
- When the BCTL.BUSE bit is set to 0.

BIDLF bit (Bus Idle Detection Flag)

The Bus Idle Condition is a period during which the Bus Available Condition is sustained continuously for a duration of at least tIDLE (see section 60, Electrical Characteristics).

[Setting conditions]

- After a STOP condition is detected, when the number of cycles (I3Cφ) that are set by BIDLCDT.IDLCYC[17:0] has passed in the state of SCL = SDA = 1.
- After setting BCTL.BUSE to 1, when the number of cycles (I3Cφ) that are set by BIDLCDT.IDLCYC[17:0] has passed in the state of SCL = SDA = 1.

[Clearing conditions]

- When SCL and SDA are other than high.
- When the BCTL.BUSE bit is set to 0.

33.2.59 SVST : Slave Status Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x214

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SVAF[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	HOAF	—	—	—	—	—	—	—	—	DVIDF	HSMCF	—	—	—	—	GCAF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	GCAF	General Call Address Detection Flag 0: General call address does not detect. 1: General call address detects.	R/W*1
4:1	—	These bits are read as 0. The write value should be 0.	R/W
5	HSMCF	Hs-mode Master Code Detection Flag 0: Hs-mode Master Code does not detect. 1: Hs-mode Master Code detects.	R/W*1
6	DVIDF	Device-ID Address Detection Flag 0: Device-ID command does not detect. 1: Device-ID command detects. ● This bit set to 1 when the first frame received immediately after a START condition is detected matches a value of (device ID (1111 100) + 0[W]).	R/W*1
14:7	—	These bits are read as 0. The write value should be 0.	R/W
15	HOAF	Host Address Detection Flag 0: Host address does not detect. 1: Host address detects. ● This bit set to 1 when the received slave address matches the host address (0001 000).	R/W*1
18:16	SVAF[2:0]	Slave Address Detection Flag n (n = 0 to 2) 0: Slave n does not detect 1: Slave n detect	R/W*1
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I2C mode.
Note 1. Clearing (to 0) condition : Writing 0 after the 1 state is read.

GCAF flag (General Call Address Detection Flag)

I2C Normal Wake-Up Mode1 / 2 sets GCAF to 1 when switching from asynchronous operation to synchronous unit.

[Setting conditions]

- This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.GCAE bit = 1 (General call address detection is enabled).
  2. When the received slave address matches the general call address (0000 000 + 0 (write)).

[Clearing conditions]

- When 0 is written to the GCAF flag after reading GCAF flag to be 1.
- When a STOP condition is detected.
- When a Repeated START condition is detected.

HSMCF flag (Hs-mode Master Code Detection Flag)

The I2C Normal Wake-Up Mode 1/2 sets 1 to HSMCF when switching from asynchronous operation to synchronous unit.

[Setting conditions]

- This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.HSMCE bit = 1 (Hs-mode master code detection is enabled).
  2. When the first byte received immediately after a START condition is detected matches a value of Hs-mode master code (0000 1XXX) + 1 (NACK).

[Clearing conditions]

- When 0 is written to the HSMCF flag after reading HSMCF flag to be 1.
- When a STOP condition is detected.

DVIDF flag (Device-ID Address Detection Flag)

[Setting conditions]

- This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.DVIDE bit = 1 (Device-ID address detection is enabled).
  2. When the first byte received immediately after a START condition or Repeated START condition is detected matches a value of Device ID (1111 100) + 0 (write).

[Clearing conditions]

- When 0 is written to the DVIDF flag after reading DVIDF flag to be 1.
- When a STOP condition is detected.
- This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the first byte when the following 1 and 2 or 1 and 3 are satisfied.
  1. The SVCTL.DVIDE bit = 1 (Device-ID address detection is enabled).
  2. When the first byte received immediately after a START condition or Repeated START condition is detected does not match a value of Device ID (1111 100).
  3. When the first byte received immediately after a START condition or Repeated START condition is detected matches a value of (device ID (1111 100) + 0 [W]) and the second byte does not match any of slave addresses 0 to 2.

HOAF flag (Host Address Detection Flag)

I2C Normal Wake-Up Mode1 / 2 sets HOAF to 1 at the time of switching from asynchronous operation to synchronous unit.

[Setting conditions]

- This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.HOAE bit = 1 (Host address detection is enabled).
  2. When the received slave address matches the host address (0001 000).

[Clearing conditions]

- When 0 is written to the HOAF flag after reading HOAF flag to be 1.
- When a STOP condition is detected.
- When a Repeated START condition is detected.

SVAF[2:0] flags (Slave Address Detection Flag n (n = 0 to 2))

I2C Normal Wake-Up Mode1 / 2 sets 1 to SVAF[2:0] when switching from asynchronous operation to synchronous unit.

[Setting conditions]

For 7-bit address format: SVDVADn.SADLG bit = 0.

- This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.SVAEn bit = 1 (Slave n enabled).
  2. When the received slave address matches the SVDVADn.SVAD[6:0] bits value.

For 10-bit address format: SVDVADn.SADLG bit = 1.

- This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the second byte when all of the followings are satisfied.
  1. The SVCTL.SVAEn bit = 1 (Slave n enabled).
  2. When the received slave address matches a value of 11110 + SVDVADn.SVAD[9:8] bits and the following address matches the SVDVADn.SVAD[7:0] value.

[Clearing conditions]

- When 0 is written to the SVAF[2:0] flag after reading SVAF[2:0] flag to be 1.
- When a STOP condition is detected.

For 7-bit address format: SVDVADn.SADLG bit = 0.

- This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.SVAEn bit = 1 (Slave n enabled).
  2. When the received slave address does not match SVDVADn.SVAD[6:0] bits value.

For 10-bit address format: SVDVADn.SADLG bit = 1.

- This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.SVAEn bit = 1 (Slave n enabled).
  2. When the received slave address does not match a value of 11110 + SVDVADn.SVAD[9:8] bits.
- This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the second byte when all of the followings are satisfied.
  1. The SVCTL.SVAEn bit = 1 (Slave n enabled).
  2. When the received slave address matches a value of 11110 + SVDVADn.SVAD[9:8] bits and the following address does not match the SVDVADn.SVAD[7:0] value.

33.2.60 WUST : Wake Up Unit Operating Status Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x218

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WUASYNF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WUASYNF	Wake-up function asynchronous operation status flag 0: I3C synchronous circuit enable condition. 1: I3C asynchronous circuit enable condition.	R
31:1	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3

WUASYNF flag (Wake-up function asynchronous operation status flag)

This bit shows whether I3C is in the TCLK asynchronous operation (WUCTL.WUFE bit = 1).

[Setting condition]

- All of the followings are satisfied.
  1. The WUCTL.WUFE bit = 1 (Wake-up function is enabled)
  2. When the BCST.BFREF flag = 1 after 0 is written to the WUCTL.WUFSYNE bit

[Clearing condition : I^2C slave]

- The WUCTL.WUFE bit = 0 (Wake-up function is disabled)
- All of the followings are satisfied.
  1. The WUCTL.WUFE bit = 1 (Wake-up function is enabled)
  2. Wake-up event is detected
  3. When 1 is written to the WUCTL.WUFSYNE bit during WUASYNF flag = 1

[Clearing condition : I3C slave]

- The WUCTL.WUFE bit = 0 (Wake-up function is disabled)
- All of the followings are satisfied.
  1. The WUCTL.WUFE bit = 1 (Wake-up function is enabled)
  2. Wake-up event is detected
  3. When 1 is written to the WUCTL.WUFSYNE bit during WUASYNF flag = 1
  4. When a STOP condition is detected.

[Clearing condition : I^2C/I3C slave]

- All of the followings are satisfied.
  1. The WUCTL.WUFE bit = 1 (Wake-up function is enabled)
  2. Wake-Up event is not detected
  3. The WUASYNF flag = 1
  4. The WUCTL.WUFSYNE bit = 1
  5. When a STOP condition is detected.

[Clearing condition : I3C master]

- The WUCTL.WUFE bit = 0 (Wake-up function is disabled)
- All of the followings are satisfied.
  1. The WUCTL.WUFE bit = 1 (Wake-up function is enabled)
  2. Wake-up event is detected.
  3. The WUASYNF flag = 1
  4. The WUCTL.WUFSYNE bit = 1

33.2.61 MRCCPT : MsyncCNT Counter Capture Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x21C

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	MSyncCNT Counter Capture Used in Async Mode 1, not used in Async Mode0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

MRCCPT[31:0] Bits

- Async Mode 1 (Asynchronous Advanced Mode)
  When ATCCNTE.ATCE is enabled, it starts counting. It captures as MSyncCNT for each aME (SDA falling edge of START condition), and store it in the capture register.

33.2.62 DATBASm : Device Address Table Basic Register m (m = 0 to 7)

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x224 + 0x08 × m

Bit position:	31	30	29	28	27	26	25	24	23							16
Bit field:	DVTYP	DVNACK[1:0]		—	—	—	—	—	DVDYAD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6						0
Bit field:	DVIBITS	DVMRRJ	DVSIRRJ	DVIBIPL	—	—	—	—	—	DVSTAD[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	DVSTAD[6:0]	Device Static Address I3C/I^2C Static Address	R/W
11:7	—	These bits are read as 0. The write value should be 0.	R/W
12	DVIBIPL	Device IBI Payload 0: IBIs from this Device do not carry a Data Payload. 1: IBIs from this Device do carry a Data Payload.	R/W
13	DVSIRRJ	Device In-Band Slave Interrupt Request Reject 0: This Device shall ACK the SIR. 1: This Device shall NACK the SIR and send the auto-disable CCC.	R/W
14	DVMRRJ	Device In-Band Master Request Reject 0: This Device shall ACK Master Requests. 1: This Device shall NACK Master Requests and send the auto-disable command.	R/W
15	DVIBITS	Device IBI Time-stamp 0: The Master shall not time-stamp IBIs from this Device with Master Time-stamps. 1: The Master shall time-stamp IBIs for this Device with Master Time-stamps.	R/W
23:16	DVDYAD[7:0]	Device I3C Dynamic Address Bit 23 is the parity bit, per the I3C specification, computed and updated by the software driver.	R/W

Bit	Symbol	Function	R/W
28:24	—	These bits are read as 0. The write value should be 0.	R/W
30:29	DVNACK[1:0]	Device NACK Retry Count Device-specific retry count	R/W
31	DVTYP	Device Type 0: I3C Device 1: I^2C Device	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

DVIBIPL bit (Device IBI Payload)

Indicates whether IBIs from this Device have a Data Payload. This field reflects the IBI Payload bit in the Device's Bus Characteristics Register (BCR).

During IBI handling for this Device, the Master shall use this field to determine whether or not to drive reception of the IBI Data Payload. Data continuation is indicated by the T-Bit.

DVSIRRJ bit (Device In-Band Slave Interrupt Request Reject)

Controls whether this Device, when operating in the Master role, will accept vs. reject Slave Interrupt Requests from other Devices.

DVMRRJ bit (Device In-Band Master Request Reject)

Controls whether this Device, when operating in the Master role, will accept vs. reject Master requests from other Devices.

This bit is only valid if I3C declares Non-Current Master Capability.

DVIBITS bit (Device IBI Time-stamp)

Enables or disables IBI time-stamping for a specific Device.

Note: The IBI Status Descriptor for each IBI event indicates whether or not the individual IBI event was actually time-stamped. Set to 0 except for Async mode 0 and Async mode 1 of timing control.

DVNACK[1:0] bits (Device NACK Retry Count)

These bits set the number of retries when a NACK response is received from the slave for the transaction set in the Command Descriptor.

Note: When ENTDAA is executed by Address Assign Command, the setting of this bit is ignored and the transaction ends when NACK is received once.

Note: I3C will retry according to the setting of DVNACK[1:0] bit, even if it receives a NACK for the broadcast address.

Note: If DVNACK[1:0] bits are 0x0, I3C will not retry even for Direct CCCs.

33.2.63 EXDATBAS : Extended Device Address Table Basic Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x2A0

Bit position:	31	30	29	28	27	26	25	24	23							16
Bit field:	EDTYP	EDNACK[1:0]		—	—	—	—	—	EDDYAD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6						0
Bit field:	—	—	—	—	—	—	—	—	—	EDSTAD[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	EDSTAD[6:0]	Extended Device Static Address I3C/I2C static address	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W
23:16	EDDYAD[7:0]	Extended Device I3C Dynamic Address Bit 23 is the parity bit, per the I3C specification, computed and updated by the software driver.	R/W
28:24	—	These bits are read as 0. The write value should be 0.	R/W
30:29	EDNACK[1:0]	Extended Device NACK Retry Count Device-specific retry count	R/W
31	EDTYP	Extended Device Type 0: I3C Device 1: I2C Device	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

33.2.64 SDATBASn : Slave Device Address Table Basic Register n (n = 0 to 2)

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x2B0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22						16
Bit field:	—	—	—	—	—	—	—	—	—	SDDYAD[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9									0
Bit field:	—	—	—	SDIBIPL	—	SDADLS	SDSTAD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	SDSTAD[9:0]	Slave Device Static Address*2 I3C Static Address	R/W
10	SDADLS	Slave Device Address Length Selection*3 0: Slave device address length 7 bits selected. 1: Slave device address length 10 bits selected. (I2C device only)	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	SDIBIPL*1	Slave Device IBI Payload*4 This bit is the mirror bit of the SVDCT.TBCR2. 0: IBIs from this device do not carry a data payload. 1: IBIs from this device carry a data payload.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
22:16	SDDYAD[6:0]*1	Slave Device I3C Dynamic Address*5	R/W
31:23	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: SW write to the SDATBAS register of the main master is prohibited.
Note 1. This bit is valid only in SDATBAS0 register.
Note 2. These bits support I2C, I3C secondary master, and I3C slave mode.
Note 3. This bit supports I2C mode.
Note 4. This bit supports I3C secondary master mode and I3C slave mode.
Note 5. These bits support I3C secondary master mode and I3C slave mode.

SDSTAD[9:0] bits (Slave Device Static Address)

When the 7-bit address format is selected (SDADLS bit is 0), the lower 7 bits of SDSTAD[9:0] function as the 7-bit address.

When the 10-bit address format is selected (SDADLS bit is 1), the SDSTAD[9:0] function as the 10-bit address. While the SVCTL.SVAEn bit is 0, the setting of this bit is ignored.

SDIBIPL bit (Slave Device IBI Payload)

Indicates whether IBIs from this Device have a Data Payload. This field reflects the IBI Payload bit in the Device's Bus Characteristics Register (BCR).

During IBI handling for this Device, the Master shall use this field to determine whether or not to drive reception of the IBI Data Payload. Data continuation is indicated by the T-Bit.

SDDYAD[6:0] bits (Slave Device I3C Dynamic Address)

[Update conditions]

- When writing Dynamic Address value.
- When Slave Address value is its own Static Address in receiving SETDASA CCC (Direct), these bits are updated to Dynamic Address value.*1
- When Dynamic Address Assignment procedure that starts by receiving ENTDAA CCC (Broadcast) is established.*1
- When receiving RSTDAA CCC (Broadcast), all bits are cleared to 0. *1
- When Slave Address value is its own Dynamic Address in receiving RSTDAA CCC (Direct), all bits are cleared to 0. *1
- When Slave Address value is its own Dynamic Address in receiving SETNEWDA CCC (Direct), these bits are updated to the Dynamic Address value. *1
- When receiving SETAASA CCC (Broadcast), these bits are updated to the value of SDSTAD[6:0] bits*2.

Note 1. See the MIPI I3C Specification v1.0.
Note 2. See the MIPI I3C Basic Specification v1.0.

33.2.65 MSDCTm : Master Device Characteristic Table Register m (m = 0 to 7)

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x2D0 + 0x04 × m

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RBCR76[1:0]		RBCR5	RBCR4	RBCR3	RBCR2	RBCR1	RBCR0	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	RBCR0	Max Data Speed Limitation*1 0: No Limitation 1: Limitation	R/W
9	RBCR1	IBI Request Capable 0: Not Capable 1: Capable	R/W

Bit	Symbol	Function	R/W
10	RBCR2	IBI Payload 0: No data byte follows the accepted IBI. 1: Mandatory one or more data bytes follow the accepted IBI. Data byte continuation is indicated by T-Bit.	R/W
11	RBCR3	Offline Capable*2 0: Device will always respond to I3C bus commands. 1: Device will not always respond to I3C bus commands.	R/W
12	RBCR4	Bridge Identifier*3 0: Not a Bridge Device 1: A Bridge Device	R/W
13	RBCR5	SDR Only / SDR and HDR Capable 0: SDR only 1: HDR Capable	R/W
15:14	RBCR76[1:0]	Device Role 0 0: I3C Slave 0 1: I3C Master*4 Others: Setting prohibited	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.
Note 1. Master shall use the GETMXDS CCC to interrogate the Slave for specific limitation.
Note 2. Offline Capable Devices retain the Dynamic Address.
Note 3. Bridge Devices are required to comply with this MIPI Specification for I3C.
Note 4. For an I3C Device acting as I3C Main Master, the BCR Device Role bits will contain the value 01.

The DCT table captures the Device characteristics (PID, BCR, DCR) and assigned Dynamic Address of each Device on the I3C Bus that participates in the Dynamic Address Allocation (ENTDAA) procedure.

RBCRn bits (Received Bus Characteristic Register)

Each I3C Device that is connected to the I3C Bus shall have an associated read-only Bus Characteristics Register (BCR). This read-only register describes the I3C compliant Device's role and capabilities for use in Dynamic Address assignment and Common Command Codes.

Note: When RBCR2 is 0 and when ACK response to Slave Interrupt Request from I3C Slave by DATBASm.DVSIRRJ = 0 (m = 0 to 7), STOP Condition is issued after ACK response. When RBCR2 is 1 and when ACK response to Slave Interrupt Request from I3C Slave by DATBASm.DVSIRRJ = 0 (m = 0 to 7), IBI Payload is received after ACK response. STOP Condition is issued after end of IBI Payload.

[Update condition]

- When receiving of Bus Characteristics Register (BCR) from Device in the Dynamic Address Assignment procedure starting by receiving ENTDAA CCC (Broadcast).*1

Note 1. See the MIPI I3C Specification v1.0

33.2.66 SVDCT : Slave Device Characteristic Table Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x320

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	0
Bit field:	TBCR76[1:0]		TBCR5	TBCR4	TBCR3	TBCR2	TBCR1	TBCR0	TDCR[7:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TDCR[7:0]	Transfer Device Characteristic Register 255 available codes for describing the type of sensor, or device. Examples: Accelerometer, gyroscope, composite devices Default value is 0x0: Generic Device	R/W
8	TBCR0	Max Data Speed Limitation*1 0: No Limitation 1: Limitation	R/W
9	TBCR1	IBI Request Capable 0: Not Capable 1: Capable	R/W
10	TBCR2	IBI Payload 0: No data byte follows the accepted IBI. 1: Mandatory one or more data bytes follow the accepted IBI. Data byte continuation is indicated by T-Bit.	R/W
11	TBCR3	Offline Capable*2 0: Device will always respond to I3C bus commands. 1: Device will not always respond to I3C bus commands.	R/W
12	TBCR4	Bridge Identifier*3 0: Not a Bridge Device 1: A Bridge Device	R/W
13	TBCR5	SDR Only / SDR and HDR Capable 0: SDR only 1: HDR capable	R/W
15:14	TBCR76[1:0]	Device Role 0 0: I3C Slave 0 1: I3C Master*4 Others: Setting prohibited	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.
Note 1. Master shall use the GETMXDS CCC to interrogate the Slave for specific limitation.
Note 2. Offline Capable Devices retain the Dynamic Address.
Note 3. Bridge Devices are required to comply with this MIPI Specification for I3C.
Note 4. For an I3C Device acting as I3C Main Master, the BCR Device Role bits will contain the value 01b.

The DCT table captures the Device characteristics (PID, BCR, DCR) and assigned Dynamic Address of each device on the I3C bus that participates in the Dynamic Address Allocation (ENTDAA) procedure.

TDCR[7:0] bits (Transfer Device Characteristic Register)

Each I3C device that is connected to the I3C bus has an associated Device Characteristics Register (DCR). This register describes the I3C compliant device type such as accelerometer and gyroscope, for use in Dynamic Address assignment and Common Command Codes.

TBCRn bits (Transfer Bus Characteristic Register)

Each I3C device that is connected to the I3C bus has an associated Bus Characteristics Register (BCR). This register describes the role and capabilities of the I3C compliant device for use in Dynamic Address assignment and Common Command Codes.

When I3C Slave issues IBI by Command Descriptor, the condition of TBCRn is described as follows:

[Slave Interrupt Request : No IBI Payload follow the accepted IBI]

- TBCR1 = 1
- TBCR2 = 0

Note: Set DATA_LENGTH[15:0] of Command Descriptor to 0.

[Slave Interrupt Request : IBI Payload follow the accepted IBI]

- TBCR1 = 1
- TBCR2 = 1

Note: Set DATA_LENGTH[15:0] of Command Descriptor to any value.

[Mastership Request]

- TBCR1 = 1
- TBCR76[1:0] = 01b

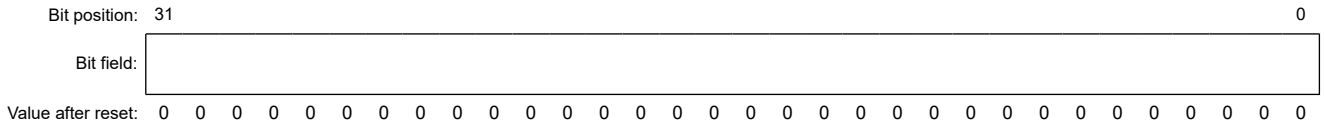
When I3C Slave receives CCC from I3C Master, it performs the following operations according to the setting of TBCRn:

- When TBCR2 = 1, CMRLG.IBIPSZ[7:0] is sent as the 3rd byte data to GETMRL CCC from I3C Master
- When TBCR0 = 0, NACK responses to GETMXDS CCC from I3C Master
- When TBCR0 = 1, ACK responses to GETMXDS CCC from I3C Master and sends data from CMDSPW, CMDSPR, and CMDSPT registers

33.2.67 SDCTPIDL : Slave Device Characteristic Table Provisional ID Low Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x324



Bit	Symbol	Function	R/W
31:0	n/a	Transfer Device Provisional ID Low Bits 31 to 16 are read as 0. Bits 15 to 0 are bits [15:0] of device's I3C PID.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

33.2.68 SDCTPIDH : Slave Device Characteristic Table Provisional ID High Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x328

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Transfer Device Provisional ID High Bits [47:16] of device's I3C PID.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

33.2.69 SVDVADn : Slave Device Address Register n (n = 0 to 2)

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x330 + 0x04 × n

Bit position:	31	30	29	28	27	26	25									16
Bit field:	SDYADV	SSTADV	—	—	SADLG	—	SVAD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0.	R
25:16	SVAD[9:0]	Slave Address*1 A slave address is set. When rewriting SVAD, change to SVAE = 0 and rewrite.	R
26	—	This bit is read as 0.	R
27	SADLG	Slave Address Length*2 0: The 7-bit address format is selected. 1: The 10-bit address format is selected.	R
29:28	—	These bits are read as 0.	R
30	SSTADV	Slave Static Address Valid*1 0: Slave address is disabled. 1: Slave address is enabled.	R
31	SDYADV*4	Slave Dynamic Address Valid*3 0: Dynamic Address is disabled. 1: Dynamic Address is enabled.	R

Note: S-TYPE3, P-TYPE3
Note 1. These bits support I^2C, I3C secondary master, and I3C slave mode.
Note 2. This bit supports I^2C mode.
Note 3. This bit supports I3C secondary master mode and I3C slave mode.
Note 4. This bit is valid only in SVDVAD0 register.

SVAD[9:0] bits (Slave Address)

The SVAD[9:0] bits indicate a valid slave address.

[The SVDVAD0.SDYADV bit = 1]

Note: This condition is only for SVDVAD0.SVAD[9:0].

- The SVAD[9:7] bits = 0
- The SVAD[6:0] bits = the SDATBAS0.SDDYAD[6:0] bits

[The SVDVADn.SSTADV bit = 1 and the SVDVADn.SADLG bit = 0]

- The SVAD[9:7] bits = 0
- The SVAD[6:0] bits = the SDATBASn.SDSTAD[6:0] bits

[The SVDVADn.SSTADV bit = 1 and the SVDVADn.SADLG bit = 1]

- The SVAD[9:0] bits = the SDATBASn .SDSTAD[9:0] bits

SADLG bit (Slave Address Length)

[Setting conditions]

- All of the followings are satisfied:

1. The PRTS.PRTMD bit = 1 (I^2C Protocol mode)
2. The SVCTL.SVAEn bit = 1 (Slave n is enabled)
3. The SDATBASn .SDADLS bit = 1 (The address length is 10 bits)

[Clearing condition]

- [Setting condition] is not satisfied.

SSTADV bit (Slave Static Address Valid)

[Setting conditions]

- All of the followings are satisfied:

1. The SVCTL.SVAEn bit = 1 (Slave n is enabled)
2. The SVDVAD0.SDYADV bit = 0 (Dynamic Address is disabled)

   Note: This condition is only for SVDVAD0.SSTADV.
3. If the SVDVADn.SADLG bit = 0, the SDATBASn .SDSTAD[6:0] bits are not all 0
   If the SVDVADn.SADLG bit = 1, the SDATBASn .SDSTAD[9:0] bits are not all 0

[Clearing condition]

- [Setting condition] is not satisfied.

SDYADV bit (Slave Dynamic Address Valid)

[Setting conditions]

- All of the followings are satisfied:

1. The PRTS.PRTMD bit = 0 (I3C Protocol mode)
2. The SVCTL.SVAEn bit = 1 (Slave n is enabled)
3. The SDATBAS0.SDDYAD[6:0] bits are not all 0

Note: This condition is only for SVDVAD0.SDYADV.

[Clearing condition]

- [Setting condition] is not satisfied.

33.2.70 CSECMD : CCC Slave Events Command Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x350

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSRQE	SVIRQE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SVIRQE	Slave Interrupt Requests Enable 0: DISABLED: Slave-initiated Interrupts is Disabled by the Master to control. 1: ENABLED: Slave-initiated Interrupts is Enabled by the Master to control.	R/W
1	MSRQE	Mastership Requests Enable 0: DISABLED: Mastership requests from Secondary Masters is Disabled by the Current Master to control. 1: ENABLED: Mastership requests from Secondary Masters is Enabled by the Current Master to control.	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

SVIRQE bit (Slave Interrupt Requests Enable)

This bit allows the Master to control when Slave-initiated Interrupts are allowed on the I3C Bus.

These four Direct (ENEC/DISEC Format 1) or Broadcast (ENEC/DISEC Format 2) CCCs allows the Master to control when Slave- initiated traffic is (Enable) vs. is not (Disable) allowed on the I3C Bus. This control governs a Slave's attempts to request an Interrupt (ENI), or to request Mastership (ENMR).

[Setting conditions]

- When writing 1
- When receiving ENEC CCC (Broadcast) with ENINT bit = 1.*1
- When ENINT bit = 1 with own Slave Address in receiving ENEC CCC (Direct).*1

[Clearing conditions]

- When writing 0.
- When receiving DISEC CCC (Broadcast) with DISINT bit = 1.*1
- When DISINT bit = 1 with own Slave Address in receiving DISEC CCC (Direct).*1

MSRQE bit (Mastership Requests Enable)

This bit allows the Current Master to control when Mastership requests from Secondary Masters are allowed on the I3C Bus.

[Setting conditions]

- When writing 1.
- When receiving ENEC CCC (Broadcast) with ENMR bit = 1.*1
- When ENMR bit = 1 with own Slave Address in receiving ENEC CCC (Direct).*1

[Clearing conditions]

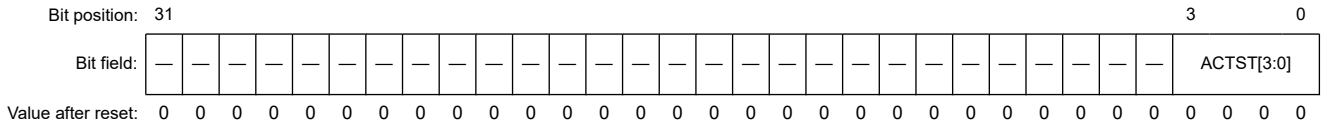
- When writing 0.
- When receiving DISEC CCC (Broadcast) with DISMR bit = 1.*1
- When DISMR bit = 1 with own Slave Address in receiving DISEC CCC (Direct).*1

Note 1. See the MIPI I3C Specification v1.0

33.2.71 CEACTST : CCC Enter Activity State Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x354



Bit	Symbol	Function	R/W
3:0	ACTST[3:0]	Activity State 0x1: ENTAS0 (1µs: Latency-free operation) 0x2: ENTAS1 (100 µs) 0x4: ENTAS2 (2 ms) 0x8: ENTAS3 (50 ms: Lowest-activity operation) Others: Setting prohibited	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

ACTST[3:0] bits (Activity State)

[Update conditions]

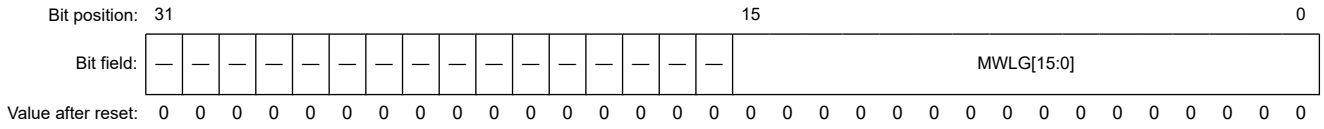
- When writing Activity State value.
- When receiving ENTAS0 CCC (Broadcast), these bits are updated to 0x1.*1
- When receiving ENTAS1 CCC (Broadcast), these bits are updated to 0x2.*1
- When receiving ENTAS2 CCC (Broadcast), these bits are updated to 0x4.*1
- When receiving ENTAS3 CCC (Broadcast), these bits are updated to 0x8.*1
- When Slave Address value is its own Slave Addres in receiving ENTAS0 CCC (Direct), these bits are updated to 0x1.*1
- When Slave Address value is its own Slave Addres in receiving ENTAS1 CCC (Direct), these bits are updated to 0x2.*1
- When Slave Address value is its own Slave Addres in receiving ENTAS2 CCC (Direct), these bits are updated to 0x4.*1
- When Slave Address value is its own Slave Addres in receiving ENTAS3 CCC (Direct), these bits are updated to 0x8.*1

Note 1. See the MIPI I3C Specification v1.0.

33.2.72 CMWLG : CCC Max Write Length Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x358



Bit	Symbol	Function	R/W
15:0	MWLG[15:0]	Max Write Length	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

MWLG[15:0] bits (Max Write Length)

These bits use for the I3C Master to set or get a maximum data write length in bytes for one Slave Device.

This Max Write Length does not affect data write lengths for Broadcast CCCs. The Set/Get Max Write Length value is transmitted over two bytes, with the most significant byte (MSB) transmitted first. The minimum value that Max Write Length can be set to is 8.

[Update conditions]

- When writing Max Write Length value.
- When receiving SETMWL CCC (Broadcast), these bits are updated to MWL value.*1
- When Slave Address value is its own Slave Address in receiving SETMWL CCC (Direct), these bits are updated to MWL value.*1

Note 1. See the MIPI I3C Specification v1.0

33.2.73 CMRLG : CCC Max Read Length Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x35C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	IBIPSZ[7:0]								MRLG[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	MRLG[15:0]	Max Read Length	R/W
23:16	IBIPSZ[7:0]	IBI Payload Size	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

MRLG[15:0] bits (Max Read Length)

These bits use for the I3C Master to set or get a maximum data read length for one Slave Device.

The Set/Get Max Read Length value is transmitted over the first two bytes, with most significant byte (MSB) transmitted first. The minimum value to which Max Read Length can be set is 16.

[Update conditions]

- When writing Max Read Length value.
- When receiving SETMRL CCC (Broadcast), these bits are updated to MRL value.*1
- When Slave Address value is its own Slave Address in receiving SETMRL CCC (Direct), these bits are updated to MRL value.*1

IBIPSZ[7:0] bits (IBI Payload Size)

These bits use for the I3C Master to set or get optionally a maximum IBI payload size.

For devices with BCR bit 2 set to 1, the Max IBI payload size value is added as a third-byte, where a value of 0 indicates an unlimited payload size. If Timing Control is used, then the minimum IBI payload size is either four bytes or five bytes. If Timing Control is not used, then the minimum IBI payload size is 1 (one byte).

This CCC is optional for the Slave, with two exceptions:

1. This CCC is required if both (a) any private Read Request Message (s) and/or any extended Read Request CCC (s) implemented by the Slave support a variable limit on the maximum number of data bytes that the Slave may return per Message, and (b) this limit is greater than 16 bytes.
2. This CCC is required if the Slave both (a) supports an IBI Payload (as indicated with BCR bit 1), and (b) will transmit more than one byte of private payload (not counting Timing Control bytes, when Timing Control used).

[Update conditions]

- When writing Max IBI payload size value.
- When receiving SETMRL CCC (Broadcast), these bits are updated to IBI payload size value.*1
- When Slave Address value is its own Slave Address in receiving SETMRL CCC (Direct), these bits are updated to IBI payload size value.*1

Note 1. See the MIPI I3C Specification v1.0.

33.2.74 CETSTMD : CCC Enter Test Mode Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x360

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSTMD[7:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TSTMD[7:0]	Test Mode 0x00: Exit Test Mode This value removes all I3C devices from Test Mode. 0x01: Vendor Test Mode This value indicates that I3C devices shall return a random 32bit value in the provisional ID during the Dynamic Address Assignment procedure. Others: Setting prohibited	R
31:8	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

TSTMD[7:0] bits (Test Mode)

When these bits set to 0x00, all I3C Devices remove from Test Mode.

When these bits set to 0x01, I3C Devices shall return a random 32bit value in the Provisional ID during the Dynamic Address Assignment procedure.

The Broadcast CCC informs all I3C Devices that the Master is entering a specified Test Mode during manufacturing or Device test. The Enter Test Mode command Frame format includes a byte that specifies which Test Mode to enter. Supporting I3C Devices shall enter the indicated Test Mode upon receipt of the Enter Test Mode CCC.

[Update condition]

- When receiving ENTTM CCC (Broadcast), these bits are updated to Test Mode Byte value.*1

Note 1. See the MIPI I3C Specification v1.0.

33.2.75 CGDVST : CCC Get Device Status Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x364

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	8	7	6	5	4	3	0
Bit field:	VDRSV[7:0]		ACTMD[1:0]		PRTE	—	PNDINT[3:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	PNDINT[3:0]	Pending Interrupt Contains the interrupt number of any pending interrupt, or 0 if no interrupts are pending. This encoding allows for up to 15 numbered interrupts. If more than one interrupt is set, then the highest priority interrupt shall be returned.	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	PRTE	Protocol Error 0: The Slave has not detected a protocol error since the last Status read. 1: The Slave has detected a protocol error since the last Status read.	R/W
7:6	ACTMD[1:0]	Slave Device's current Activity Mode 0 0: Activity Mode 0 0 1: Activity Mode 1 1 0: Activity Mode 2 1 1: Activity Mode 3	R/W
15:8	VDRSV[7:0]	Vendor Reserved Reserved for vendor-specific meaning	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

PRTE bit (Protocol Error)

If this bit set to 1, then the Slave detects a protocol error since the last Status read.

The Slave checks for such errors. Note that this value self-clears by the hardware upon every successful completion of a Master read of the Slave's Status.

The Direct CCC is a Get request for one I3C Slave Device to return its current Status, in the two-byte format detailed. Note that byte 0 is the LSB, and byte 1 is the MSB.

[Setting condition]

- When the Slave detected a protocol error.*1

[Clearing condition]

- When transmission by own Slave Address is completed without error after receiving GETSTATUS CCC (Direct).*1

ACTMD[1:0] bits (Slave Device's current Activity Mode)

Contains the two-bit ID of the Slave Device's current Activity Mode (readiness to support data read of sensor or related information).

Note 1. See the MIPI I3C Specification v1.0.

33.2.76 CMDSPW : CCC Max Data Speed W (Write) Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x368

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	MSWDR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	MSWDR[2:0]	Maximum Sustained Write Data Rate 0 0 0: fscl Max (default value) 0 0 1: 8 MHz 0 1 0: 6 MHz 0 1 1: 4 MHz 1 0 0: 2 MHz Others: Setting prohibited	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

33.2.77 CMDSPR : CCC Max Data Speed R (Read) Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x36C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5		3	2		0
Bit field:	—	—	—	—	—	—	—	—	—	—	CDTTIM[2:0]			MSRDR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	MSRDR[2:0]	Maximum Sustained Read Data Rate 0 0 0: fscl Max (default value) 0 0 1: 8 MHz 0 1 0: 6 MHz 0 1 1: 4 MHz 1 0 0: 2 MHz Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
5:3	CDTTIM[2:0]	Clock to Data Turnaround Time (TSCO) 0 0 0: 8 ns or less (default value) 0 0 1: 9 ns or less 0 1 0: 10 ns or less 0 1 1: 11 ns or less 1 0 0: 12 ns or less 1 1 1: TSCO is more than 12 ns, and is reported by private agreement. Others Setting prohibited	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

33.2.78 CMDSPT : CCC Max Data Speed T (Turnaround) Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x370

Bit position:	31								23																				0
Bit field:	MRTE	—	—	—	—	—	—	—	MRTTIM[23:0]																				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
23:0	MRTTIM[23:0]	Maximum Read Turnaround Time 24-bit field can encode turnaround times from 0.0 seconds to 16 seconds. 0x000000: 0 µs (minimum value) 0x000001: 1 µs (resolution) ⋮ 0xF42400: 16 seconds (maximum value) Others: Setting prohibited	R/W
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	MRTE	Maximum Read Turnaround Time Enable 0: Disables transmission of the Maximum Read Turnaround Time. (GETMXDS Format 1: Without Turnaround) 1: Enables transmission of the Maximum Read Turnaround Time. (GETMXDS Format 2: With Turnaround)	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

33.2.79 CETSM : CCC Exchange Timing Support Information M (Mode) Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x374

Bit position:	31	30	29	28	27	26	25	24	23							16	15							8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	INAC[7:0]								FREQ[7:0]								—	—	—	—	—	SPTASYN1	SPTASYN0	SPTSYN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPTSYN	Supports Sync Mode 0: Sync Mode is not supported. 1: Sync Mode is supported.	R/W
1	SPTASYN0	Support Async Mode 0 0: Async Mode 0 is not supported. 1: Async Mode 0 is supported.	R/W
2	SPTASYN1	Support Async Mode 1 0: Async Mode 1 is not supported. 1: Async Mode 1 is supported.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
15:8	FREQ[7:0]	Frequency Byte This byte represents the Slave's internal oscillator frequency in increments of 0.5 MHz (500 kHz), up to 127.5 MHz. 0x00: 32.0 KHz 0x01: 0.5 MHz 0x02: 1.0 MHz ⋮ 0xFD: 126.5 MHz 0xFE: 127.0 MHz 0xFF: 127.5 MHz	R/W
23:16	INAC[7:0]	Inaccuracy Byte This byte represents the maximum variation of the Slave's internal oscillator in 1/10th percent (0.1%) increments, up to 25.5%. 0x00: 0.0% 0x01: 0.1% 0x02: 0.2% ⋮ 0xFD: 25.3% 0xFE: 25.4% 0xFF: 25.5%	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

SPTSYN bit (Supports Sync Mode)

Bit mask indicating which Supports Sync Mode of Timing Control Mode (s) the target Slave supports.

If this bit set (has value 1), then that Slave supports the corresponding Supports Sync Mode of Timing Control Mode.

SPTASYN0 bit (Support Async Mode 0)

Bit mask indicating which Supports Async Mode 0 of Timing Control Mode (s) the target Slave supports.

If this bit set (has value 1), then that Slave supports the corresponding Supports Async Mode 0 of Timing Control Mode.

SPTASYN1 bit (Support Async Mode 1)

Bit mask indicating which Supports Async Mode 1 of Timing Control Mode (s) the target Slave supports.

If this bit set (has value 1), then that Slave supports the corresponding Supports Async Mode 1 of Timing Control Mode.

The Directed CCC provides the framework for the Master to query the Exchange Timing capabilities supported by the I3C Slaves. The Get Exchange Timing Support Information CCC causes the addressed Slave to return four data bytes containing key information on supported Timing Control modes, current state, and internal oscillator/clock frequency and inaccuracy.

33.2.80 CETSS : CCC Exchange Timing Support Information S (State) Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x378

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	ICOVF	—	—	—	—	ASYNE[1:0]		SYNE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SYNE	Sync Mode Enabled 0: Sync Mode Disabled 1: Sync Mode Enabled	R/W
2:1	ASYNE[1:0]	Async Mode Enabled Async Mode 3, 2 are unsupported and set to 0. 0 0: All Mode Disable 0 1: Async Mode 0 Enabled 1 0: Async Mode 1 Enabled Others: Setting prohibited	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	ICOVF	Internal Counter Overflow 0: Slave has not experienced a counter overflow since the most recent previous check. 1: Slave experienced a counter overflow since the most recent previous check.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

Bit mask indicating which Timing Control Mode (if any) is currently enabled for the target Slave, and whether any counter overflows have occurred since the most recent previous check. If a Timing Control Mode bit is set (has value 1), then that Slave has currently enabled the corresponding Timing Control Mode. If the Overflow bit is set (has value 1), then that Slave experienced a counter overflow since the most recent previous check.

ASYNE[0] Bit (Async Mode 0 Enabled)

Slave Timing Control Async Mode 0 is enabled.

[Setting condition]

- When writing 1.
- When CETSM.SPTASYN[0] bit = 1 and either of the following 1 or 2 are satisfied.

1. When receiving SETXTIME CCC (Broadcast) with Defining byte value 0xDF.
2. When Slave Address value is its own Slave Address in receiving SETXTIME CCC (Direct) with Defining byte value 0xDF.

[Clearing condition]

- When writing 0.
- When CETSM.SPTASYN[0] bit = 1 and either of the following 1 or 2 are satisfied.

1. When receiving SETXTIME CCC (Broadcast) with Defining byte value 0xEF.

2. When Slave Address value is its own Slave Address in receiving SETXTIME CCC (Direct) with Defining byte value 0xEF.

ASYNE[1] Bit (Async Mode 1 Enabled)

Slave Timing Control Async Mode 1 is enabled.

[Setting condition]

- When writing 1.
- When CETSM.SPTASYN[1] bit = 1 and either of the following 1 or 2 are satisfied.

1. When receiving SETXTIME CCC (Broadcast) with Defining byte value 0xEF.
2. When Slave Address value is its own Slave Address in receiving SETXTIME CCC (Direct) with Defining byte value 0xEF.

[Clearing condition]

- When writing 0.
- When CETSM.SPTASYN[1] bit = 1 and either of the following 1 or 2 are satisfied.

1. When receiving SETXTIME CCC (Broadcast) with Defining byte value 0xDF.
2. When Slave Address value is its own Slave Address in receiving SETXTIME CCC (Direct) with Defining byte value 0xDF.

33.2.81 CGHDRCAP : CCC Get HDR Capability Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x37C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSLEN	TSPEN	DDREN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DDREN	HDR-DDR Operation Enable 0: HDR-DDR Operation is Disabled. 1: HDR-DDR Operation is Enabled.	R/W
1	TSPEN	HDR-TSP Operation Enable 0: HDR-TSP Operation is Disabled. 1: HDR-TSP Operation is Enabled.	R/W
2	TSLEN	HDR-TSL Operation Enable 0: HDR-TSL Operation is Disabled. 1: HDR-TSL Operation is Enabled.	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

DDREN bit (HDR-DDR Operation Enable)

This bit is used to enable or disable the HDR-DDR communication when used with I3C Slave.

HDR-DDR communication is permitted only when SVDCT.TBCR5 = 1 and DDREN = 1.

[Setting condition]

- When writing 1.

[Clearing condition]

- When writing 0.

TSPEN bit (HDR-TSP Operation Enable)

This bit is used to enable or disable the HDR-TSP communication when used with I3C Slave.

HDR-TSP communication is permitted only when SVDCT.TBCR5 = 1 and TSPEN = 1.

[Setting condition]

- When writing 1.

[Clearing condition]

- When writing 0.

TSLEN bit (HDR-TSL Operation Enable)

This bit is used to enable or disable the HDR-TSL communication when used with I3C Slave.

HDR-TSL communication is permitted only when SVDCT.TBCR5 = 1 and TSLEN = 1.

[Setting condition]

- When writing 1.

[Clearing condition]

- When writing 0.

Note: Refer to the MIPI I3C Specification v1.0.
The lower 8 bits of this register are used as the response data of GETHDRCAP CCC.

33.2.82 BITCNT : Bit Count Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x380

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BCNT[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	BCNT[4:0]	Bit Counter Indicates the number of bits remaining to be transferred. For details on the values, see Table 33.7 and Table 33.8.	R
31:5	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3

BCNT[4:0] bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a sampling edge on the I3C_SCL line.

Table 33.7 I²C / Legacy I²C transfer (1 of 2)

BCNT[4:0]	Master		Slave	
	Address phase	Data phase	Address phase	Data phase
0x00	2 to 1 bits	2 to 1 bits	3 to 1 bits	2 to 1 bits
0x01	3 bits	3 bits	4 bits	3 bits

Table 33.7 I²C / Legacy I²C transfer (2 of 2)

BCNT[4:0]	Master		Slave	
	Address phase	Data phase	Address phase	Data phase
0x02	4 bits	4 bits	5 bits	4 bits
0x03	5 bits	5 bits	6 bits	5 bits
0x04	6 bits	6 bits	7 bits	6 bits
0x05	7 bits	7 bits	8 bits	7 bits
0x06	8 bits	8 bits	9 bits	8 bits
0x07	9 bits	9 bits	—	9 bits

Table 33.8 I3C transfer

BCNT[4:0]	SDR*1		HDR-DDR		HDR-TS
	Transmission	Reception	Command/Data	CRC	
0x00	1 bit	2 to 1 bits	19, 1 bits	11, 1 bits	1 Symbol
0x01	2 bits	3 bits	20, 2 bits	12, 2 bits	2 Symbols
0x02	3 bits	4 bits	3 bits	3 bits	3 Symbols
0x03	4 bits	5 bits	4 bits	4 bits	4 Symbols
0x04	5 bits	6 bits	5 bits	5 bits	5 Symbols
0x05	6 bits	7 bits	6 bits	6 bits	6 Symbols
0x06	7 bits	8 bits	7 bits	7 bits	7 Symbols
0x07	8 bits	9 bits	8 bits	8 bits	8 Symbols
0x08	9 bits	—	9 bits	9 bits	9 Symbols
0x09	—	—	10 bits	10 bits	10 Symbols
0x0A	—	—	11 bits	—	11 Symbols
0x0B	—	—	12 bits	—	12 Symbols
0x0C	—	—	13 bits	—	—
0x0D	—	—	14 bits	—	—
0x0E	—	—	15 bits	—	—
0x0F	—	—	16 bits	—	—
0x10	—	—	17 bits	—	—
0x11	—	—	18 bits	—	—

Note 1. The address phase is the same as in Table 33.7.

33.2.83 NQSTLV : Normal Queue Status Level Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x394

Bit position:	31	30	29	28	24	23	16	15	8	7	0
Bit field:	—	—	—	IBISCNT[4:0]		IBIQLV[7:0]		RSPQLV[7:0]		CMDQFLV[7:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
7:0	CMDQFLV[7:0]	Normal Command Queue Free Level*1 Number of free buffer entries currently in the Normal Command Queue. Reset value is the depth of the Normal Command Queue.	R

Bit	Symbol	Function	R/W
15:8	RSPQLV[7:0]	Normal Response Queue Level*1 Number of buffer entries currently in the Normal Response Queue.	R
23:16	IBIQLV[7:0]	Normal IBI Queue Level*1 Number of buffer entries currently in the Normal IBI Queue.	R
28:24	IBISCNT[4:0]	Normal IBI Status Count*2 Number of IBI Status entries currently in the Normal IBI Queue.	R
31:29	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note 1. These bits support all I3C mode.
Note 2. These bits support I3C master mode and I3C secondary master mode.

33.2.84 NDBSTLV0 : Normal Data Buffer Status Level Register 0

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x398

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	8	7	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDBLV[7:0]		TDBFLV[7:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
7:0	TDBFLV[7:0]	Normal Tx Data Buffer Free Level Indicates the number of free Tx Data Buffer entries in the Tx Data Queue. Reset value is the depth of the Tx Data Queue.	R
15:8	RDBLV[7:0]	Normal Rx Data Buffer Level Indicates the number of Rx Data Buffer entries in the Rx Data Queue.	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

33.2.85 NRSQSTLV : Normal Receive Status Queue Status Level Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x3C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RSQLV[7:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RSQLV[7:0]	Normal Receive Status Queue Level	R
31:8	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

33.2.86 HQSTLV : High Priority Queue Status Level Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x3C4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RSPQLV[7:0]								CMDQLV[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
7:0	CMDQLV[7:0]	High Priority Command Queue Level Number of free buffer entries currently in the High Priority Command Queue. Reset value is the depth of the High Priority Command Queue.	R
15:8	RSPQLV[7:0]	High Priority Response Queue Level Number of buffer entries currently in the High Priority Response Queue.	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

33.2.87 HDBSTLV : High Priority Data Buffer Status Level Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x3C8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15							8	7							0
Bit field:	RDBLV[7:0]								TDBFLV[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
7:0	TDBFLV[7:0]	High Priority Tx Data Buffer Free Level Indicates the number of free High Priority Tx Data Buffer entries in the High Priority Tx Data Queue. Reset value is the depth of the High Priority Tx Data Queue.	R
15:8	RDBLV[7:0]	High Priority Rx Data Buffer Level Indicates the number of High Priority Rx Data Buffer entries in the High Priority Rx Data Queue.	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

33.2.88 PRSTDBG : Present State Debug Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x3CC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	SDOLV	SCOLV	SDILV	SCILV
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit	Symbol	Function	R/W
0	SCILV	I3C_SCL Line Signal Level This bit is used to check the I3C_SCL Line level, in order to recover from errors and for debugging.	R
1	SDILV	I3C_SDA Line Signal Level This bit is used to check the I3C_SDA Line level, in order to recover from errors and for debugging.	R
2	SCOLV	SCL Output Level 0: I3C has driven the I3C_SCL pin low. 1: I3C has released the I3C_SCL pin.	R
3	SDOLV	SDA Output Level 0: I3C has driven the I3C_SDA pin low. 1: I3C has released the I3C_SDA pin.	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3

SCILV bit (I3C_SCL Line Signal Level)

This bit is used to check the I3C_SCL Line level, in order to recover from errors and for debugging.

SDILV bit (I3C_SDA Line Signal Level)

This bit is used to check the I3C_SDA Line level, in order to recover from errors and for debugging.

SCOLV bit (SCL Output Level)

This bit is used to select the output level of I3C_SCL pin.

SDOLV bit (SDA Output Level)

This bit is used to select the output level of I3C_SDA pin.

33.2.89 MSERRCNT : Master Error Counters Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x3D0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	M2ECNT[7:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	M2ECNT[7:0]	M2 Error Counter Counts I3C Type M2 errors on the I3C Bus. Cleared upon read out.	R
31:8	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

33.2.90 SC1CPT : SC1 Capture Monitor Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x3E0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SC1C[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SC1C[15:0]	SC1 Capture	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

SC1C[15:0] bit (SC1 Capture)

- Async Mode 0 (Asynchronous Basic Mode)
  After enabling ATCCNTE.ATCE, SC1C[15:0] Counter counts up from SC1C[15:0] count trigger[*1] to SCL rise edge next to ACK for the IBI, and capture it as SC1C[15:0].
- Async Mode 1 (Asynchronous Advanced Mode)
  After enabling ATCCNTE.ATCE, SC1C[15:0] Counter counts up from SC1C[15:0] count trigger[*1] to the first aME, and capture it as SC1C[15:0].

Note: As the timing control specification, the SC1C[15:0] counter value is included in the IBI frame as IBI data and is sent to I3C Master, therefore it is not necessary for the I3C Slave to read this register. If the I3C Slave needs to read this register, read it after completing the IBI frame.

Note 1. SW or external trigger can be selected by selection bits.

33.2.91 SC2CPT : SC2 Capture Monitor Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x3E4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SC2C[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SC2C[15:0]	SC2 Capture	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

SC2C[15:0] bits (SC2 Capture)

- Async Mode 0 (Asynchronous Basic Mode)
  After enabling ATCCNTE.ATCE, SC2C[15:0] Counter counts up from SCL rise edge next to ACK for the IBI transmitted from I3C Slave to SCL rise edge next to Tbit after Mandatory Byte, and capture it as SC2C[15:0].
- Async Mode 1 (Asynchronous Advanced Mode)
  After enabling ATCCNTE.ATCE, SC2C[15:0] Counter counts up from SCL rise edge next to ACK for the IBI transmitted from I3C Slave to SCL rise edge next to Tbit after Mandatory Byte, and capture it as SC2C[15:0].

Note: As the timing control specification, the SC2C[15:0] counter value is included in the IBI frame as IBI data and is sent to I3C Master, therefore it is not necessary for the I3C Slave to read this register. If the I3C Slave needs to read this register, read it after completing the IBI frame.

33.2.92 CECTL : Clock Enable Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLKE	Clock Enable 0: Clock disable 1: Clock enable	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

CLKE bit (Clock Enable)

- This bit controls enabling/disabling of clock supply of the communication function.

33.3 Operation

33.3.1 Data Structures

33.3.1.1 Command Descriptor

The write-only Command Descriptor structure is 64 bits in length. The Command Descriptor is put to the Command Queue with writes to the Command Queue Port (High Priority or Normal).

Write to the Command Queue Port (High Priority or Normal) in the following order:

1. First write : The least significant DWORD (Command Descriptor Structure Low).
2. Second write : The most significant DWORD (Command Descriptor Structure High).

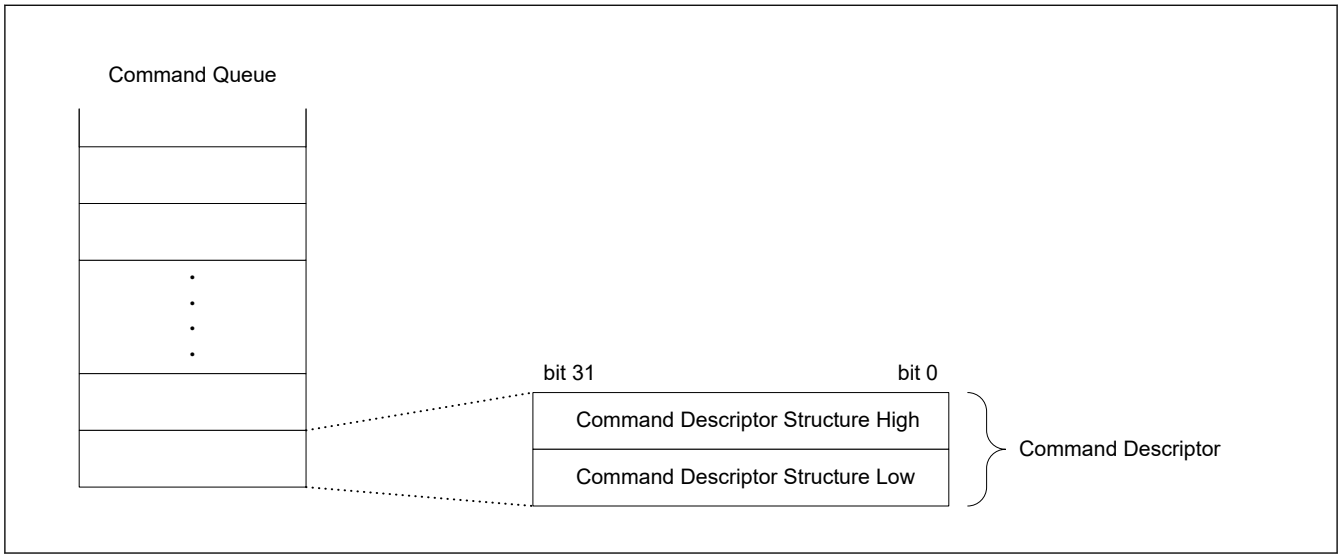


Figure 33.2 Command descriptor data structure

I3C provides a Command Descriptor structure for each command type as follows:

- Address Assign Command
- Immediate Transfer Command
- Regular Transfer Command
- Combo Transfer Command
- Internal Control Command

Details are explained in the following sections.

33.3.1.1.1 Address Assign Command

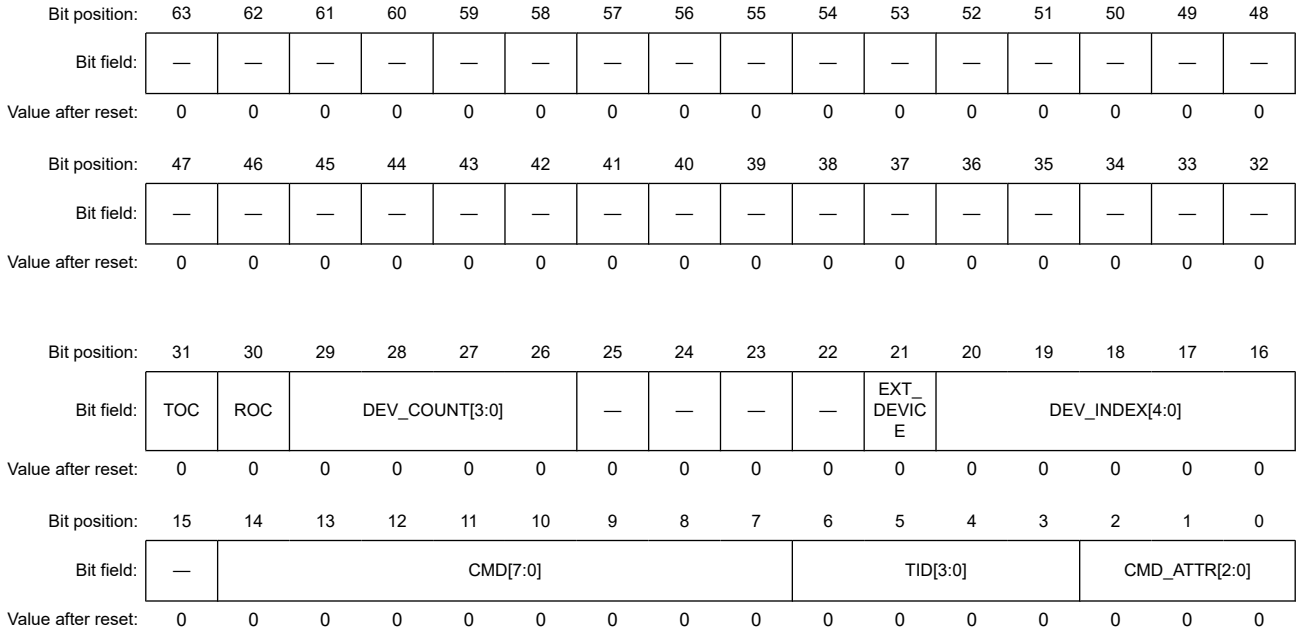
This command is used for address assignment (ENTDAA, SETDASA).

Note: When issuing SETAASA CCC, use the Immediate Transfer command.

The I3C provides an address assign command for the following mode:

- I3C Master mode

Details of the Address Assign command structure are as follows.



Bit position:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TOC	ROC	DEV_COUNT[3:0]				—	—	—	—	EXT_DEVICE	DEV_INDEX[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CMD[7:0]								TID[3:0]				CMD_ATTR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CMD_ATTR[2:0]	Command Attributes 0x0: XFER: Regular Transfer 0x1: IMMED_DATA_XFER: Immediate Data Transfer 0x2: ADDR_ASSGN_CMD: Address Assignment Command 0x3: WWR_COMBO_XFER: Write + Write/Read Combo Transfer 0x7: INTERNAL_CONTROL: Internal Control command Others: Setting prohibited	W
6:3	TID[3:0]	Transaction ID	W
14:7	CMD[7:0]	Transfer Command CCC Value	W
15	—	The write value should be 0.	W
20:16	DEV_INDEX[4:0]	Device Index	W
21	EXT_DEVICE	Extended Device Index 0: Use the DATBASm table indicated by DEV_INDEX[4:0]. 1: Use the EXDATBAS table.	W
25:22	—	The write value should be 0.	W
29:26	DEV_COUNT[3:0]	Device Count	W
30	ROC	Response on Completion 0: NOT_REQUIRED: Response Status is not required. 1: REQUIRED: Response Status is required.	W
31	TOC	Terminate on Completion 0: RESTART: Issue Repeated START (Sr) at end of transfer 1: STOP: Issue STOP (P) at end of transfer	W
63:32	—	The write value should be 0.	W

CMD_ATTR[2:0] bits (Command Attributes)

Command Type, defining the format of the other fields.

TID[3:0] bits (Transaction ID)

Used as an identification tag for this command. This field shall be populated by the software Driver, and the same value shall be reflected in the Response Descriptor.

CMD[7:0] bits (Transfer Command CCC Value)

Specifies CCC code indicating whether Address Assignment uses ENTDAA or SETDASA commands. The field comprises the entire command code (ENTDAA or SETDASA).

DEV_INDEX[4:0] bits (Device Index)

Indicates the DATBASm table index for the Slave device being addressed with the transfer. Static and device addressing related information are stored to this index in the DATBASm.

DEV_COUNT[3:0] bits (Device Count)

Indicates the number of devices that a dynamic address is assigned to.

ROC bit (Response on Completion)

Controls whether Response Status is sent after successful completion of the Transfer command. The successful completion is read from register NRSPQP. Upon unsuccessful transfer the Response Status is sent.

TOC bit (Terminate on Completion)

Controls what bus condition to issue after the Transfer command completes.

For ENTDAA, a STOP condition is issued regardless of the setting value of TOC. It is meaningful for SETDASA transfers.

When sending SETDASA CCC by TOC = 0 (RESTART), the next command must be set to SETDASA CCC with the Address Assign Command.

When the next command is not the same SETDASA CCC flame, it must be set to TOC = 1 (STOP).

33.3.1.1.2 Immediate Transfer Command

This structure directly contains data (max 4 bytes) to be transferred, and as a result is only useful for Transfers/CCCs that write data. This structure shall not be used for Read operations (for example, to Rx data).

When transmitting data of 4 bytes or less, use this Immediate Transfer Command to communicate.

When transmitting data of 5 bytes or more, use the Regular Transfer Command to communicate.

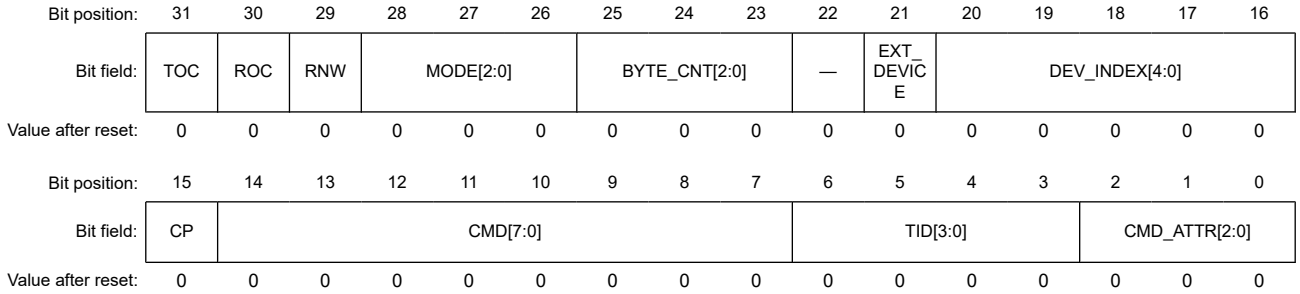
For the Regular Transfer Command, see section 33.3.1.1.3. Regular Transfer Command.

I3C provides an Immediate Transfer Command for the following mode:

- I3C Master Mode

Details of the Immediate Transfer Command Structure of each mode are shown in this section.

Bit position:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Bit field:	DATA_BYTE_4[7:0]								DATA_BYTE_3[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field:	DATA_BYTE_2[7:0]								DATA_BYTE_1[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TOC	ROC	RNW	MODE[2:0]			BYTE_CNT[2:0]			—	EXT_DEVICE	DEV_INDEX[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CP	CMD[7:0]								TID[3:0]				CMD_ATTR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CMD_ATTR[2:0]	Immediate Data Transfer Command Attribute 0x0: XFER: Regular Transfer 0x1: IMMED_DATA_XFER: Immediate Data Transfer 0x2: ADDR_ASSGN_CMD: Address Assignment Command 0x3: WWR_COMBO_XFER: Write + Write/Read Combo Transfer 0x7: INTERNAL_CONTROL: Internal Control command Others: Setting prohibited	W
6:3	TID[3:0]	Immediate Data Transfer Transaction ID	W
14:7	CMD[7:0]	Immediate Data Transfer CCC / HDR Command Code Value For CCC: 8 bits For HDR: 7 bits	W
15	CP	Immediate Data Transfer Command Present 0: TRANFSER: This structure describes an SDR transfer, so the CMD field is not valid. 1: CCC_HDR: This structure describes a CCC or HDR transfer, so the CMD field is valid.	W
20:16	DEV_INDEX[4:0]	Immediate Data Transfer Device Index	W
21	EXT_DEVICE	Immediate Data Transfer Extended Device Index 0: Use the DATBASm table indicated by DEV_INDEX[4:0]. 1: Use the EXDATBAS table.	W
22	—	The write value should be 0.	W
25:23	BYTE_CNT[2:0]	Immediate Data Transfer Byte Count 0x0: No payload 0x1 to 0x4: N bytes are valid. Others: Setting prohibited	W
28:26	MODE[2:0]	Immediate Data Transfer Mode and Speed Values 0x0: I3C SDR0 / Data rate : STDBR (I3C mode) Legacy I^2C Message 0 / Data rate : STDBR (I^2C mode) 0x1: I3C SDR1 / Data rate : EXTBR (I3C mode) Legacy I^2C Message 0 / Data rate : EXTBR (I2C mode) 0x2: I3C SDR2 / Data rate : STDBR × 2 (I3C mode) Reserved (I^2C mode) 0x3: I3C SDR3 / Data rate : EXTBR × 2 (I3C mode) Reserved (I^2C mode) 0x4: I3C SDR4 / Data rate : EXTBR × 4 (I3C mode) Reserved (I^2C mode) 0x5: I3C HDR-TS / Data rate : STDBR (I3C mode) HDR-Ternary Mode. Selection of HDR-TSP vs. HDR-TSL depends on the value of register BCTL, field BMDS Reserved (I^2C mode) 0x6: I3C HDR-DDR / Data rate : STDBR (I3C mode) Reserved (I^2C mode) Others: Setting prohibited	W
29	RNW	Immediate Data Transfer R/W 0: WRITE: Write transfer 1: READ: Read transfer	W

Bit	Symbol	Function	R/W
30	ROC	Immediate Data Transfer Response on Completion 0: NOT_REQUIRED: Response Status is not required. 1: REQUIRED: Response Status is required.	W
31	TOC	Immediate Data Transfer Terminate on Completion 0: RESTART: Issue Repeated START (Sr) at end of data transfer 1: STOP: Issue STOP (P) at end of data transfer	W
39:32	DATA_BYTE_1[7:0]	Immediate Data Transfer Data Byte 1 Direct argument	W
47:40	DATA_BYTE_2[7:0]	Immediate Data Transfer Data Byte 2 Direct argument	W
55:48	DATA_BYTE_3[7:0]	Immediate Data Transfer Data Byte 3 Direct argument	W
63:56	DATA_BYTE_4[7:0]	Immediate Data Transfer Data Byte 4 Direct argument	W

CMD_ATTR[2:0] bits (Immediate Data Transfer Command Attribute)

Command Type, defining the format of the other fields.

TID[3:0] bits (Immediate Data Transfer Transaction ID)

Used as an identification tag for this command. This field shall be populated by the software Driver, and the same value shall be reflected in the Response Descriptor.

CP bit (Immediate Data Transfer Command Present)

Indicates whether CMD field is valid for CCC or HDR Transfer.

DEV_INDEX[4:0] bits (Immediate Data Transfer Device Index)

Indicates the DATBASm Table index for the Slave Device being addressed with the transfer. Static and Device addressing related information will be stored to this index in the DATBASm.

BYTE_CNT[2:0] bits (Immediate Data Transfer Byte Count)

Number of valid data bytes to use in this Immediate Data Transfer Descriptor.

This field must be set to non-zero value, except for CCCs that does not have payload defined.

This field must be set to even value, when specifying HDR mode (MODE[2:0] = 0x5 or 0x6).

MODE[2:0] bits (Immediate Data Transfer Mode and Speed Values)

Sets the mode and speed for the I3C or I^2C transfer.

Interpretation of this field depends on whether the Device is in I3C Mode vs. I^2C Mode (see the DEVICE field in the DATBASm Table entry indexed by field DEV_INDEX).

RNW bit (Immediate Data Transfer R/W)

Identifies direction of the transfer.

This field shall always be set to 0, because Immediate transfers are valid for Write transactions only.

ROC bit (Immediate Data Transfer Response on Completion)

Controls whether Response Status is required after successful completion of the data transfer command. The successful completion shall be read from NRSPQP register. Upon unsuccessful transfer the Response Status shall always be sent.

TOC bit (Immediate Data Transfer Terminate on Completion)

Controls what Bus condition is issued after completion of the data transfer.

When sending Direct CCC by TOC = 0 (RESTART), next command must be set to same Direct CCC.

When the next command is not the same Direct CCC, must be set to TOC = 1 (STOP).

33.3.1.1.3 Regular Transfer Command

This structure does not contain data to be transferred.

For Master Mode, the data buffer is available through Transfer Data Queue Port (Rx Data Queue Port and Tx Data Queue Port).

When transmitting data of 5 bytes or more, use this Regular Transfer Command to communicate.

When transmitting data of 4 bytes or less, use the Immediate Transfer Command to communicate.

For the Immediate Transfer Command, see section 33.3.1.1.2. Immediate Transfer Command.

For I3C Slave Mode, the IBI Payload buffer is available through IBI Queue Port.

I3C provides a Regular Transfer Command for the following modes:

- I3C Master Mode
- I3C Slave Mode

Details of the regular transfer command structure of each mode are as follows.

(1) I3C Master Mode

Bit position:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Bit field:	DATA_LENGTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TOC	ROC	RNW	MODE[2:0]			—	—	—	—	EXT_DEVICE	DEV_INDEX[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CP	CMD[7:0]								TID[3:0]				CMD_ATTR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CMD_ATTR[2:0]	Data Transfer Command Attribute Command Type, defining the format of the other fields. Values: 0x0: XFER: Regular Transfer 0x1: IMMED_DATA_XFER: Immediate Data Transfer 0x2: ADDR_ASSGN_CMD: Address Assignment Command 0x3: WWR_COMBO_XFER: Write + Write/Read Combo Transfer 0x7: INTERNAL_CONTROL: Internal Control command Others: Setting prohibited	W
6:3	TID[3:0]	Data Transfer Transaction ID Identification tag for this command	W
14:7	CMD[7:0]	Data Transfer CCC / HDR Command Code Value Specifies the I3C Command code For CCC: 8 bits For HDR: 7 bits	W
15	CP	Data Transfer Command Present 0: TRANFSER: This structure describes an SDR transfer, so the CMD field is not valid. 1: CCC_HDR: This structure describes a CCC or HDR transfer, so the CMD field is valid.	W

Bit	Symbol	Function	R/W
20:16	DEV_INDEX[4:0]	Data Transfer Device Index	W
21	EXT_DEVICE	Data Transfer Extended Device Index 0: Use the DATBASm Table indicated by DEV_INDEX[4:0]. 1: Use the EXDATBAS table.	W
25:22	—	The write value should be 0.	W
28:26	MODE[2:0]	Data Transfer Speed and Mode 0x0: I3C SDR0 / Data rate : STDBR (I3C mode) Legacy I2C Message 0 / Data rate : STDBR (I2C mode) 0x1: I3C SDR1 / Data rate : EXTBR (I3C mode) Legacy I2C Message 0 / Data rate : EXTBR (I2C mode) 0x2: I3C SDR2 / Data rate : STDBR × 2 (I3C mode) Reserved (I2C mode) 0x3: I3C SDR3 / Data rate : EXTBR × 2 (I3C mode) Reserved (I2C mode) 0x4: I3C SDR4 / Data rate : EXTBR × 4 (I3C mode) Reserved (I2C mode) 0x5: I3C HDR-TS / Data rate : STDBR × 4 (I3C mode) HDR-Ternary Mode. Selection of HDR-TSP vs. HDR-TSL depends on the value of register BCTL, field BMDS Reserved (I2C mode) 0x6: I3C HDR-DDR / Data rate : STDBR (I3C mode) Reserved (I2C mode) Others: Setting prohibited	W
29	RNW	Data Transfer R/W 0: WRITE: Write transfer 1: READ: Read transfer	W
30	ROC	Data Transfer Response on Completion 0: NOT_REQUIRED: Response Status is not required. 1: REQUIRED: Response Status is required.	W
31	TOC	Data Transfer Terminate on Completion 0: RESTART: Issue Repeated START (Sr) at end of transfer 1: STOP: Issue STOP (P) at end of transfer	W
47:32	—	The write value should be 0.	W
63:48	DATA_LENGTH [15:0]	Data Transfer Data Length Indicates the number of bytes to be transferred. This field must be set to non-zero value, except for CCCs that does not have payload defined.	W

CMD_ATTR[2:0] bits (Data Transfer Command Attribute)

Command Type, defining the format of the other fields.

TID[3:0] bits (Data Transfer Transaction ID)

Used as an identification tag for this command. This field shall be populated by the software Driver, and the same value shall be reflected in the Response Descriptor.

CP bit (Data Transfer Command Present)

Indicates whether he contents of the CMD field is valid for a CCC or HDR Transfer.

DEV_INDEX[4:0] bits (Data Transfer Device Index)

Indicates the DATBASm Table index for the Slave Device being addressed with the transfer. Static and Device addressing related information will be stored to this index in the DATBASm.

MODE[2:0] bits (Data Transfer Speed and Mode)

Sets the mode and speed for the I3C or I^{2}C transfer.

Interpretation of this field depends on whether the Device is in I3C Mode vs. I^{2}C Mode (see the DEVICE field in the DATBASm Table entry indexed by field DEV_INDEX).

RNW bit (Data Transfer R/W)

Identifies direction of the transfer.

ROC bit (Data Transfer Response on Completion)

Controls whether Response Status is required after successful completion of the transfer command. The successful completion shall be read from NRSPQP register. Upon unsuccessful transfer the Response Status shall always be sent.

TOC bit (Data Transfer Terminate on Completion)

Controls what Bus condition will be issued after completion of the transfer.

When sending Direct CCC by TOC = 0 (RESTART), next command must be set to same Direct CCC.

When the next command is not the same Direct CCC, must be set to TOC = 1 (STOP).

DATA_LENGTH[15:0] bits (Data Transfer Data Length)

Number of valid data bytes to use in this Regular Transfer Descriptor.

This field must be set to non-zero value, except for CCCs that does not have payload defined.

This field must be set to even value, when specifying HDR mode (MODE[2:0] = 0x5 or 0x6).

Length setting of GETMXDS command should be fixed to 5.

(2) I3C Slave Mode

Bit position:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Bit field:	DATA_LENGTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	ROC	RNW	—	—	—	—	—	—	ITS	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	TID[3:0]				CMD_ATTR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CMD_ATTR[2:0]	Data Transfer Command Attribute Command Type, defining the format of the other fields. Values: 0x0: XFER: Regular Transfer 0x1: IMMED_DATA_XFER: Immediate Data Transfer 0x2: ADDR_ASSGN_CMD: Address Assignment Command 0x3: WWR_COMBO_XFER: Write + Write/Read Combo Transfer 0x7: INTERNAL_CONTROL: Internal Control command Others: Setting prohibited	W
6:3	TID[3:0]	Data Transfer Transaction ID Identification tag for this command	W

Bit	Symbol	Function	R/W
21:7	—	The write value should be 0.	W
22	ITS	Include timestamp for Async Mode 0: Do not include timestamp. 1: Include timestamp.	W
28:23	—	The write value should be 0.	W
29	RNW	Data Transfer R/W 0: WRITE: Write transfer (Mastership Request) 1: READ: Read transfer (Slave Interrupt Request)	W
30	ROC	Data Transfer Response on Completion 0: NOT_REQUIRED: Response Status is not required. 1: REQUIRED: Response Status is required.	W
47:31	—	The write value should be 0.	W
63:48	DATA_LENGTH [15:0]	Data Transfer Data Length Indicates the number of bytes to be transferred. This field must be set to non-zero value, except for CCCs that does not have payload defined.	W

CMD_ATTR[2:0] bits (Data Transfer Command Attribute)

Command Type, defining the format of the other fields.

TID[3:0] bits (Data Transfer Transaction ID)

Used as an identification tag for this command. This field shall be populated by the software Driver, and the same value shall be reflected in the Response Descriptor.

RNW bit (Data Transfer R/W)

Identifies direction of the transfer.

ROC bit (Data Transfer Response on Completion)

Controls whether Response Status is required after successful completion of the transfer command. The successful completion shall be read from NRSPQP register. Upon unsuccessful transfer the Response Status shall always be sent.

33.3.1.1.4 Combo Transfer Command

This structure contains a combined Write + Read/Write operation.

The data buffer is available through Transfer Data Queue Port (Rx Data Queue Port and Tx Data Queue Port).

I3C provides a Combo Transfer Command for the following mode:

- I3C Master mode

Details of the Combo Transfer Command Structure of each mode are as follows.

Bit position:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Bit field:	DATA_LENGTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field:	OFFSET[15:0]/SUBOFFSET[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TOC	ROC	RNW	MODE[2:0]			16_BIT_SUBOFFSET	FIRST_PHASE_MODE	DATA_LENGTH_POSITION[1:0]		EXT_DEVICE	DEV_INDEX[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CP	CMD[7:0]								TID[3:0]				CMD_ATTR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CMD_ATTR[2:0]	Combo Transfer Command Attribute Command Type, defining the format of the other fields. 0x0: XFER: Regular Transfer 0x1: IMMED_DATA_XFER: Immediate Data Transfer 0x2: ADDR_ASSGN_CMD: Address Assignment Command 0x3: WWR_COMBO_XFER: Write + Write/Read Combo Transfer 0x7: INTERNAL_CONTROL: Internal Control command Others: Setting prohibited	W
6:3	TID[3:0]	Combo Transfer Transaction ID Identification tag for the command	W
14:7	CMD[7:0]	Combo Transfer HDR Command Code Value Specifies the I3C Command code (7 bits).	W
15	CP	Combo Transfer Command Present Indicates whether the CMD field is valid for a HDR Transfer 0: TRANFSER: This structure describes an SDR transfer, so the CMD field is not valid. 1: CCC_HDR: This structure describes a HDR transfer, so the CMD field is valid.	W
20:16	DEV_INDEX[4:0]	Combo Transfer Device Index	W
21	EXT_DEVICE	Combo Transfer Extended Device Index 0: Use the DATBASm table indicated by DEV_INDEX[4:0]. 1: Use the EXDATBAS table.	W
23:22	DATA_LENGTH_POSITION[1:0]	Data Length Field Position 0 0: NO: Do not put length field. 0 1: FIRST: Put length as first field. 1 0: SECOND: Put length as second field. Others: Setting prohibited	W
24	FIRST_PHASE_MODE	Combo Transfer First Phase Mode 0: SDR: First phase is executed in SDR mode. 1: MODE: First phase is executed in the mode indicated by the MODE field.	W
25	16_BIT_SUBOFFSET	Combo Transfer Sub Offset Size 0: 8_BIT_SUBOFFSET: Sub-offset is 8-bits long. Value is encoded in Lower Byte of OFFSET / SUBOFFSET field. 1: 16_BIT_SUBOFFSET: Sub-offset is 16-bits long.	W
28:26	MODE[2:0]	Combo Transfer Speed and Mode Values for I3C Mode 0x0: I3C SDR0 / Data rate : STDBR 0x1: I3C SDR1 : Data rate : EXTBR 0x2: I3C SDR2 / Data rate : STDBR × 2 0x3: I3C SDR3 / Data rate : EXTBR × 2 0x4: I3C SDR4 / Data rate : EXTBR × 4 0x5: I3C HDR-TS / Data rate : STDBR × 4 HDR-Ternary Mode. Selection of HDR-TSP vs. HDR-TSL depends on the value of register BCTL, field BMDS 0x6: I3C HDR-DDR / Data rate : STDBR Others: Setting prohibited	W

Bit	Symbol	Function	R/W
29	RNW	Combo Transfer R/W Identifies direction of the transfer 0: WRITE: Write transfer 1: READ: Read transfer	W
30	ROC	Combo Transfer Response on Completion 0: NOT_REQUIRED: Response Status is not required. 1: REQUIRED: Response Status is required.	W
31	TOC	Combo Transfer Terminate on Completion 0: RESTART: Issue Repeated START (Sr) at end of transfer 1: STOP: Issue STOP (P) at end of transfer	W
47:32	OFFSET[15:0]/ SUBOFFSET[15:0]	Combo Transfer Offset / Sub-Offset Offset of the target operation	W
63:48	DATA_LENGTH [15:0]	Combo Transfer Data Length Number of bytes to be transferred. This field must be set to non-zero value.	W

CMD_ATTR[2:0] bits (Combo Transfer Command Attribute)

Command Type, defining the format of the other fields.

TID[3:0] bits (Combo Transfer Transaction ID)

Used as an identification tag for this command. This field shall be populated by the software Driver, and the same value shall be reflected in the Response Descriptor.

CP bit (Combo Transfer Command Present)

Indicates whether the contents of the CMD field is valid for a HDR Transfer.

DEV_INDEX[4:0] bits (Combo Transfer Device Index)

Indicates the DATBASm table index for the Slave Device being addressed with the transfer. Static and Device addressing related information will be stored to this index in the DATBASm.

DATA_LENGTH_POSITION[1:0] bits (Data Length Field Position)

Indicates whether and where to put Data Length (DATA_LENGTH) in the first phase of the transfer. This field is only applicable if first phase of the transfer is executed in HDR mode.

Whether 8-bit or 16-bit of Data Length field is used is indicated with 16_BIT_SUBOFFSET field. In case of 8-bit value, it is encoded in Lower Byte of DATA_LENGTH field.

FIRST_PHASE_MODE bits (Combo Transfer First Phase Mode)

Indicates whether the first phase of the Combo Transfer is executed in SDR mode, or the mode indicated by the MODE field.

MODE[2:0] bits (Combo Transfer Speed and Mode Values for I3C Mode)

Sets the mode and speed for the I3C or I^2C transfer.

Interpretation of this field depends on whether the Device is in I3C mode or I^2C mode (see the DEVICE field in the DATBASm Table entry indexed by field DEV_INDEX).

RNW bit (Combo Transfer R/W Identifies direction of the transfer)

Identifies direction of the transfer.

ROC bit (Combo Transfer Response on Completion)

Controls whether Response Status is required after successful completion of the data transfer command. The successful completion shall be read from NRSPQP register. Upon unsuccessful transfer the Response Status shall always be sent.

TOC bit (Combo Transfer Terminate on Completion)

Controls what Bus condition is issued after completion of the data transfer.

When sending 2nd flame by TOC = 0 (RESTART) and HDR mode (MODE[2:0] = 0x5 or 0x6), next command must be set to that 1st flame will be the same as HDR mode (MODE[2:0] = 0x5 or 0x6).

When the next command is not the same HDR mode (MODE[2:0] = 0x5 or 0x6), must be set to TOC = 1 (STOP).

DATA_LENGTH[15:0] bit (Combo Transfer Data Length)

Number of valid data bytes to use in this Combo Transfer Descriptor.

This field must be set to non-zero value.

This field must be set to even value, when specifying HDR mode (MODE[2:0] = 0x5 or 0x6).

33.3.1.1.5 Internal Control Command

This structure is used for controlling I3C itself (not for transfer commands).

I3C provides an Internal Control Command for the following mode:

- I3C Master mode

Details of the Internal Control Command Structure are as follows:

Bit position:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	ON_OFF	MIPI_CMD[3:0]				—	TID[3:0]				CMD_ATTR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CMD_ATTR[2:0]	Command Attribute*2 Command Type, defining the format of the other fields. 0x0: XFER: Regular Transfer 0x1: IMMED_DATA_XFER: Immediate Data Transfer 0x2: ADDR_ASSGN_CMD: Address Assignment Command 0x3: WWR_COMBO_XFER: Write + Write/Read Combo Transfer 0x7: INTERNAL_CONTROL: Internal Control command Others: Setting prohibited	W
6:3	TID[3:0]	Transaction ID Identification tag for the command	W
7	—	The write value should be 0.	W
11:8	MIPI_CMD[3:0]	MIPI Alliance Command 0x00: NoOp, so the ON_OFF field is not valid. 0x02: Include 7E (IBA), so the ON_OFF field is valid. Others: Setting prohibited	W
12	ON_OFF	Bus Instance 7E On / Off*1 Enables or disables automatic transmission of the I3C Broadcast Header after every START condition on this I3C Bus instance. 0: IBA_INCLUDE off 1: IBA_INCLUDE on	W

Bit	Symbol	Function	R/W
63:13	—	The write value should be 0.	W

Note 1. The IBA_INCLUDE on state set by MIPI_CMD [3:0] = 0x2 and ON_OFF = 1 is cleared by setting RSTCTL.INTLRST to 1.
Note 2. The Response descriptor is not stored when the Internal Control Command is executed.

33.3.1.2 Response Descriptor

The Response Descriptor is a read-only structure describing the success or failure of a command, and the amount of data transferred.

The Response Descriptor is read from Response Queue with reads from Response Queue Port.

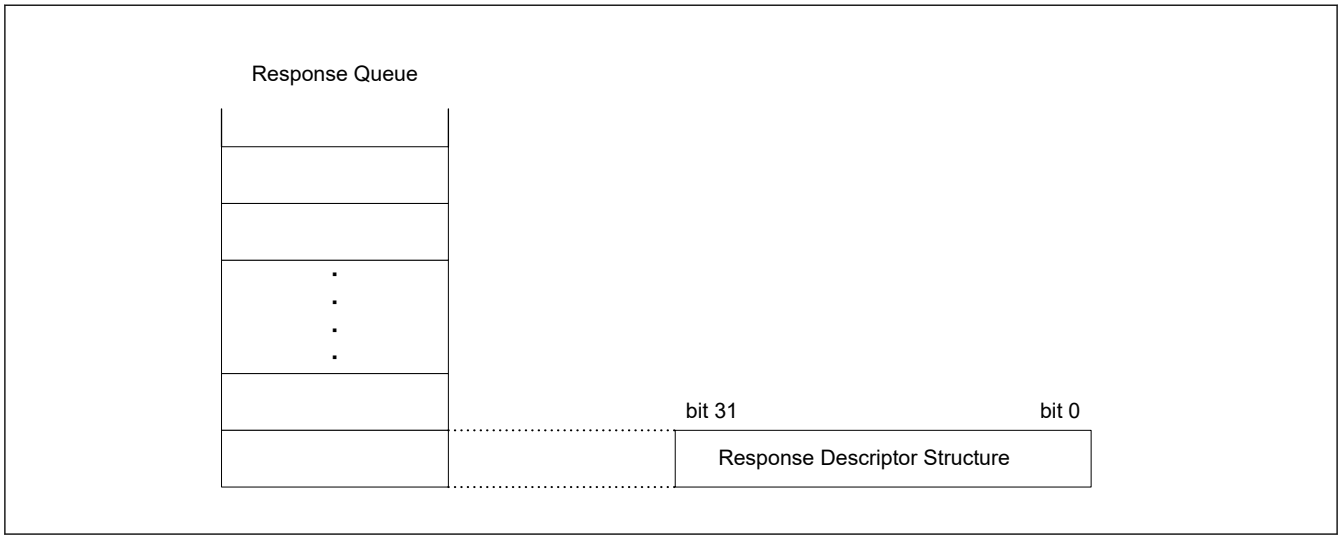


Figure 33.3 Response descriptor data structure

I3C provides a Response Descriptor for the following modes:

- I3C Master mode
- I3C Slave mode

Details of the Response Descriptor structure of each mode are shown in the following sections.

(1) I3C Master Mode

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ERR_STATUS[3:0]				TID[3:0]				—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DATA_LENGTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DATA_LENGTH [15:0]	Data Length / Device Count The meaning of this field depends on the context: For Write Transfer: Remaining data length (in bytes) For Read Transfer: Received data length (in bytes) For Address Assignment: Remaining Device count	R
23:16	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
27:24	TID[3:0]	Command/Response Transaction ID Identification tag for the command. This value shall match one of commands sent on the Bus. 0x0-0x7: Valid Transaction IDs Others: Reserved	R
31:28	ERR_STATUS[3:0]	Response Error Status 0x0: SUCCESS: Transfer successful, no error 0x1: CRC: CRC Error 0x2: PARITY: Parity Error 0x3: FRAME: Frame Error 0x4: ADDR_HEADER: Address Header Error 0x5: NACK: Address NACKed or Dynamic Address Assignment NACKed 0x6: OVL: Receive Overflow or Transfer Underflow Error 0x8: ABORTED: Aborted 0x9: I^2C_WR_DATA_NACK: NACK received for the I^2C Write Data transfer 0xA: NOT_SUPPORTED: Command with specific parameters not supported by I3C implementation (for example, specific Internal Control codes may not be supported) Others: Reserved	R

Note: In I3C Master mode, when an abnormal command with a specific parameter that is not supported is stored in Command Descriptor, it is indicated as NOT_SUPPORTED (0xA) in ERR_STATUS [3:0].

(2) I3C Slave Mode

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ERR_STATUS[3:0]				TID[3:0]				—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DATA_LENGTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DATA_LENGTH [15:0]	Data Length Remaining data length (in bytes) for Slave Interrupt Request	R
23:16	—	These bits are read as 0.	R
27:24	TID[3:0]	Command/Response Transaction ID Identification tag for the command. This value matches one of commands sent on the bus. 0x0-0x7: Valid Transaction IDs Others: Reserved	R
31:28	ERR_STATUS[3:0]	Response Error Status 0x0: SUCCESS: Transfer successful, no error. 0x3: FRAME: Frame Error 0x4: ADDR_HEADER: Address Header Error 0x5: NACK: Address NACK'ed or Dynamic Address Assignment NACK'ed 0x6: OVL: Receive Overflow or Transfer Underflow Error 0x8: ABORTED: Aborted 0xA: NOT_SUPPORTED: Command with specific parameters not supported by I3C implementation (for example, specific Internal Control codes may not be supported) Others: Reserved	R

Note: In I3C Slave mode, it is indicated as NOT_SUPPORTED (0xA) in ERR_STATUS[3:0] in the following cases:

- When an abnormal command with a specific parameter that is not supported is stored in the Command Descriptor.
- When the IBI to be transmitted is disabled in the CSECMD register.
- After the normal command for IBI transmission is prepared in the Command Queue, when that IBI is disabled in the CSECMD register by the DISEC CCC frame from the I3C Master.

33.3.1.3 IBI Status Descriptor

The IBI Status Descriptor is a read-only structure describing an IBI event received from a Slave device on the I3C Bus.

The IBI Status Descriptor is read from IBI Queue with reads from IBI Queue Port.

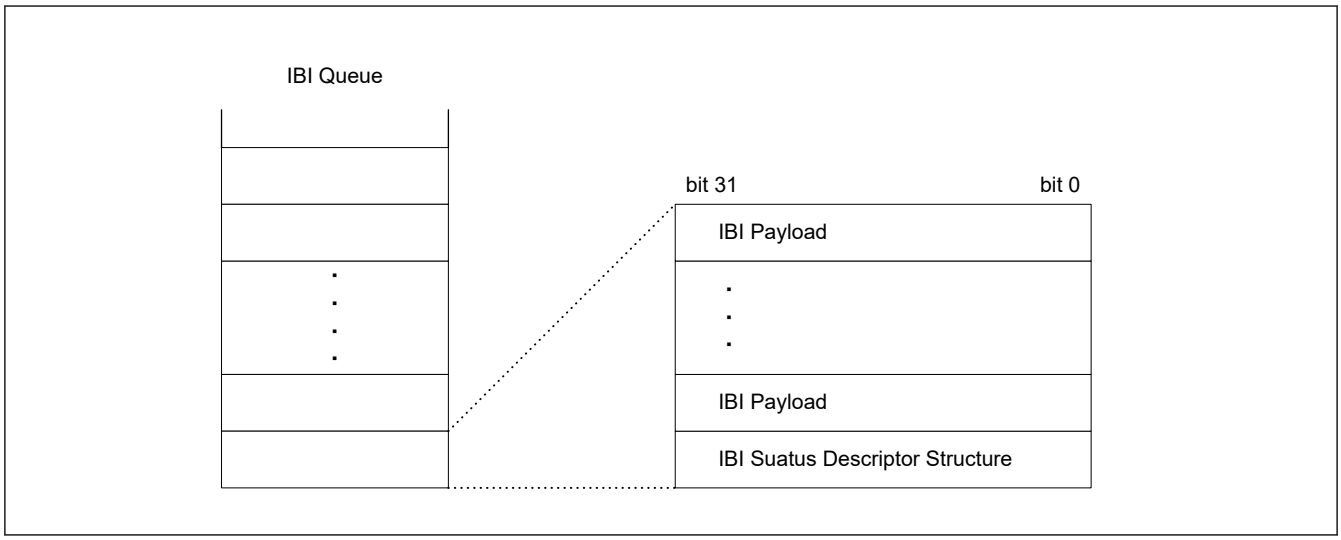


Figure 33.4 IBI status descriptor data structure

I3C provides a IBI Status Descriptor for the following mode:

- I3C Master mode

Details of the IBI Status Descriptor Structure are as follows.

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	IBI_ST	—	—	ERR_STATUS[2:0]			TS	LAST_STATUS	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	IBI_ID[7:0]								DATA_LENGTH[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DATA_LENGTH[7:0]	IBI Data Length Number of data bytes in IBI Data.	R
15:8	IBI_ID[7:0]	IBI Received ID The meaning of this field depends on the context: For Slave Interrupt or Master Request: Bits 15:9 contain the Slave's Device Address, and bit 8 contains the R/W bit.	R
23:16	—	These bits are read as 0. The write value should be 0.	R
24	LAST_STATUS	Last IBI Status Last IBI status for the IBI transaction.	R

Bit	Symbol	Function	R/W
25	TS	IBI Time-stamp Present Indicates whether a time-stamp is available for the IBI. 0: OFF: IBI is not time-stamped. 1: ON: IBI is time-stamped.	R
28:26	ERR_STATUS[2:0]	IBI Error Status 0x0: SUCCESS 0x3: ERROR: FRAME (Frame Error) 0x4: ERROR: ADDR_HEADER (Address Header Error) 0x5: NACK: Address NACKed 0x7: ERROR: ABORT (Aborted to Master) Others: Reserved	R
30:29	—	These bits are read as 0.	R
31	IBI_ST	IBI Received Status Indicates how the received IBI was handled. 0: The IBI was handled with ACK. 1: NACK: The IBI was handled with NACK, and then Auto-Disabled.	R

LAST_STATUS bits (Last IBI Status)

Even if LAST_STATUS is set to 0, the software driver still evaluates the data payload length by examining the DATA_LENGTH field.

33.3.1.4 Receive Status Descriptor

The Receive Status Descriptor is a read-only structure describing the success or failure of read/write operation from the master, and the amount of data transferred.

The Receive Status Descriptor is read from Receive Status Queue with reads from Receive Status Queue Port.

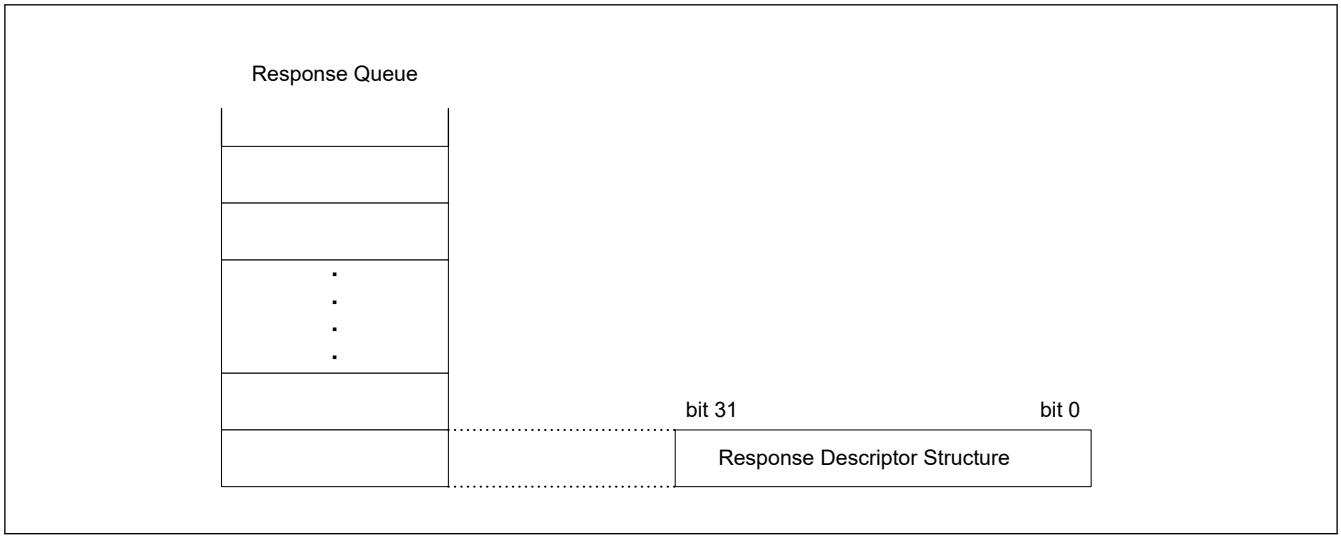


Figure 33.5 Receive status descriptor data structure

I3C provides a Receive Status Descriptor for the following mode:

- I3C Slave mode

Details of the Receive Status Descriptor structure of each mode are as follows.

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DEV_INDEX[2:0]			TRANSFER_TYPE[1:0]		ERR_STATUS[2:0]			CMD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DATA_LENGTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DATA_LENGTH [15:0]	Data Length The meaning of this field depends on the context. For Write Transfer: Received data length (in bytes) For Read Transfer: Transmitted data length (in bytes)	R
23:16	CMD[7:0]	The contents are different depending on the operation mode. Details are as follows: [SDR Private Message Mode] CMD[7]: R/W Type CMD[6:4]: Reserved CMD[3]: I3C_I^{2}C Type CMD[2:0]: Reserved [SDR CCC Mode] CCC code[7:0] [HDR-DDR mode] HDR CMD[7:0] 0x00 – 0x7F: Write Command 0x80 – 0xFF: Read Command [HDR-TSL/TSP mode] HDR CMD[7:0]	R
26:24	ERR_STATUS[2:0]	Error Status 0x0: SUCCESS 0x1: ERROR: CRC (CRC Error) 0x2: ERROR: PARITY (Parity Error) 0x3: ERROR: FRAME (Frame Error) 0x4: ERROR: ADDR_HEADER (Address Header Error) 0x5: ERROR: NACK (Slave NACKed) 0x6: ERROR: OVL (FIFO Overflow/Underflow) 0x7: ERROR: ABORT (Aborted to Master)	R
28:27	TRANSFER_TYPE [1:0]	Transfer Type 0 0: I3C SDR/I^{2}C Message 0 1: I3C CCC 1 0: I3C HDR-DDR 1 1: I3C HDR-TS	R
31:29	DEV_INDEX[2:0]	Device Index Indicates the SVDVADn index for the response with the transfer.	R

33.3.2 Details of Function

33.3.2.1 Operation Mode

The support relationship between the mode select (I3C mode / I^{2}C mode) and operation mode (Master / Slave) on the I3C bus or the I^{2}C bus is shown in Table 33.9.

Table 33.9 Support of operating mode

I3C/I^{2}C Bus	I3C mode		I^{2}C mode	
	Master	Slave	Master	Slave
I3C Bus	✓	✓	—	✓
I^{2}C Bus	—	—	✓	✓

Note: ✓: Supported

—: Un-Supported

33.3.2.1.1 Master Mode Operation

(1) I2C Master Operation

(a) Data Write Transfer (Single Buffer transfer)

In master transmit operation, I3C outputs the SCL clock and transmitted data signals as the master device, and the slave device returns acknowledgments. Figure 33.143 shows an example of usage of master transmission and Figure 33.6 to Figure 33.8 show the timing of operations in master transmission.

The following describes the procedure and operations for master transmission.

1. Initial settings. For details, see section 33.3.3.1. Initial Setting Flow.
2. Read the BCST.BFREF flag to check that the bus is open, and then set the CNDCTL.STCND bit to 1 (START condition issuance request). Upon receiving the request, I3C issues a START condition. At the same time, the BFREF flag bit is automatically set to 0, the BST.STCNDDF flag is automatically set to 1 and the STCND bit is automatically set to 0. At this time, if the START condition is detected and the internal levels for the SDA output state and the levels on the I3C_SDA line have matched while the STCND bit = 1, I3C recognizes that issuing of the START condition as requested by the STCND bit has been successfully completed, and bits CRMS and TRMD in the PRSST register are automatically set to 1, placing I3C in master transmit mode. The NTST.TDBEF0 flag is also automatically set to 1 in response to setting of the TRMD bit to 1.
3. Check that the NTST.TDBEF0 flag = 1, and then write the value for transmission (the slave address and the R/W# bit) to the NTDTBP0 register. Once the data for transmission are written to the NTDTBP0 register, the TDBEF0 flag is automatically set to 0, the data are transferred from the Normal Tx Data Buffer to the Shift Register, and the TDBEF0 flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRMD bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, I3C continues in master transmit mode. Because the BST.NACKDF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the CNDCTL.SPCND bit to issue a STOP condition. For data transmission with an address in the 10-bit format, start by writing 1111 0, the 2 higher-order bits of the slave address, and W to the NTDTBP0 register as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to the NTDTBP0 register.
4. After confirming that the NTST.TDBEF0 flag = 1, write the data for transmission to the NTDTBP0 register. I3C automatically holds the I3C_SCL line low until the data for transmission are ready or a STOP condition is issued.
5. After all bytes of data for transmission have been written to the NTDTBP0 register, wait until the value of the BST.TENDF flag returns to 1, and then set the CNDCTL.SPCND bit to 1 (STOP condition issuance request). Upon receiving a STOP condition issuance request, I3C issues the STOP condition.
6. Upon detecting the STOP condition, I3C automatically sets bits CRMS and TRMD in the PRSST register to 00 and enters slave receive mode. Furthermore, it automatically sets the TDBEF0 and TENDF flags to 0, and sets the BST.SPCNDDF flag to 1.
7. After checking that the BST.SPCNDDF flag = 1, set the BST.NACKDF and SPCNDDF flags to 0 for the next transfer operation.

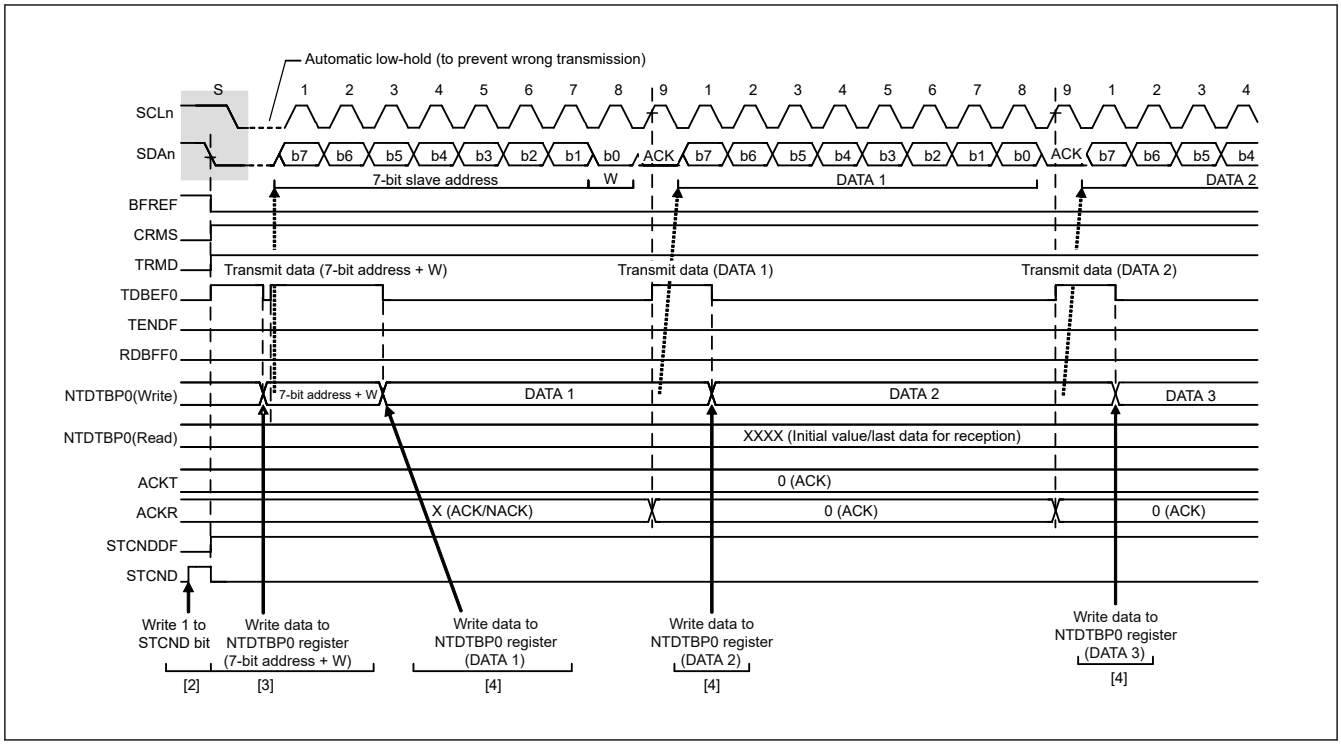


Figure 33.6 Master transmit operation timing (1) (7-bit address format)

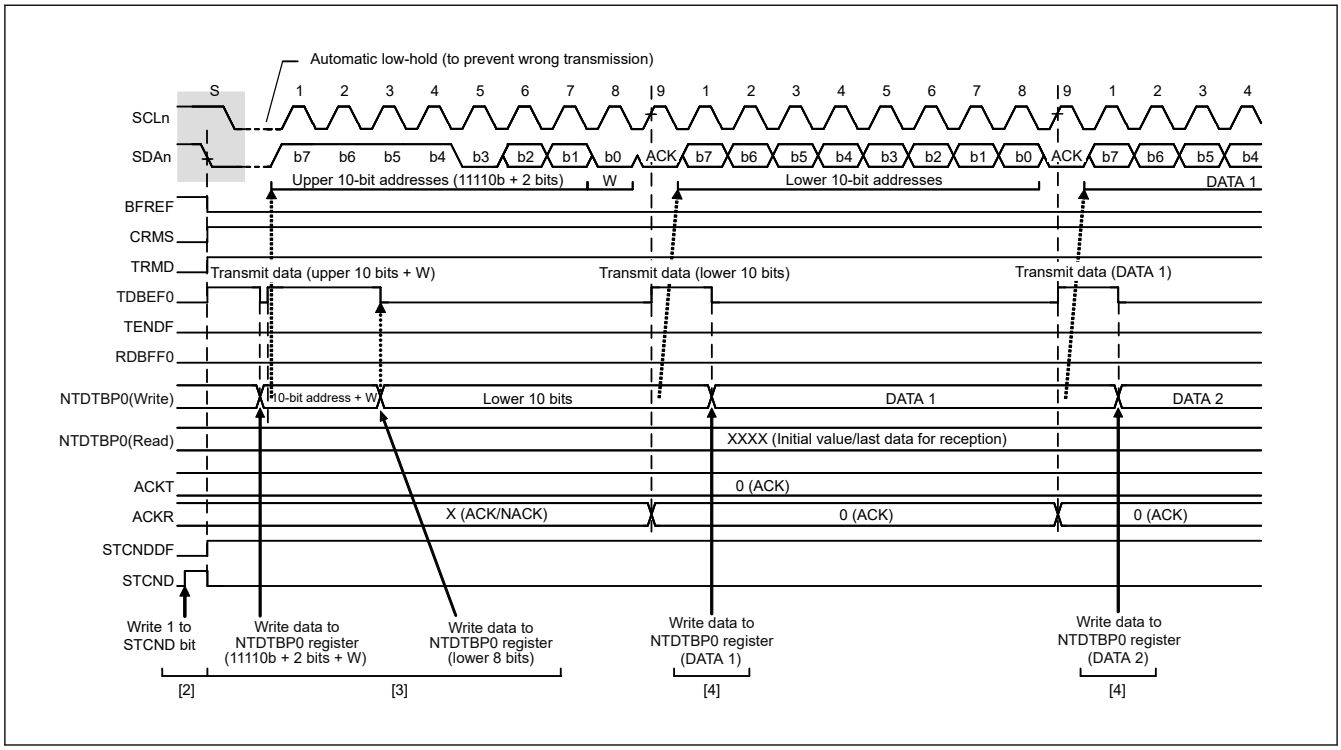


Figure 33.7 Master transmit operation timing (2) (10-bit address format)

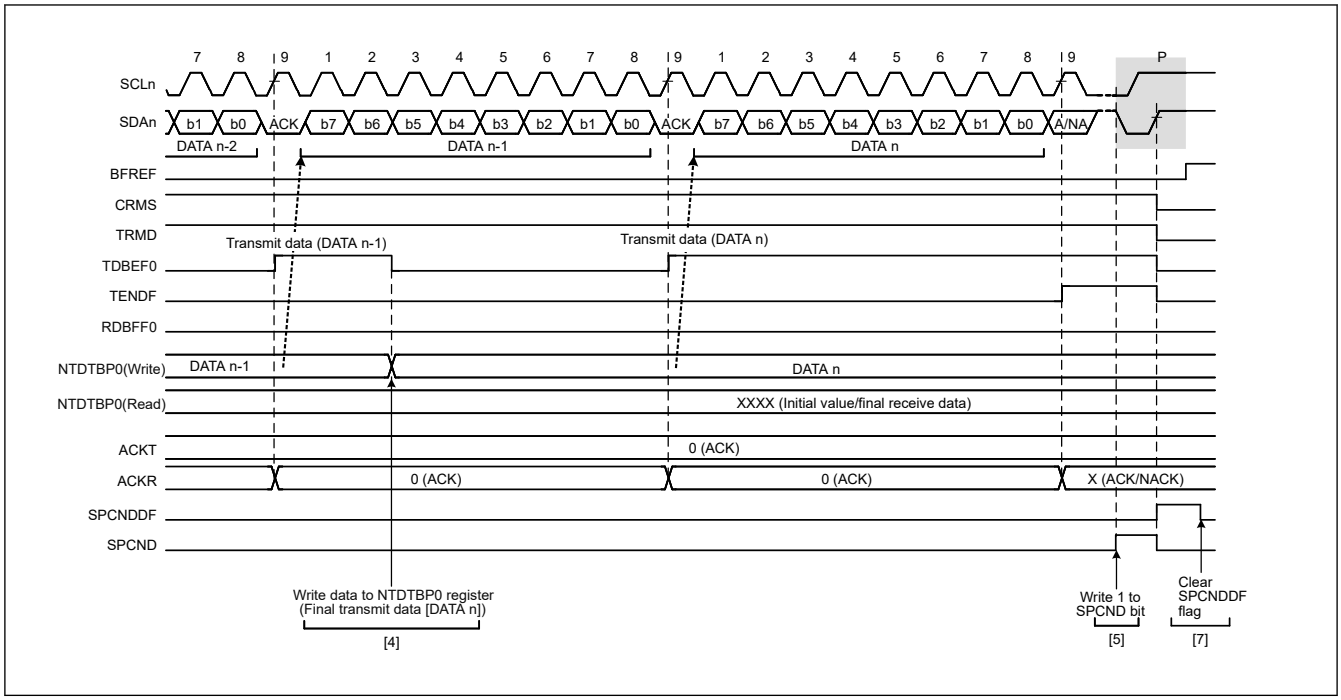


Figure 33.8 Master transmit operation timing (3)

(b) Data Read Transfer (Single Buffer transfer)

In master receive operation, I3C as a master device outputs the SCL clock, receives data from the slave device, and returns acknowledgments. Because I3C must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.

Figure 33.144 and Figure 33.145 show examples of usage of master reception (7-bit address format) and Figure 33.9 to Figure 33.11 show the timing of operations in master reception.

The following describes the procedure and operations for master reception.

1. Initial settings. For details, see section 33.3.3.1. Initial Setting Flow.
2. Read the BCST.BFREF flag to check that the bus is open, and then set the CNDCTL.STCND bit to 1 (START condition issuance request). Upon receiving the request, I3C issues a START condition. When I3C detects the START condition, the BFREF flag is automatically set to 0 and the BST.STCNDDF flag is automatically set to 1 and the STCND bit is automatically set to 0. At this time, if the START condition is detected and the levels for the SDA output and the levels on the I3C_SDA line have matched while the STCND bit = 1, I3C recognizes that issuing of the START condition as requested by the STCND bit has been successfully completed, and bits CRMS and TRMD in the PRSST register are automatically set to 1, placing I3C in master transmit mode. The NTST.TDBEF0 flag is also automatically set to 1 in response to setting of the TRMD bit to 1.
3. Check that the NTST.TDBEF0 flag = 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to the NTDTBP0 register. Once the data for transmission are written to the NTDTBP0 register, the TDBEF0 flag is automatically set to 0, the data are transferred from the Normal Tx Data Buffer to the Shift Register, and the TDBEF0 flag is again set to 1. Once the byte containing the slave address and R/W# bit has been transmitted, the value of the PRSST.TRMD bit is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the TRMD bit is set to 0 on the rising edge of the ninth cycle of SCL clock, placing I3C in master receive mode. At this time, the TDBEF0 flag is set to 0. The NTST.RDBFF0 flag is automatically set to 1 when ACK response is received from the slave device. If the slave device is not recognized or a communication failure occurs, the BST.NACKDF flag will be set to 1. At this time, set 1 to the CNDCTL.SPCND bit to issue a STOP condition. For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue a Repeated START condition. After that, transmitting 1111 0, the two higher-order bits of the slave address, and the R bit places I3C in master receive mode.
4. Dummy read the NTDTBP0 register after confirming that the NTST.RDBFF0 flag = 1; this makes I3C start output of the SCL clock and start data reception.
5. After 1 byte of data has been received, the NTST.RDBFF0 flag is set to 1 on the rising edge of the eighth or ninth cycle of SCL clock (the clock signal) as selected by the SCSTRCTL.ACKTWE bit. Reading the NTDTBP0 register at

this time will produce the received data, and the RDBFF0 flag is automatically set to 0 at the same time. Furthermore, the value of the acknowledgment field received during the ninth cycle of SCL clock is returned as the value set in the ACKCTL.ACKT bit. Furthermore, if the next byte to be received is the next to last byte, set the SCSTRCTL.RWE bit to 1 (for wait insertion) before reading the NTDTBP0 register (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the ACKCTL.ACKT bit to 1 (NACK) in step 6, due to other interrupts, etc., this fixes the I3C_SCL line to the low level on the falling edge of the ninth clock cycle in reception of the last byte, so the state is such that issuing a STOP condition is possible.

6. When the SCSTRCTL.ACKTWE bit = 0 and the slave device must be notified that it is to end transfer for data reception after transfer of the next (final) byte, set the ACKCTL.ACKT bit to 1 (NACK).
7. After reading the byte before last from the NTDTBP0 register, if the value of the NTST.RDBFF0 flag is confirmed to be 1, write 1 to the CNDCTL.SPCND bit (STOP condition issuance request) and then read the last byte from the NTDTBP0 register. When 1 is written to the CNDCTL.SPCND bit, I3C is released from the wait state and issues the STOP condition after low-level output in the ninth clock cycle is completed or the I3C_SCL line is released from the low-hold state.
8. Upon detecting the STOP condition, I3C automatically sets bits CRMS and TRMD in the PRSST register to 00 and enters slave receive mode. Furthermore, detection of the STOP condition leads to setting of the BST.SPCNDDF flag to 1.
9. After checking that the BST.SPCNDDF flag = 1, set the BST.NACKDF and SPCNDDF flags to 0 for the next transfer operation.

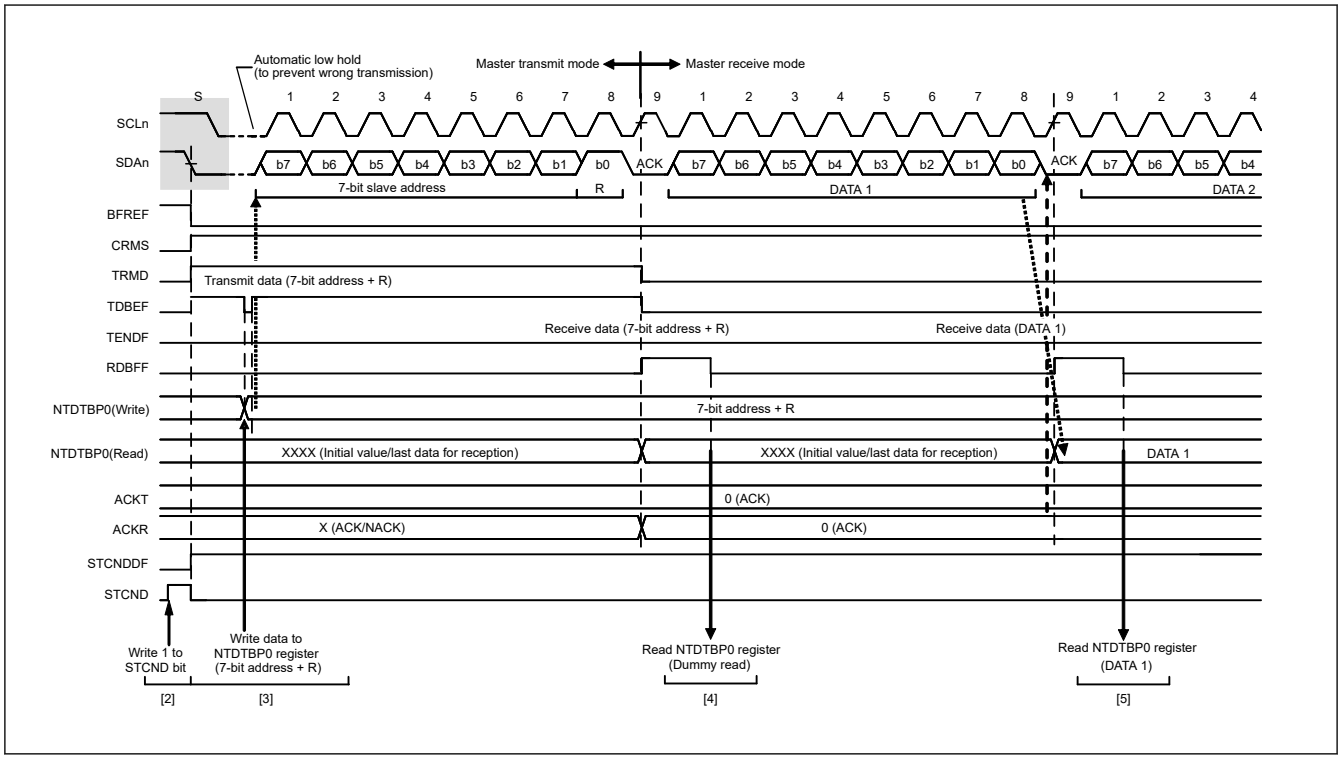


Figure 33.9 Master receive operation timing (1) (7-bit address format, when ACKTWE = 0)

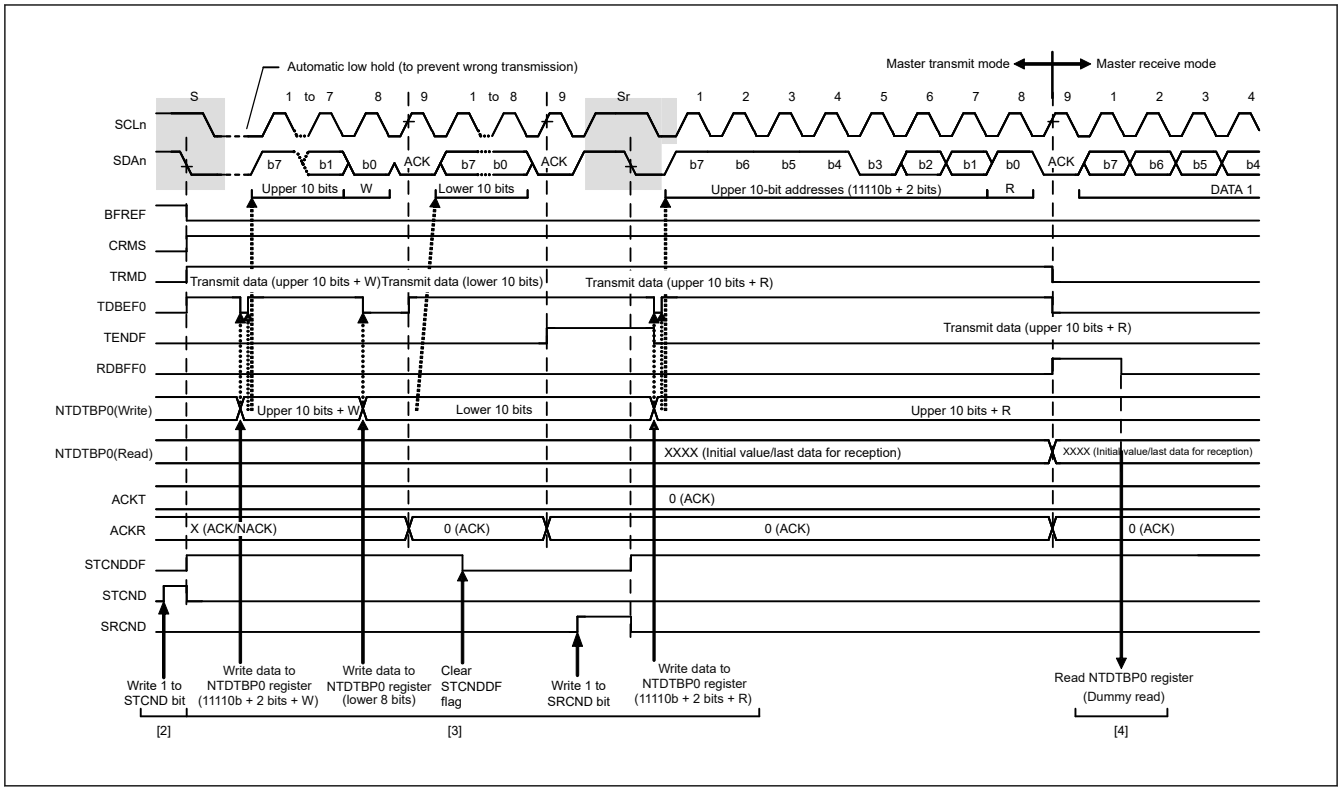


Figure 33.10 Master receive operation timing (2) (10-bit address format, when ACKTWE = 0)

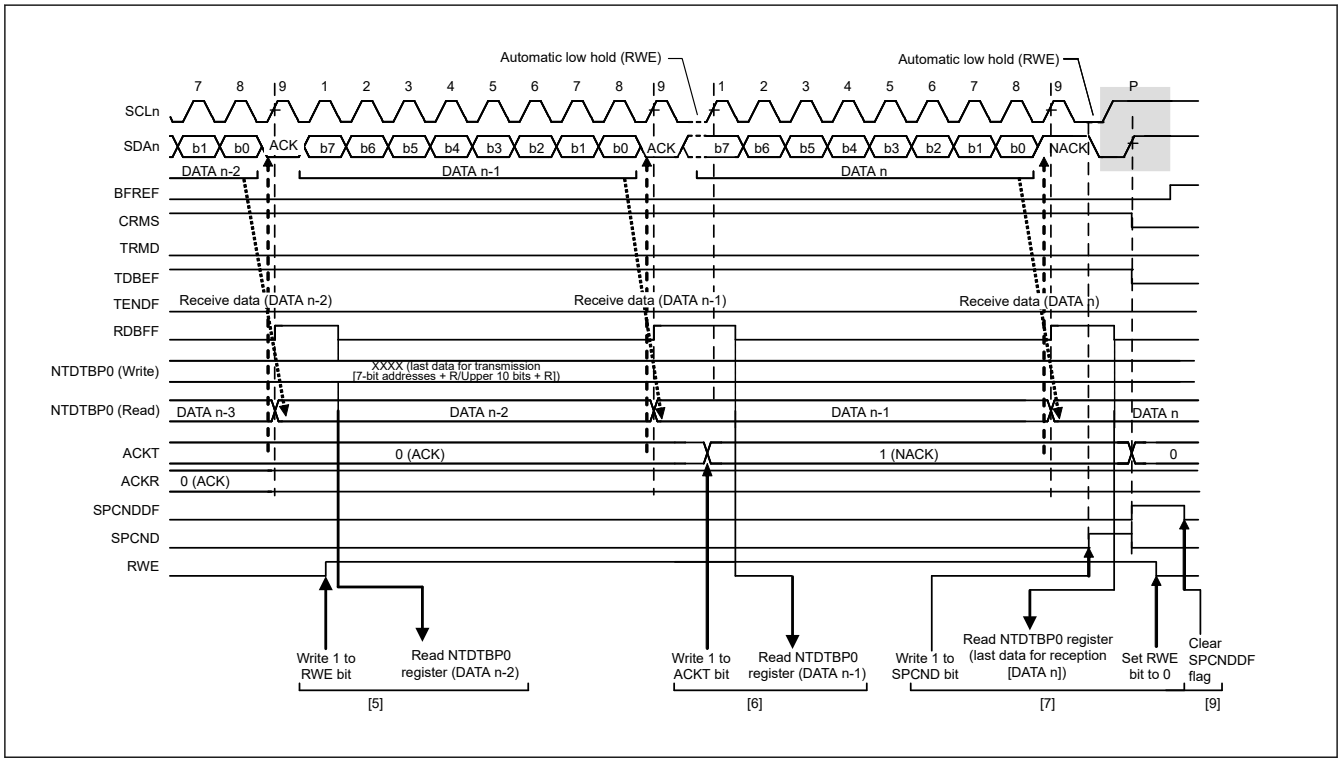


Figure 33.11 Master receive operation timing (3) (when ACKTWE = 0)

(2) I3C Master Operation

(a) Dynamic Address Assign Procedure

After initializing I3C, first execute Dynamic Address Assign Procedure for I3C Slave connected on the I3C Bus. The following describes the procedure.

1. Initial setting (see section 33.3.3.1.2. I3C Initial Setting Flow for details)
2. Execute Dynamic Address Assign with ENTDAA or SETDASA Common Command Code (CCC) for I3C Slave set in DAT (DATBASm register).
   Write Command Descriptor (Address Assign Command) to Command Buffer via the NCMDQP register.
3. When Command Descriptor is written to Command Buffer, Transaction is issued on I3C Bus.
4. When ENTDAA is specified for CMD[7:0] of Address Assign Command:
   Execute Dynamic Address Assign for I3C Slave for the number of DATs specified by DEV_COUNT[3:0] starting with DAT specified by DEV_INDEX[4:0] of Address Assign Command.
   When SETDASA is specified for CMD[7:0] of Address Assign Command:
   Execute Dynamic Address Assign for I3C Slave indicated by DAT specified by DEV_INDEX[4:0] of Address Assign Command.
5. In case of ENTDAA, the Provisional ID, BCR, DCR transmitted from I3C Slave is stored in Receive Data Buffer (BCR is also automatically stored in the MSDCTm register).
   Read the Provisional ID, BCR, and DCR from the Receive Data Buffer via the NTDTBPn register with an interrupt by RDBFF0 = 1.
6. When execution of Dynamic Address Assign is completed, issue STOP condition and store the Response Descriptor into the Response Buffer.
7. Read the Response Descriptor via the NRSPQP register and check the status.
8. Check whether the value of the DATA_LENGTH[15:0] bits of the Response Descriptor matches the value of DEV_COUNT[3:0] of the Address Assign Command.

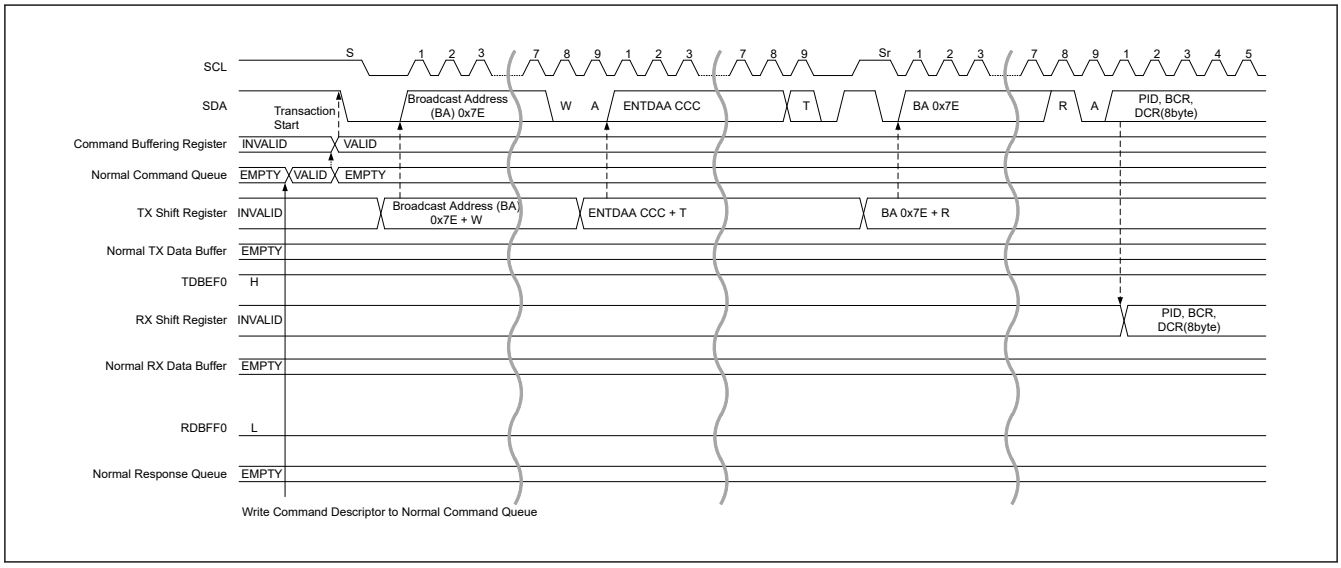


Figure 33.12 Dynamic address assign procedure (ENTDAA CCC) timing (1/3)

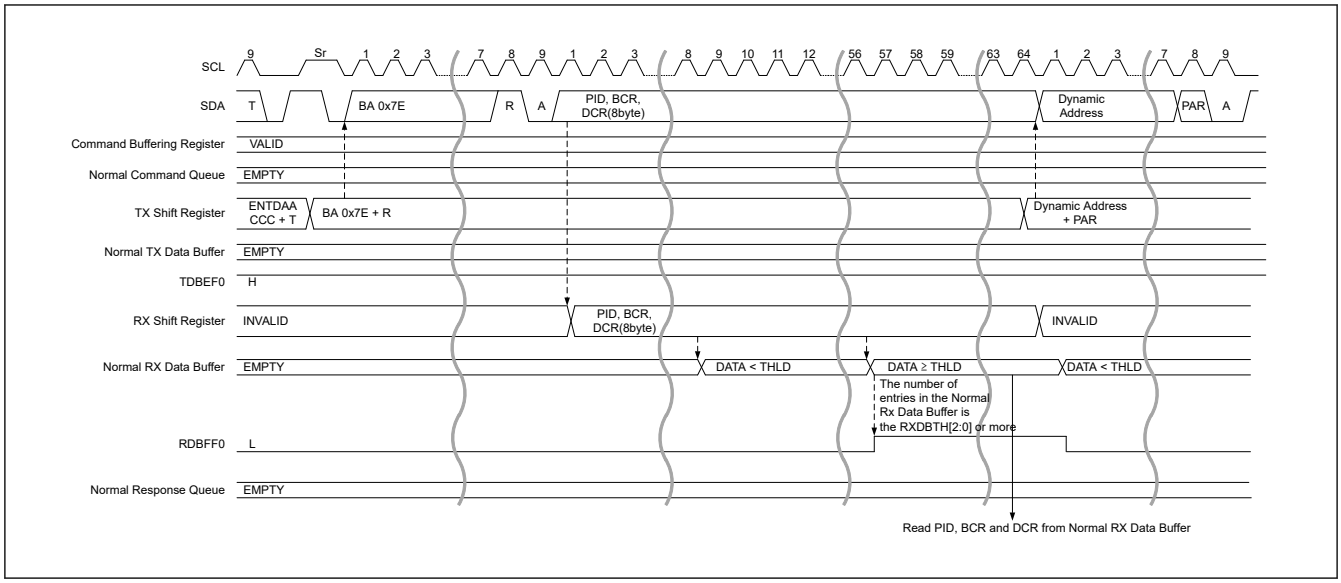


Figure 33.13 Dynamic address assign procedure (ENTDAA CCC) timing (2/3)

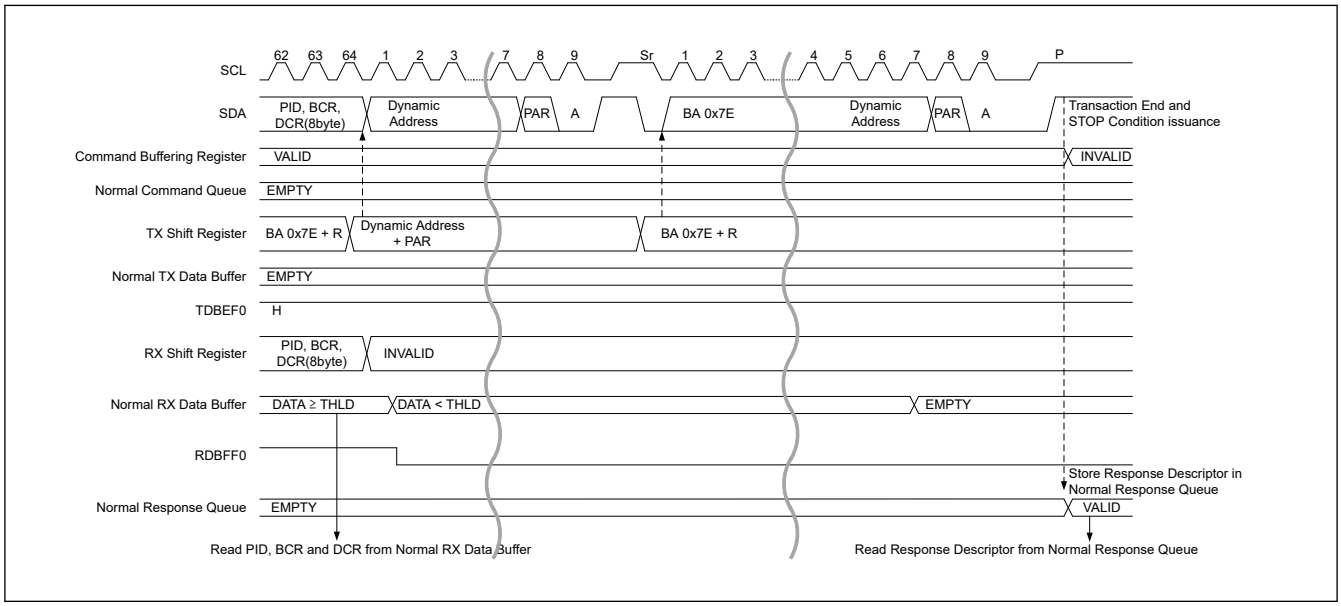


Figure 33.14 Dynamic address assign procedure (ENTDAA CCC) timing (3/3)

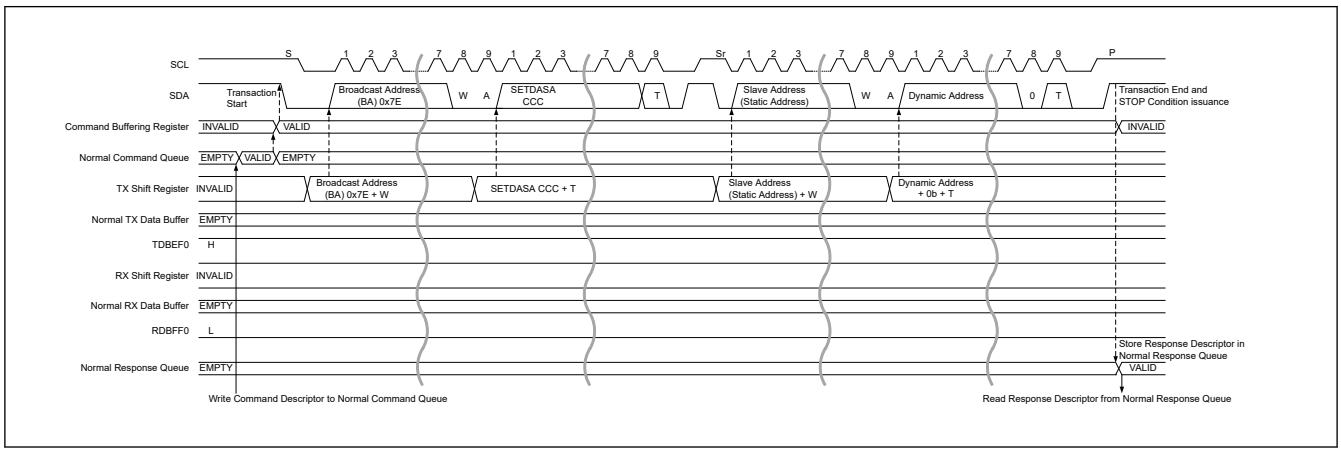


Figure 33.15 Dynamic address assign procedure (SETDASA CCC) timing

(b) SDR Data Write Transfer

1. Write data for transmission to the Tx Data Buffer via the NTDTBPn register.
2. Write a Command Descriptor (Immediate Transfer Command or Regular Transfer Command or Combo Transfer Command) for Data Transfer to the Command Buffer via the NCMDQP register.
3. When Command Descriptor is written to Command Buffer, transaction is issued on I3C Bus.
   When NACK is received with the Address Header, transaction of the same command is automatically issued according to the NACK Retry Count value (DATBASm.DVNACK) of DAT.
4. If data for transmission still remain, write data for transmission by an interrupt with TDBEF0 = 1 to the Tx Data Buffer via the NTDTBPn register.
5. When data transmission for the number of Data Length specified by the DATA_LENGTH[15:0] bits of the Command Descriptor is completed, the Repeated START condition or STOP condition is issued and the Response Descriptor is stored in the Response Buffer.
6. Read the Response Descriptor via the NRSPQP register and check the status.
7. Check that the value of the DATA_LENGTH[15:0] bits of the Response Descriptor is 0.

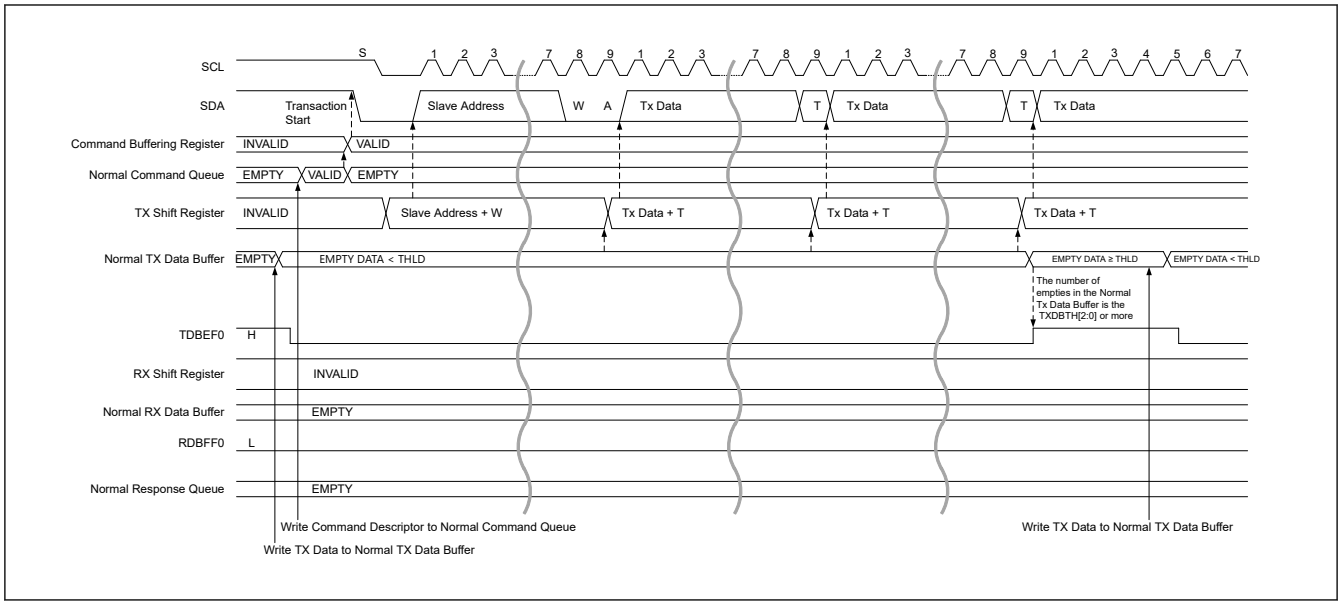


Figure 33.16 SDR data write transfer timing (1/2)

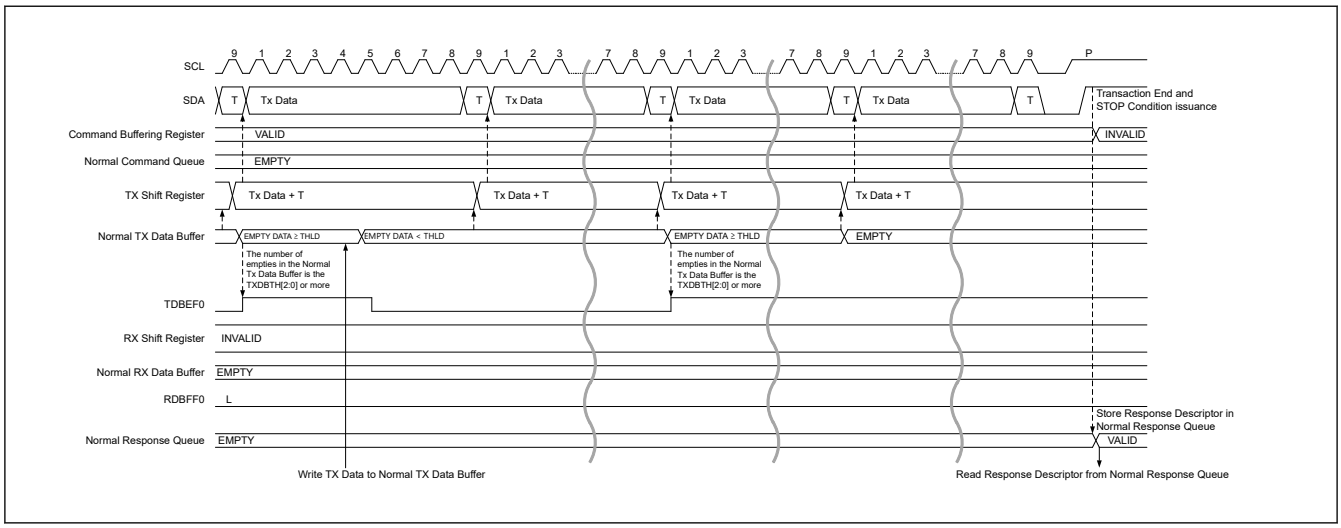


Figure 33.17 SDR data write transfer timing (2/2)

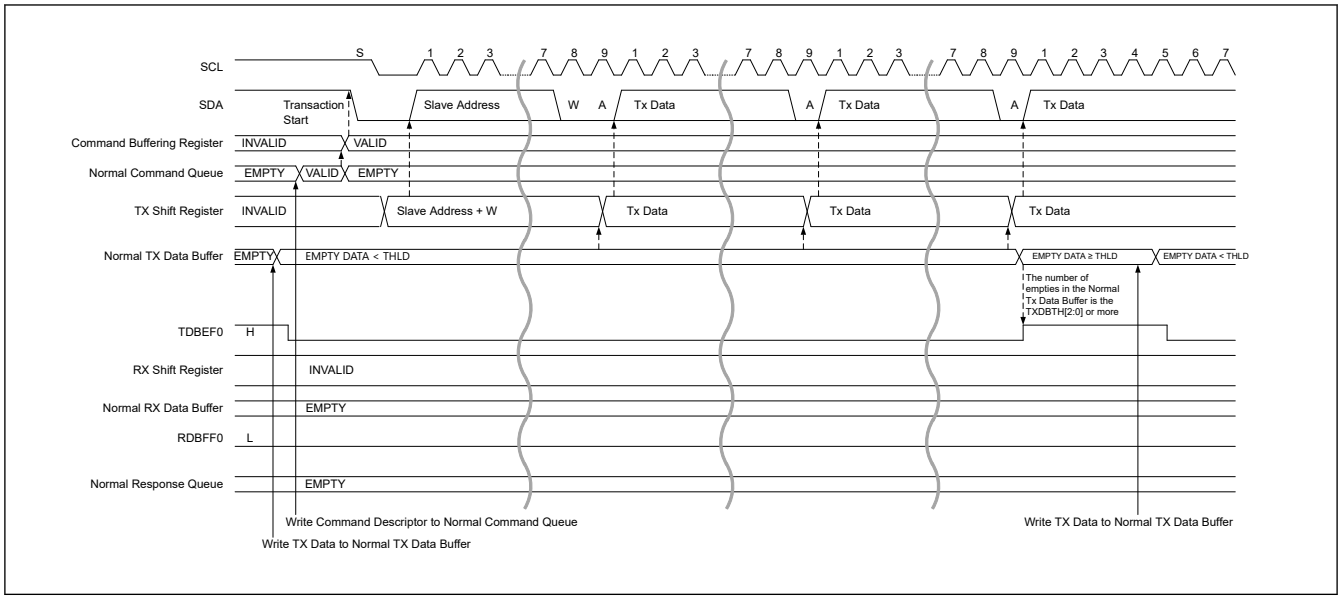


Figure 33.18 Legacy I²C message data write timing (1/2)

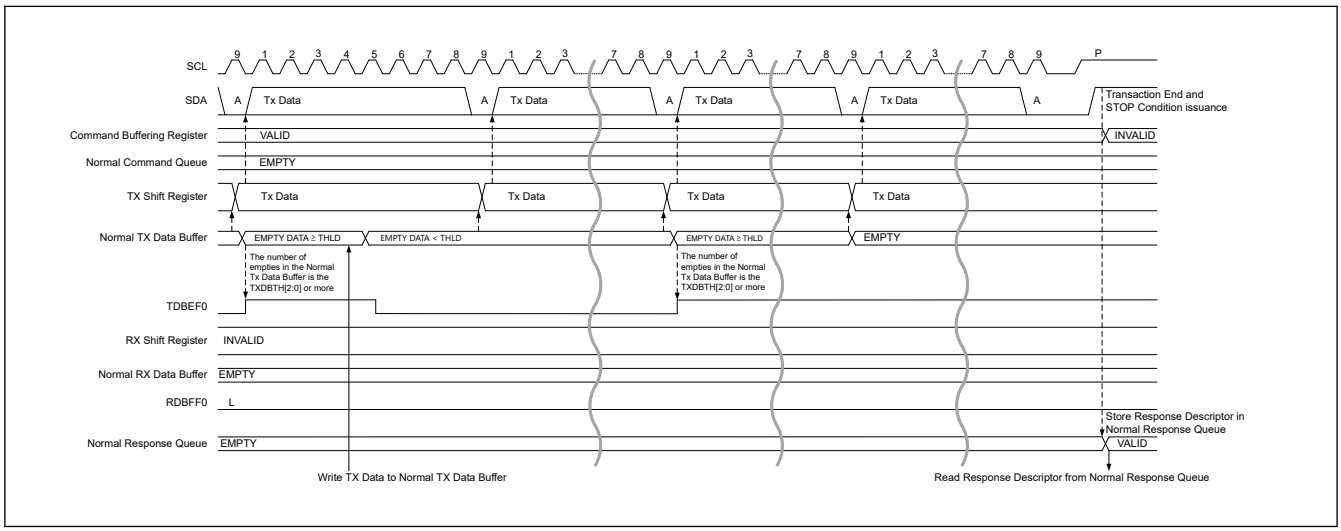


Figure 33.19 Legacy I²C message data write timing (2/2)

(c) SDR Data Read Transfer

1. Write a Command Descriptor (Immediate Transfer Command or Regular Transfer Command or Combo Transfer Command) for Data Transfer to the Command Buffer via the NCMDQP register.
2. When Command Descriptor is written to Command Buffer, transaction is issued on I3C Bus.
   When NACK is received with the Address Header, transaction of the same command is automatically issued according to the NACK Retry Count value (DATBASm.DVNACK) of DAT.
3. Data received from the I3C Slave is stored in the Receive Data Buffer.
4. With the RDBFF0 = 1 interrupt, the received data is read from the Receive Data Buffer via the NTDTBPn register.
5. SDR:
   Detecting Low in T-bit or receiving Data for the number of Data Length specified by the DATA_LENGTH[15:0] bits of Command Descriptor is completed, issue Repeated START condition or STOP condition and store the Response Descriptor into the Response Buffer.
   Legacy I²C Message:
   When data reception for the number of Data Length specified by the DATA_LENGTH[15:0] bits of Command Descriptor is completed, NACK is issued. After that, issue a Repeated START condition or STOP condition and store the Response Descriptor into the Response Buffer.
6. Read the Response Descriptor via the NRSPQP register and check the status.

7. Check whether the value of the DATA_LENGTH[15:0] bits of the Response Descriptor matches the data length setting value of the Command Descriptor.

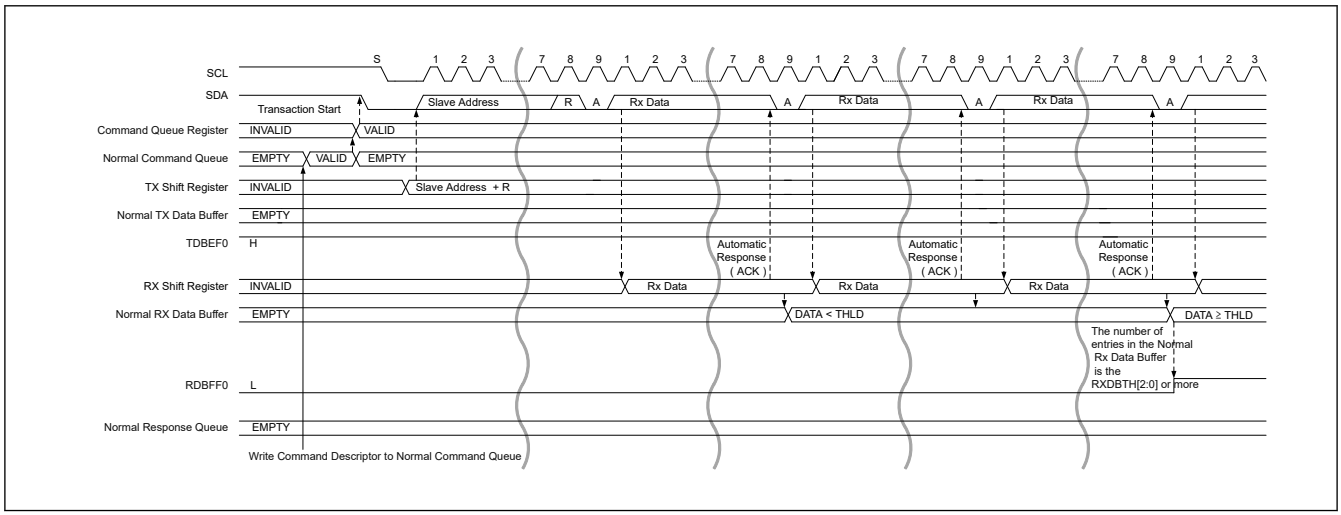


Figure 33.20 SDR data read transfer timing (1/2)

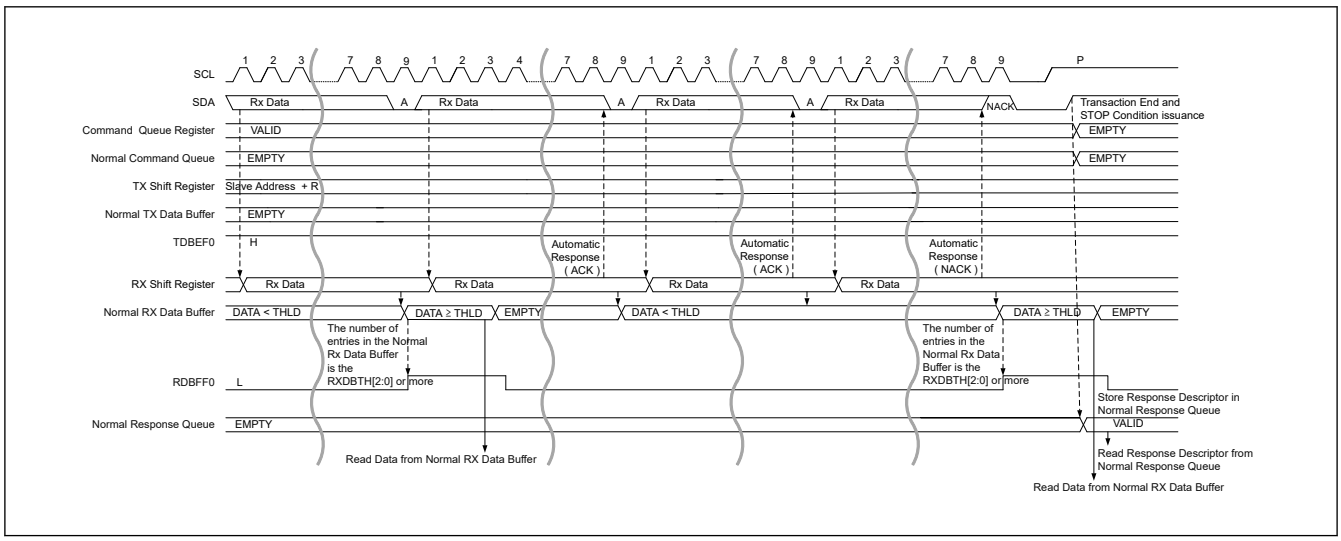


Figure 33.21 SDR data read transfer timing (2/2)

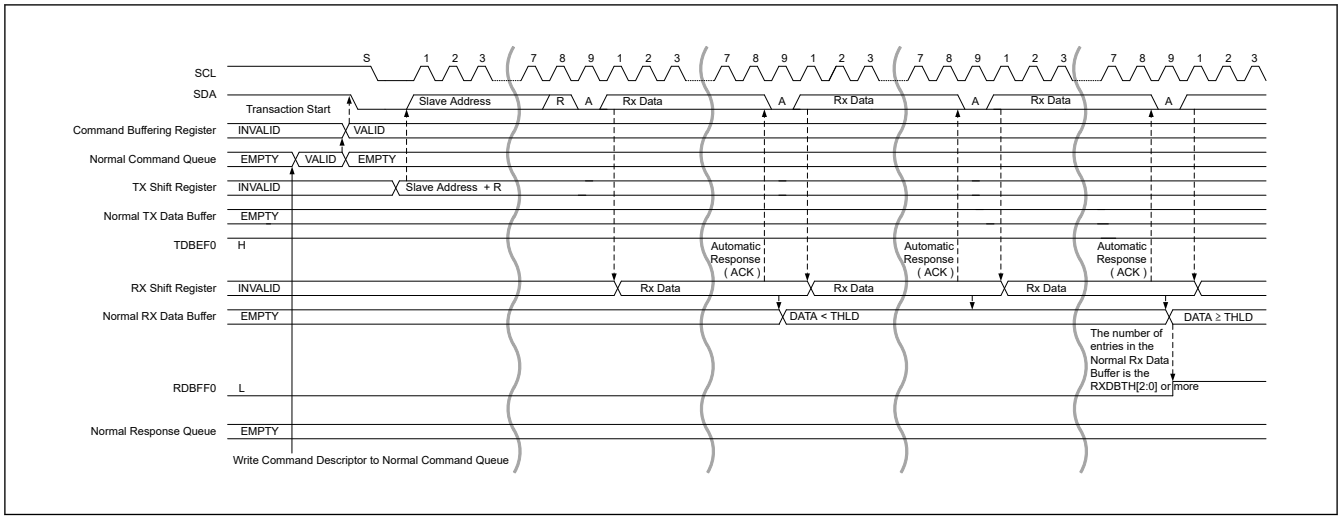


Figure 33.22 Legacy I²C message data read transfer timing (1/2)

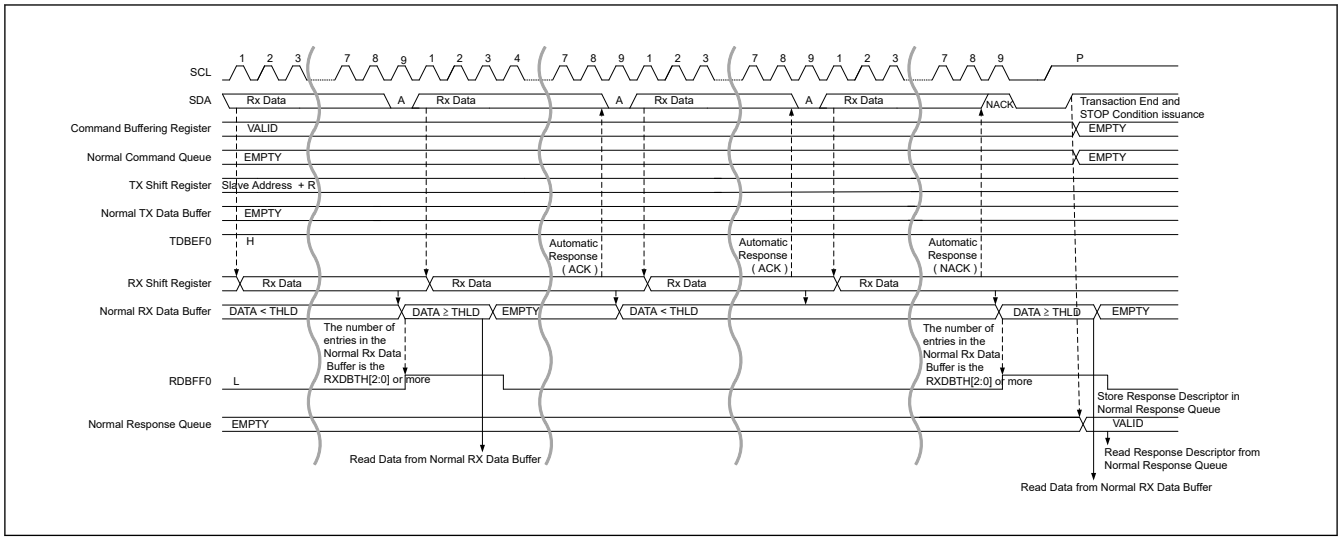


Figure 33.23 Legacy I²C message data read transfer timing (2/2)

(d) HDR Data Write Transfer

1. Write data for transmission to the Tx Data Buffer via the NTDTBPn register.
2. Write a Command Descriptor (Immediate Transfer Command or Regular Transfer Command or Combo Transfer Command) for Data Transfer to the Command Buffer via the NCMDQP register.
3. When Command Descriptor is written to Command Buffer, transaction is issued on I3C Bus.
   When NACK is received with the Address Header, transaction of the same command is automatically issued according to the NACK Retry Count value (DATBASm.DVNACK) of DAT.
4. If data for transmission still remains, write data for transmission by an interrupt with TDBEF0 = 1 to the Tx Data Buffer via the NTDTBPn register.
5. HDR-DDR:
   When the data transmission for the number of Data Length specified by the DATA_LENGTH[15:0] bits of the Command Descriptor is completed, CRC Word is transmitted. After that, issue HDR Restart Pattern or HDR Exit Pattern + STOP condition and store the Response Descriptor into Response Buffer.
   HDR-Ternary:
   When the data transmission for the number of Data Length specified by the DATA_LENGTH[15:0] bits of the Command Descriptor is completed, the HDR Restart Pattern or HDR Exit Pattern + STOP condition is issued and the Response Descriptor is stored in the Response Buffer.
6. Read the Response Descriptor via the NRSPQP register and check the status.
7. Check that the value of the DATA_LENGTH[15:0] bits of the Response Descriptor is 0.

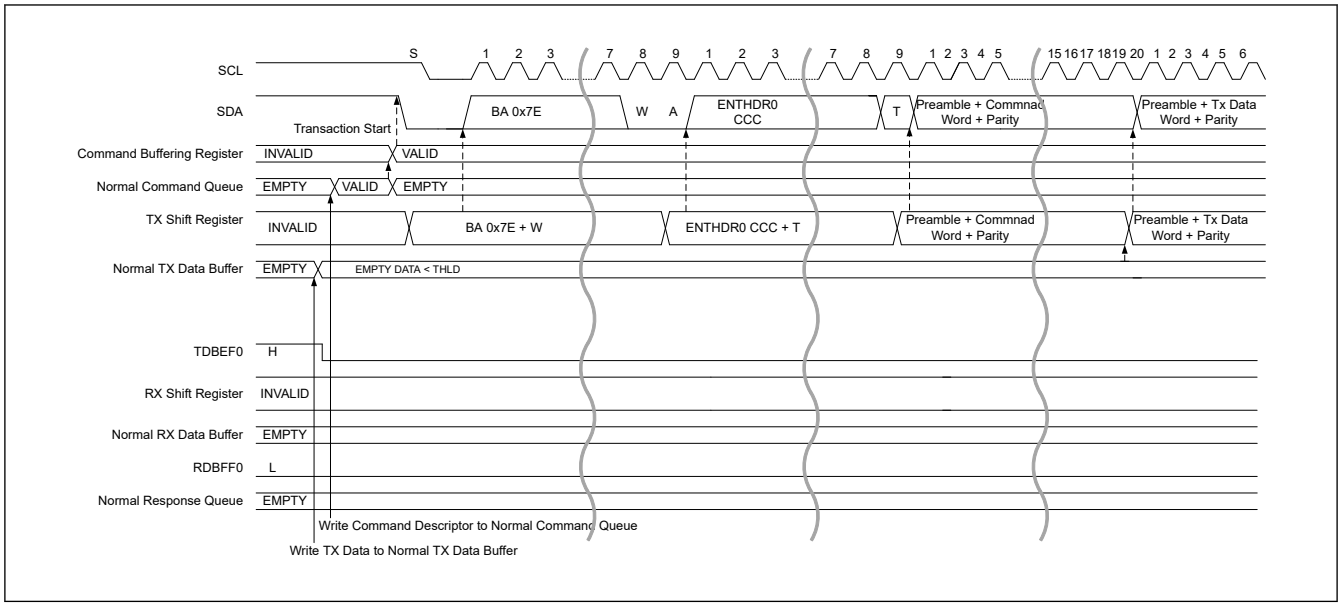


Figure 33.24 HDR data write transfer (HDR-DDR) timing (1/4)

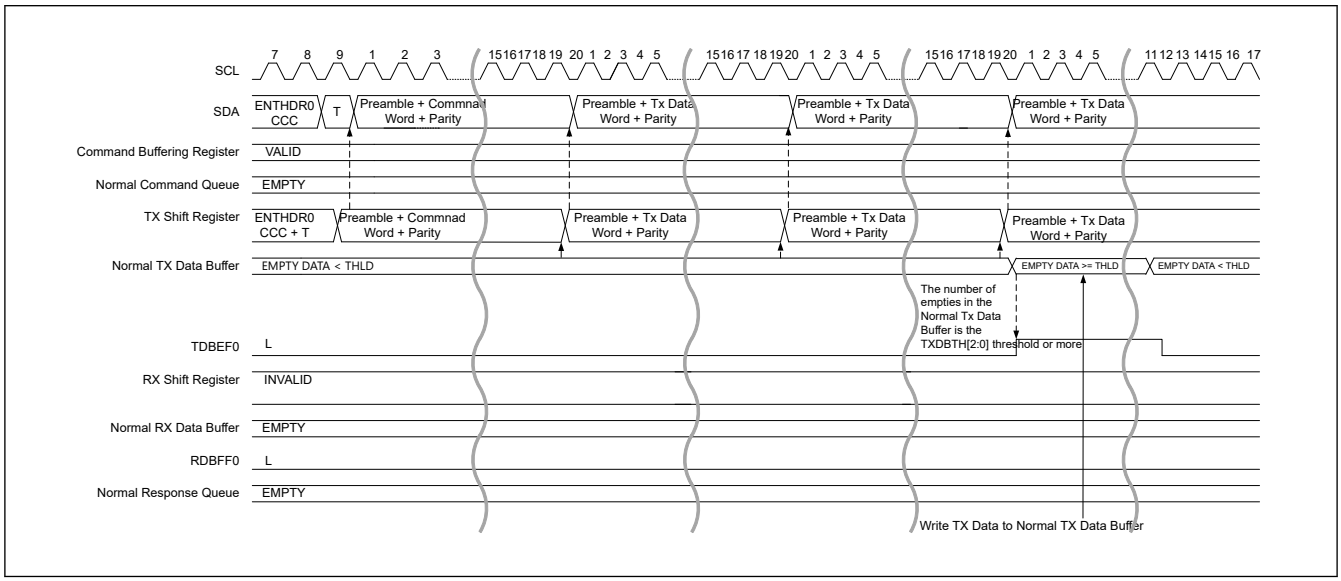


Figure 33.25 HDR data write transfer (HDR-DDR) timing (2/4)

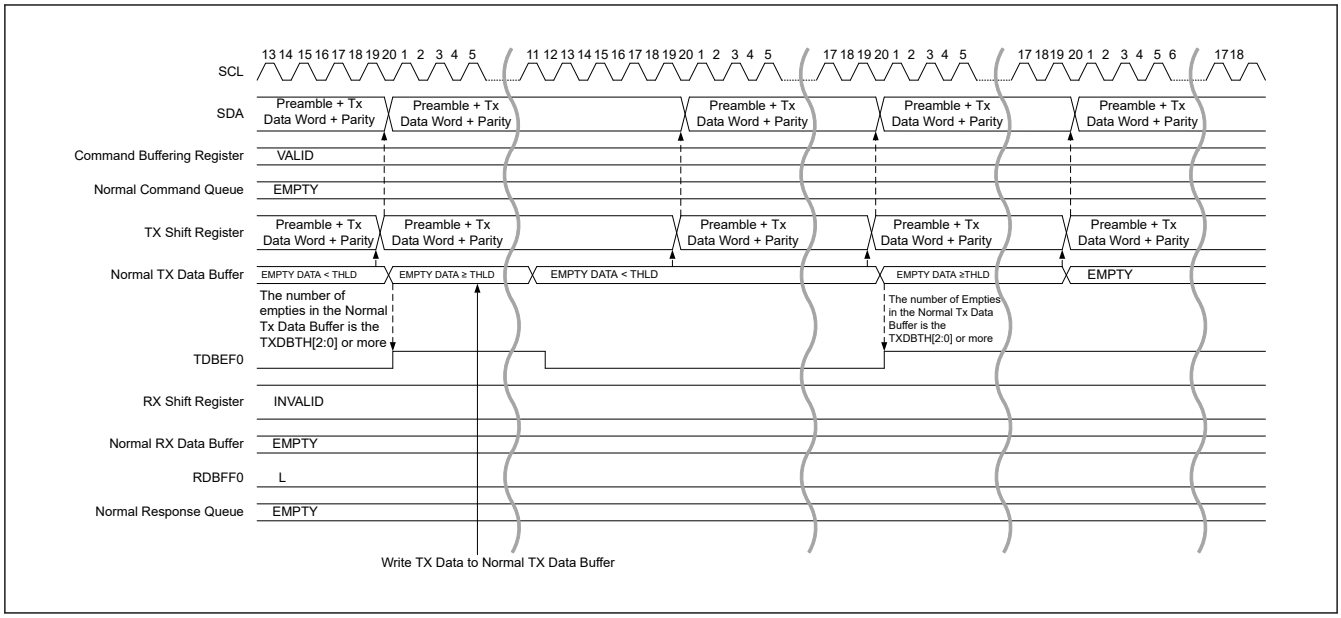


Figure 33.26 HDR data write transfer (HDR-DDR) timing (3/4)

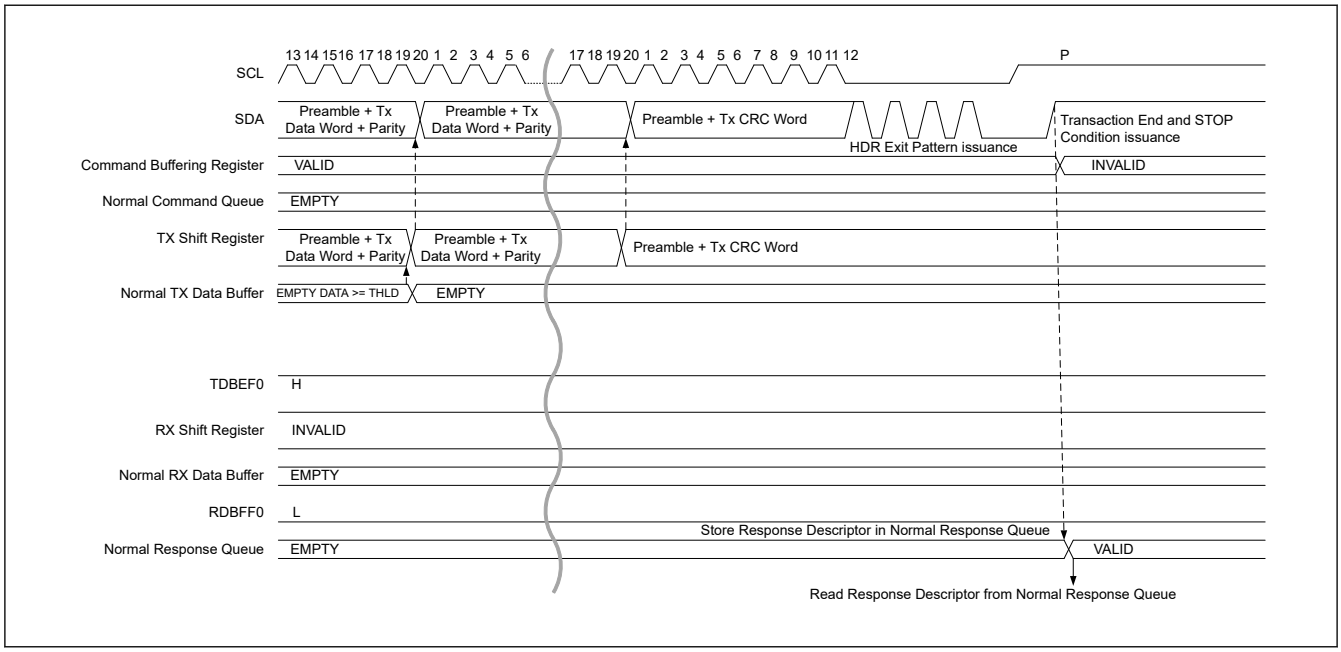


Figure 33.27 HDR data write transfer (HDR-DDR) timing (4/4)

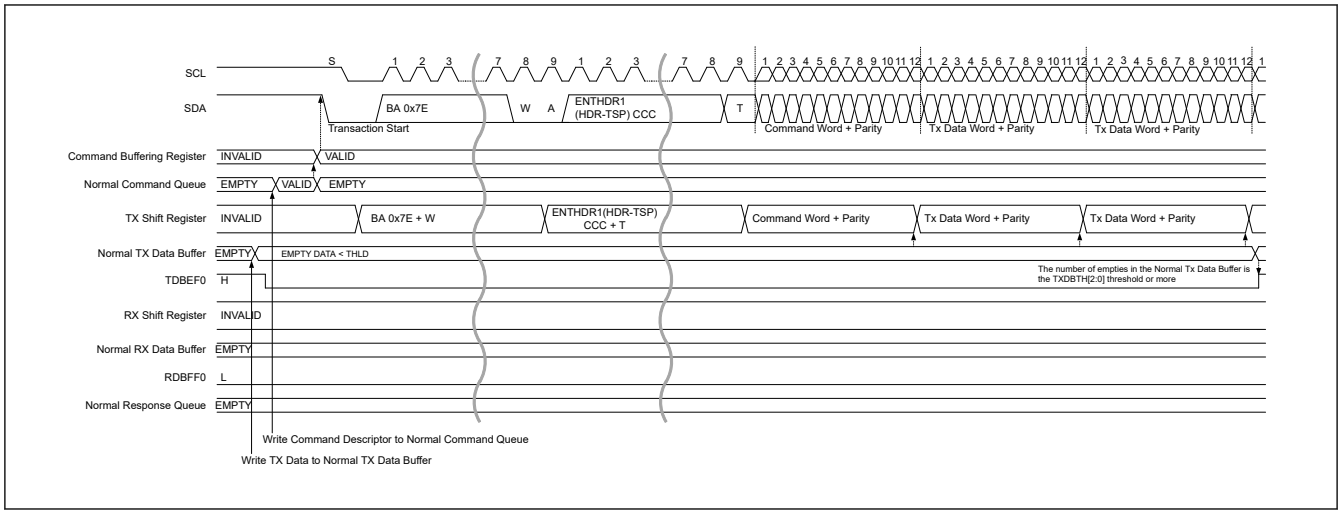


Figure 33.28 HDR data write transfer (HDR-TSP, TSL) timing (1/2)

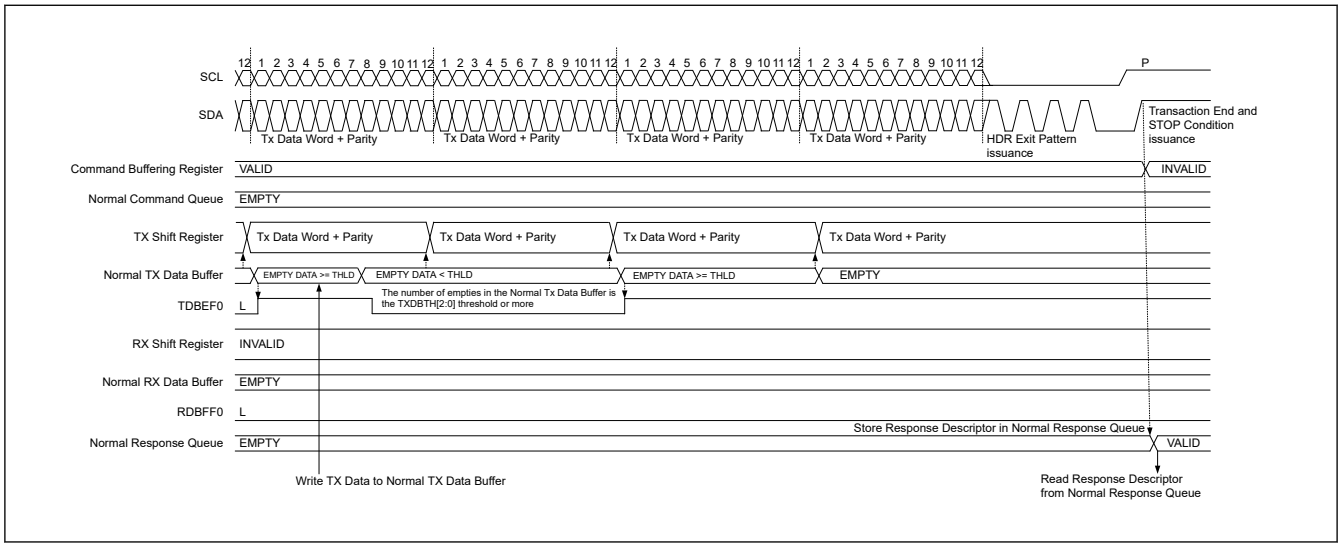


Figure 33.29 HDR data write transfer (HDR-TSP, TSL) timing (2/2)

(e) HDR Data Read Transfer

1. Write a Command Descriptor (Immediate Transfer Command or Regular Transfer Command or Combo Transfer Command) for Data Transfer to the Command Buffer via the NCMDQP register.
2. When Command Descriptor is written to Command Buffer, transaction is issued on I3C Bus.
   When NACK is received with the Address Header, transaction of the same command is automatically issued according to the NACK Retry Count value (DATBASm.DVNACK) of DAT.
3. Data received from the I3C Slave is stored in the Receive Data Buffer.
4. With the RDBFF0 = 1 interrupt, the received data is read from the Receive Data Buffer via the NTDTBPn register.
5. HDR-DDR:
   (a) When CRC Word is received, it issues HDR Restart Pattern or HDR Exit Pattern + STOP condition and stores the Response Descriptor into the Response Buffer.
   (b) When the preamble following the received data for the number of Data Length specified by the DATA_LENGTH[15:0] bits of Command Descriptor indicates Data Word (11), drive (Abort) low to the second bit of the preamble. After that, issue HDR Restart Pattern or HDR Exit Pattern + STOP condition and store the Response Descriptor into the Response Buffer.

   HDR-Ternary:
   When detecting the start of the HDR Restart Pattern or HDR Exit Pattern from I3C Slave, issue HDR Restart Pattern or HDR Exit Pattern + STOP condition and store the Response Descriptor into the Response Buffer.

6. Read the Response Descriptor via the NRSPQP register and check the status.
7. Check whether the value of the DATA_LENGTH[15:0] bits of the Response Descriptor matches the data length setting value of the Command Descriptor.

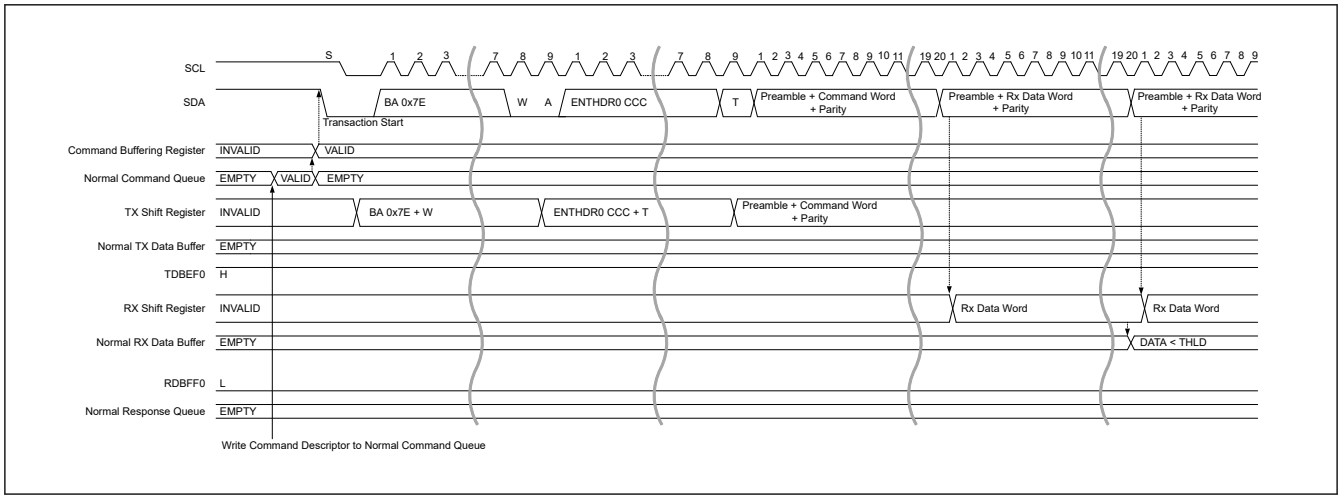


Figure 33.30 HDR data read transfer (HDR-DDR) timing (1/2)

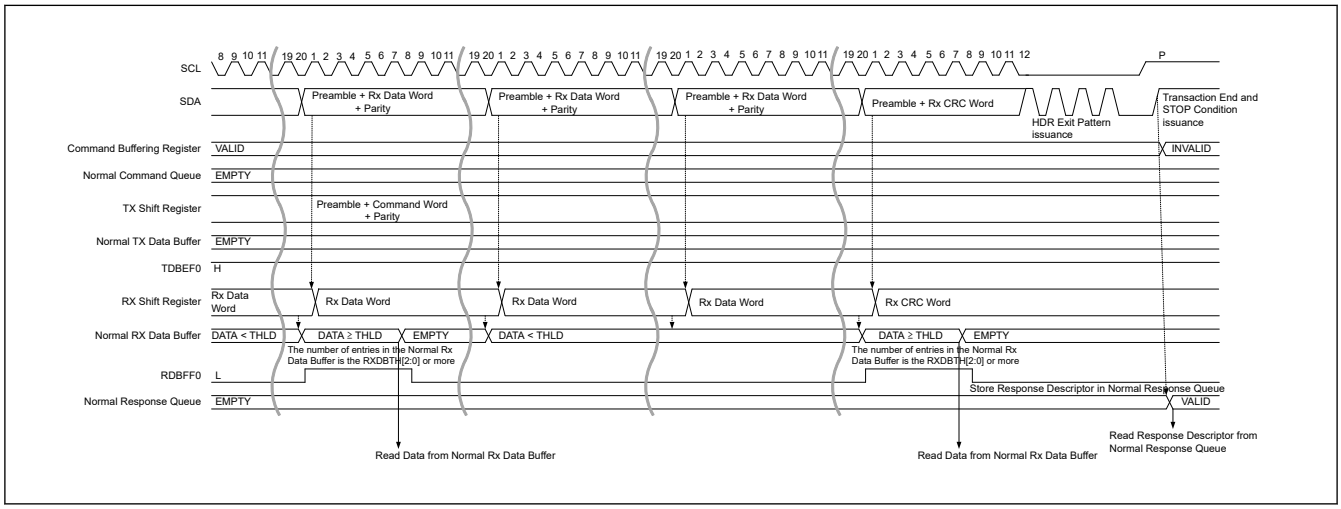


Figure 33.31 HDR data read transfer (HDR-DDR) timing (2/2)

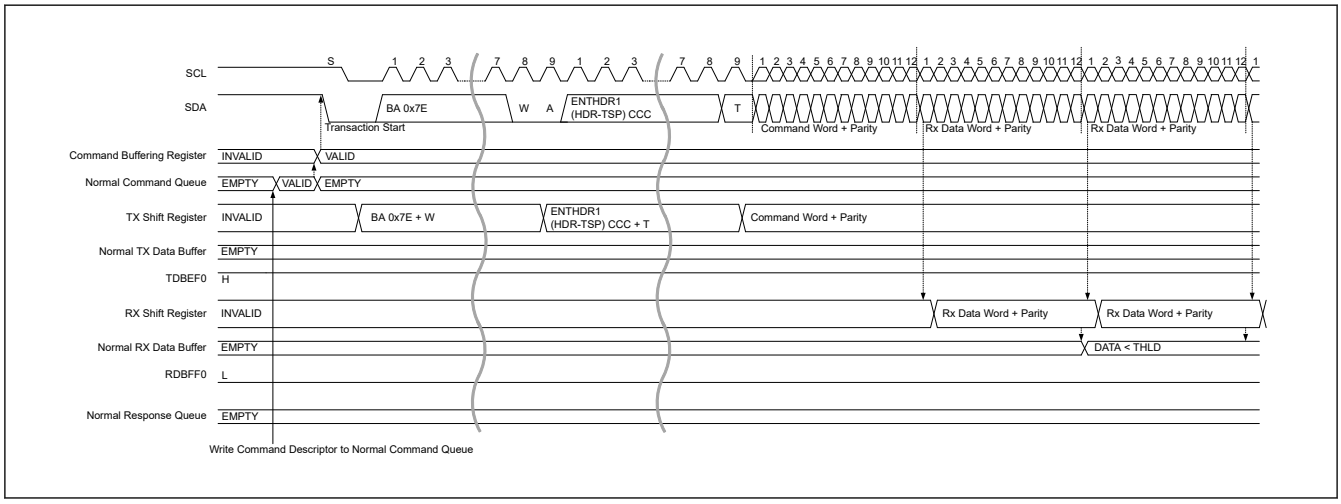


Figure 33.32 HDR data read transfer (HDR-TSP, TSL) timing (1/2)

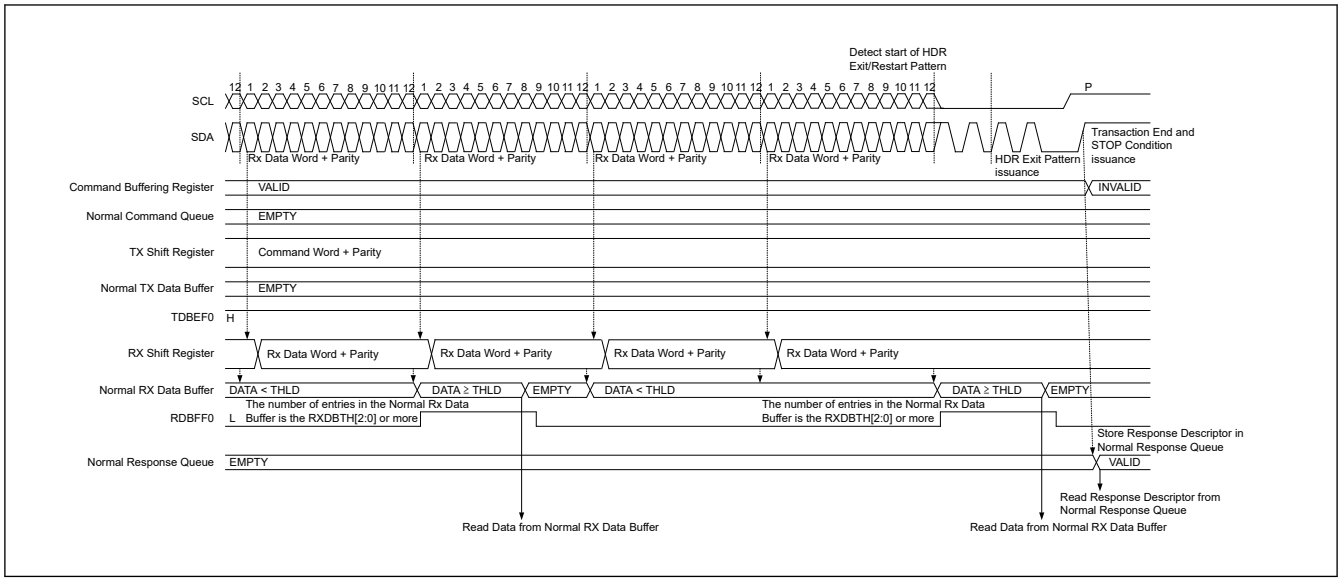


Figure 33.33 HDR data read transfer (HDR-TSP, TSL) timing (2/2)

(f) IBI Transfer

1. Write Command Descriptor to the Command Buffer and issue Transaction on I3C Bus.
   If START Request (SDA Low Drive) is issued from the slave device, I3C drives SCL to Low and completes START condition.
   Thereafter, the SCL is supplied and In-Band Interrupt Request is received.
2. In Slave Address with RnW of the Address Header, if losing Arbitration by issuing In-Band Interrupt from I3C Slave, stop issuing Transaction.
3. According to section 33.3.2.3.8. In-Band Interrupt [I3C mode], detect In-Band Interrupt and process.
4. In the interrupt with IBIQEFF = 1, read the IBI Status Descriptor from the IBI Queue via the NIBIQP register and check the status.
   When detected a Slave Interrupt Request and responded with ACK, Read the IBI Data for the Data Length indicated by the DATA_LENGTH[15:0] bits of the IBI Status Descriptor from the IBI Data Buffer via the NIBIQP register.
5. Restart issuing Transaction of Command of Step1.

An example of the processing procedure after detection of In-Band Interrupt is shown below.

Processing procedure for detecting Mastership Request and transferring master right to Secondary Master

1. If the I3C Secondary Master wins the Arbitration, issue a DEFSLVS CCC and notify Slave information to Secondary Master.
2. Issue a GETACCMST CCC and complete CCC by a STOP condition.

The Mastership processing flow is shown in Figure 33.36

Note:
- After transferring master right to Secondary Master, to get master right again, issue a Mastership Request according to (f) IBI Transfer of (2) I3C Slave Operation.
- After Mastership Request is accepted by the Current Master, to get master right again at receiving the GETACCMST CCC and complete CCC by a STOP condition.

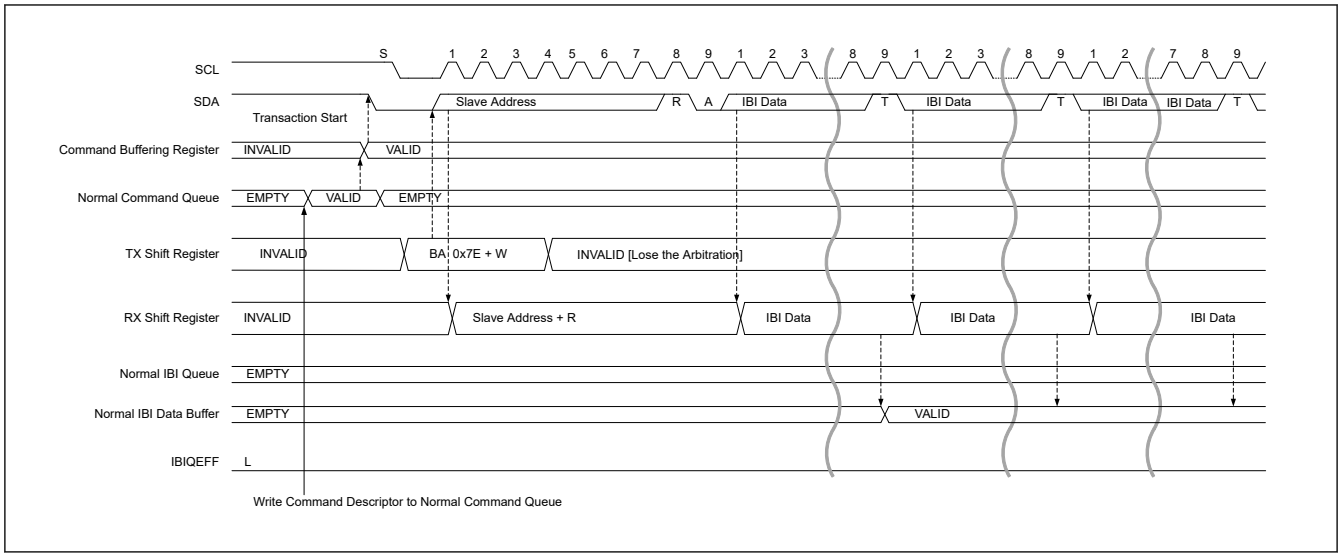


Figure 33.34 I3C master IBI transfer timing (1/2)

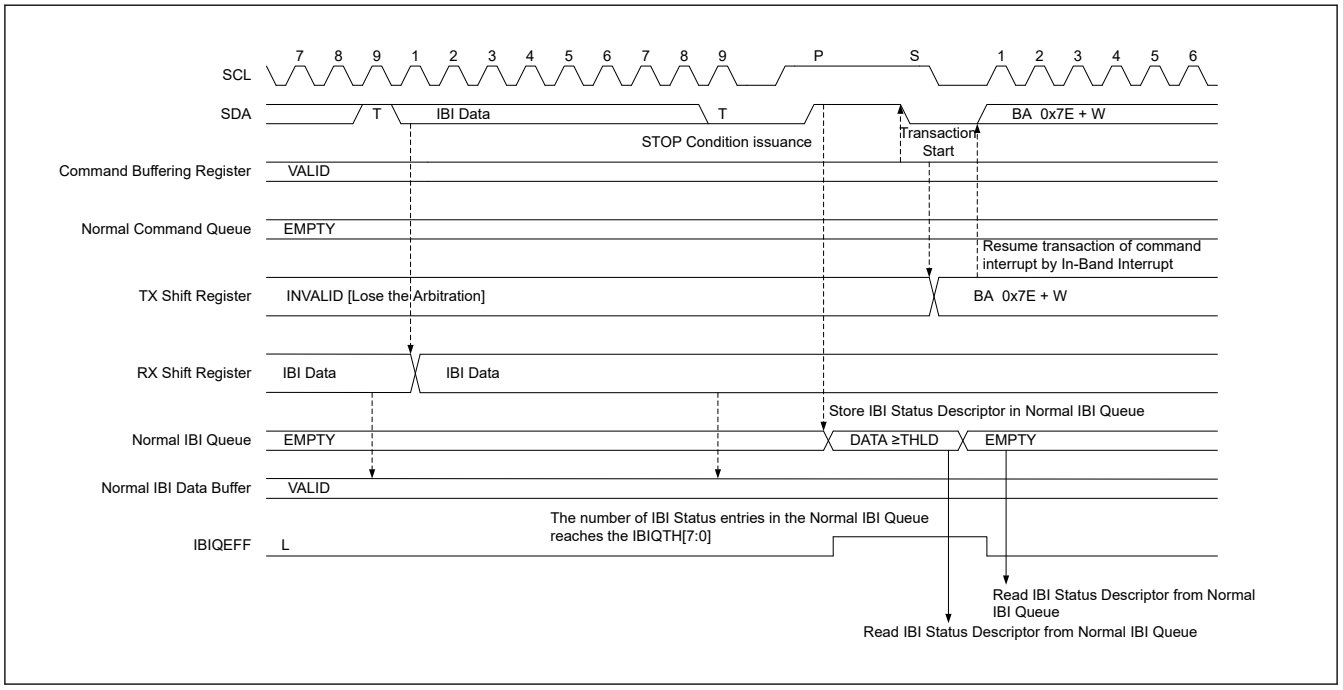


Figure 33.35 I3C master IBI transfer timing (2/2)

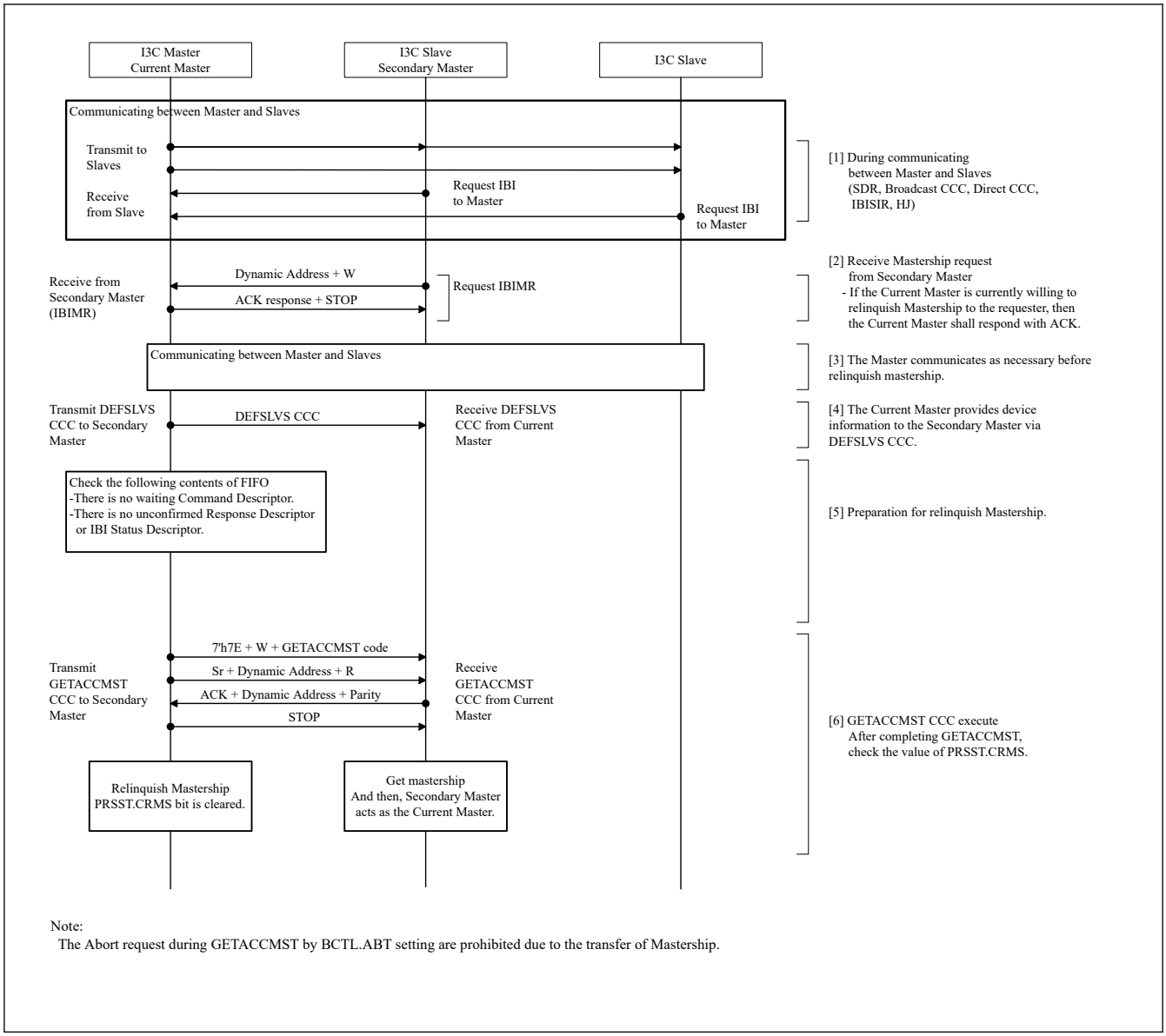


Figure 33.36 I3C Master Mastership processing flow

33.3.2.1.2 Slave Mode Operation

(1) I²C Slave Operation

(a) Data Write Transfer (Single Buffer transfer)

In slave receive operation, the master device outputs the SCL clock and transmit data, and I3C returns acknowledgments as a slave device.

Figure 33.153 shows an example of usage of slave reception and Figure 33.37 and Figure 33.38 show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

1. Initial settings. For details, see section 33.3.3.1. Initial Setting Flow. After initial settings, I3C will stay in the standby state until it receives a slave address that it matches.
2. After receiving a matching slave address, I3C sets one of the corresponding bits SVST.HOAF, GCAF, and SVAF[n] (n = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and outputs the acknowledge bit (ACK) on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 0, I3C continues to place itself in slave receive mode and sets the NTST.RDBFF0 flag to 1.

3. After the BST.SPCNDDF flag is confirmed to be 0 and the NTST.RDBFF0 flag to be 1, dummy read the NTDTBP0 register (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8 bits when the 10-bit address format is selected).
4. When the NTDTBP0 register is read, I3C automatically sets the NTST.RDBFF0 flag to 0. If reading of the NTDTBP0 register is delayed and a next byte is received while the RDBFF0 flag is still set to 1, I3C holds the I3C_SCL line low from one SCL cycle before the timing with which RDBFF0 should be set. In this case, reading the NTDTBP0 register releases the I3C_SCL line from being held at the low level. When the BST.SPCNDDF flag = 1 and the NTST.RDBFF0 flag is also 1, read the NTDTBP0 register until all the data is completely received.
5. Upon detecting the STOP condition, I3C automatically clears bits SVST.HOAF, GCAF, and SVAF[n] (n = 0 to 2) to 0.
6. After checking that the BST.SPCNDDF flag = 1, set the BST.SPCNDDF flag to 0 for the next transfer operation.

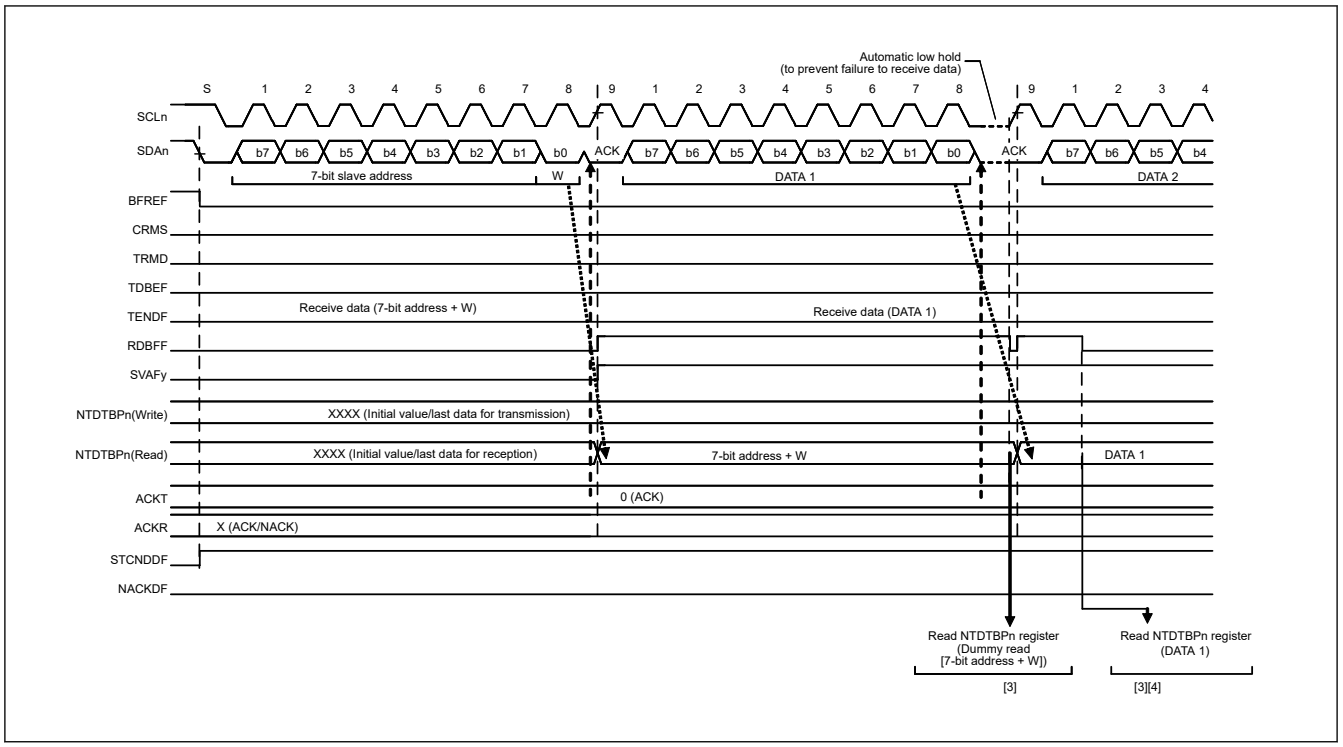


Figure 33.37 Slave receive operation timing (1) (7-bit address format, when ACKTWE = 0)

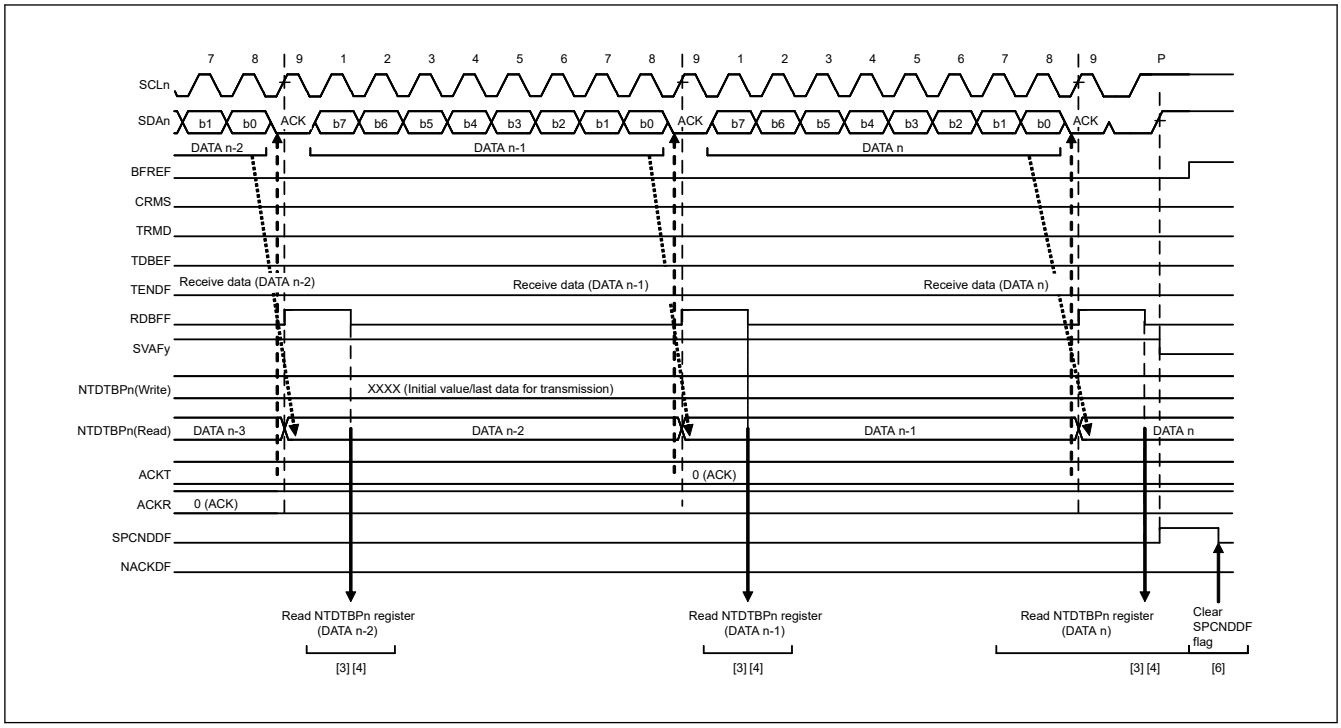


Figure 33.38 Slave receive operation timing (2) (when ACKTWE = 0)

(b) Data Read Transfer (Single Buffer transfer)

In slave transmit operation, the master device outputs the SCL clock, I3C transmits data as a slave device, and the master device returns acknowledgments.

Figure 33.152 shows an example of usage of slave transmission and Figure 33.39 and Figure 33.40 show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

1. Initial settings. For details, see section 33.3.3.1. Initial Setting Flow.
   After initial settings, I3C will stay in the standby state until it receives a slave address that it matches.
2. After receiving a matching slave address, I3C sets one of the corresponding bits SVST.HOAF, GCAF, and SVAF[n] (n = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and outputs the acknowledge bit (ACK) on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 1, I3C automatically places itself in slave transmit mode by setting both the PRSST.TRMD bit and the NTST.TDBEF0 flag to 1.
3. After the NTST.TDBEF0 flag is confirmed to be 1, write the data for transmission to the NTDTBP0 register.
   At this time, if I3C does not receive acknowledge from the master device (receives an NACK signal) while the BSTE.NACKDE bit = 1, I3C aborts transfer of the next data.
4. Wait until the following (a) or (b) condition.
   (a) The BST.NACKDF flag is set to 1.
   (b) The BST.TENDF flag is set to 1 while the NTST.TDBEF0 flag = 1, after the last byte for transmission is written to the NTDTBP0 register.
5. When the BST.NACKDF flag or the BST.TENDF flag = 1, dummy read the NTDTBP0 register to complete the processing. This releases the I3C_SCL line.
6. Upon detecting the STOP condition, I3C automatically sets bits SVST.HOAF, GCAF, and SVAF[n] (n = 0 to 2), flags NTST.TDBEF0 and BST.TENDF, and the PRSST.TRMD bit to 0, and enters slave receive mode.
7. After checking that the BST.SPCNDDF flag = 1, set the BST.NACKDF and SPCNDDF flags to 0 for the next transfer operation.

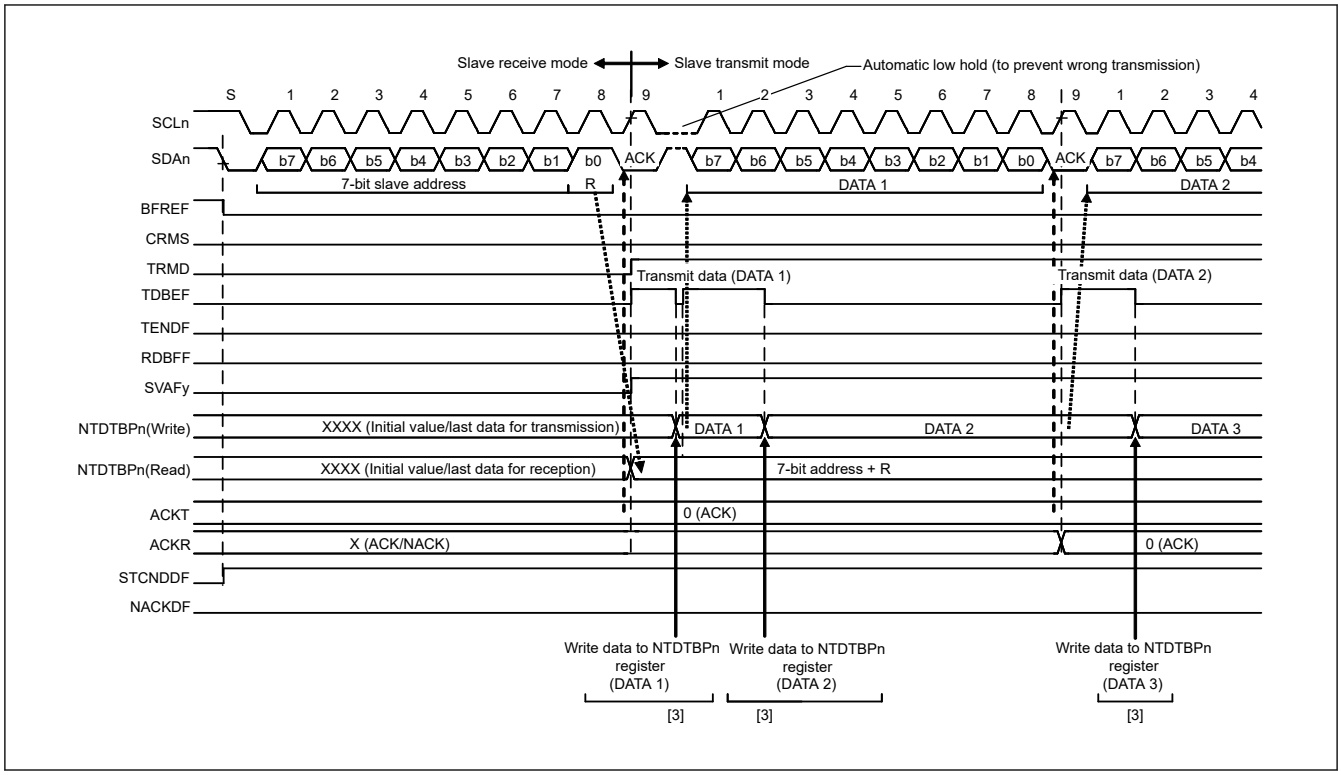


Figure 33.39 Slave transmit operation timing (1) (7-bit address format)

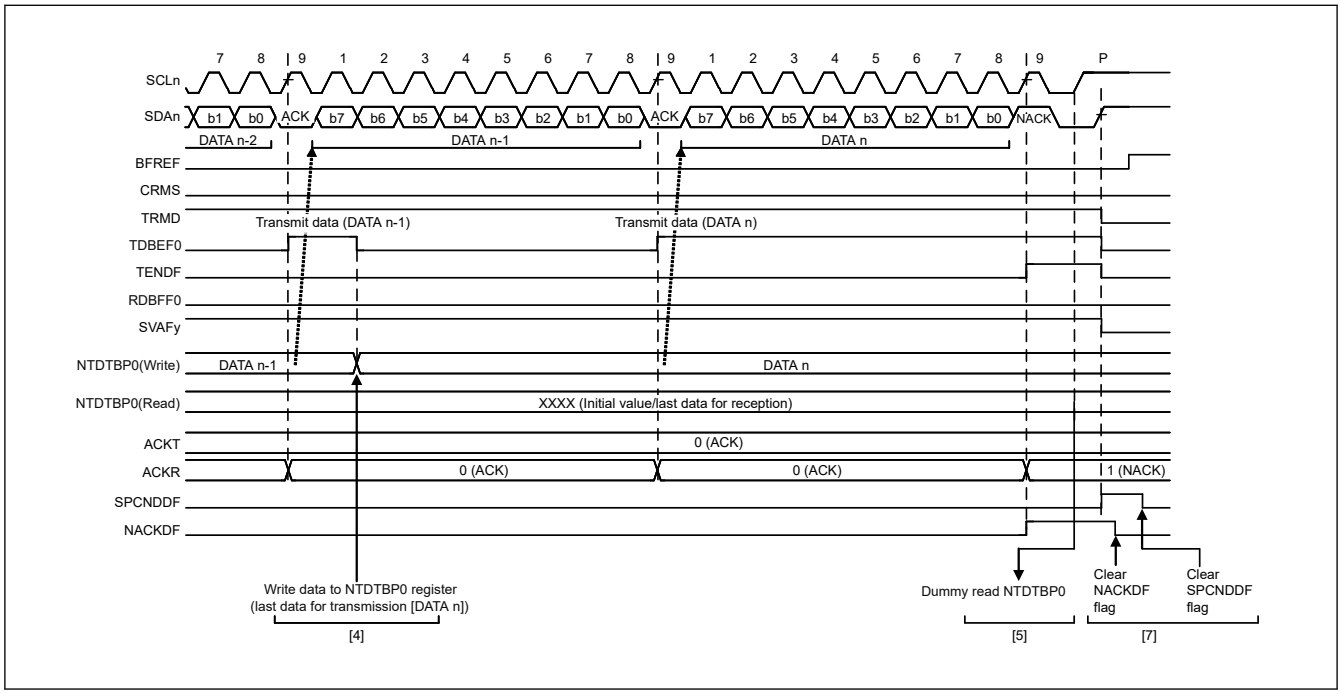


Figure 33.40 Slave transmit operation timing (2)

(2) I3C Slave Operation

(a) Dynamic Address Assign Procedure

After initializing I3C, the I3C master first performs Dynamic Address Assign Procedure.

The operation of R-I3 during the Dynamic Address Assign Procedure by ENTDAA CCC is described below.

1. Initial setting (For details, see section 33.3.3.1.2. I3C Initial Setting Flow)

2. When ENTDAA CCC is received, I3C transmits Provisional ID (SDCTPIDH[31:0], SDCTPIDL[15:0]), BCR (SVDCT.TBCRn), DCR (SVDCT.TDCR[7:0]) until a dynamic address is assigned. (For details, see "In case of Broadcast CCC (ENTDAA)" of (6) CCC detection function [I3C mode].)
3. When ENTDAA CCC is completed and a STOP condition is detected, Receive Status Descriptor is stored in Receive Status Queue.
4. Read Receive Status Descriptor via NRSQP register and check the status.
5. Read the data for the Data Length indicated by the DATA_LENGTH[15:0] bits of the Receive Status Descriptor from the Receive Data Buffer via the NTDTBP0 register.

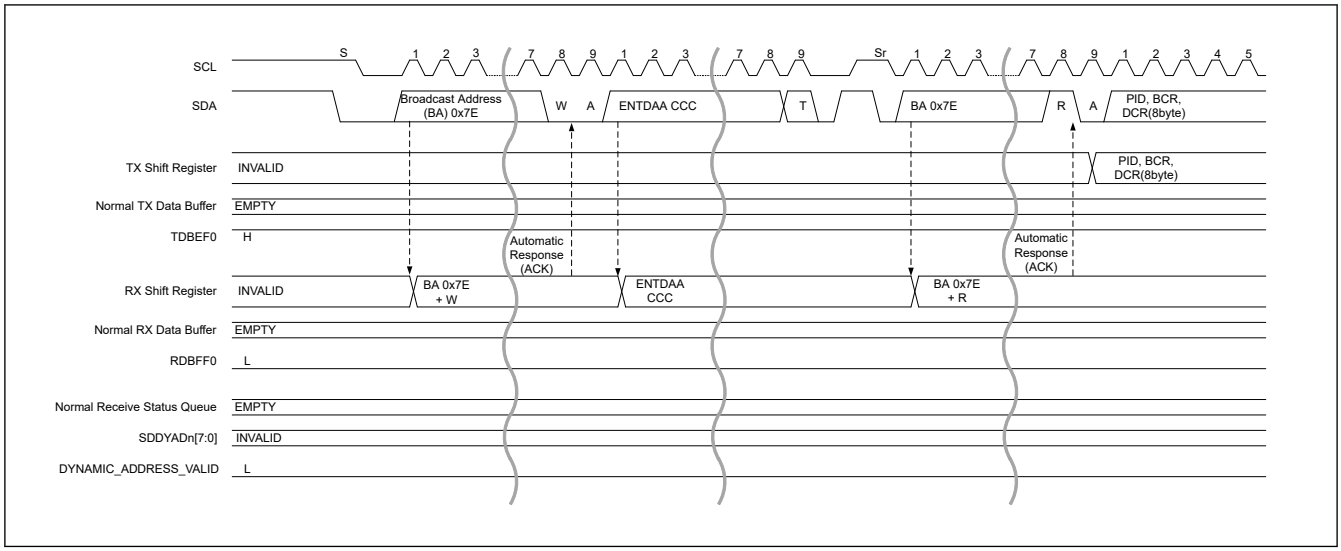


Figure 33.41 Dynamic address assign procedure (ENTDAA CCC) timing (1/3)

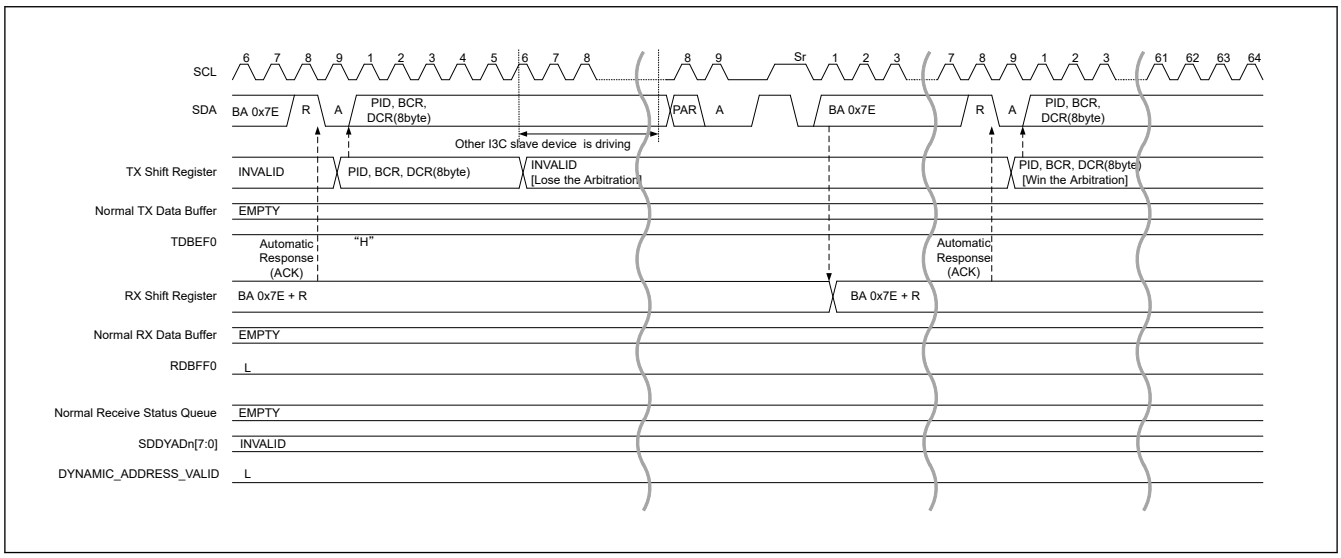


Figure 33.42 Dynamic address assign procedure (ENTDAA CCC) timing (2/3)

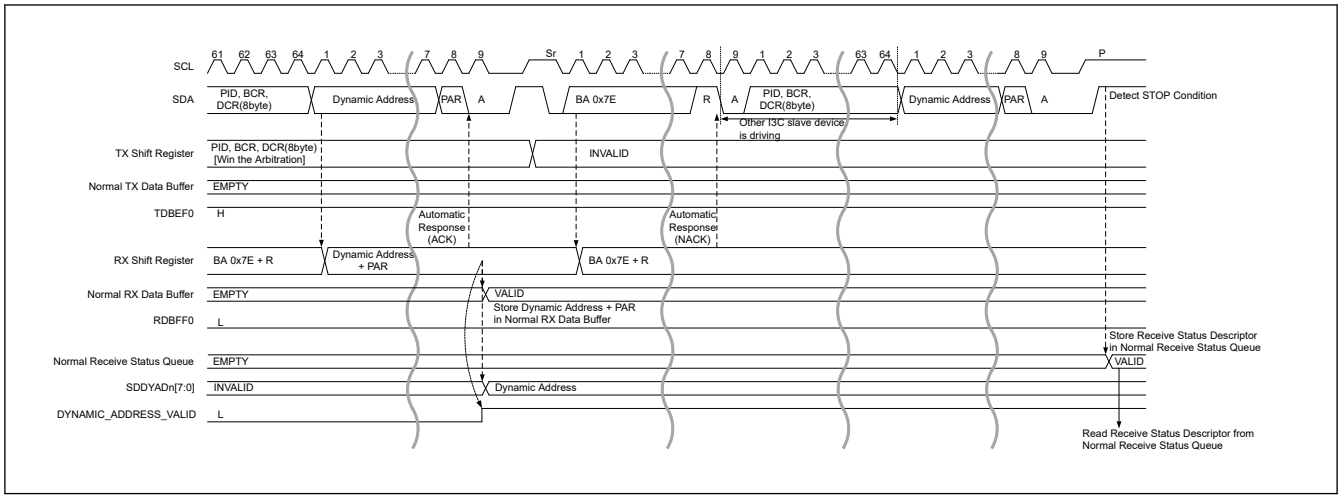


Figure 33.43 Dynamic address assign procedure (ENTDAA CCC) timing (3/3)

When communicating with a Static Address until the Dynamic Address is assigned from the I3C Master, by setting to the DVSTAD[6:0] bit of DAT (SDATBASn register), the SSTADV bit of the SVDVADn register is set to 1 and the Static Address Will be effective.

If the I3C Slave has a Static Address and the I3C Master executes the Dynamic Address Assign Procedure, it is possible to assign a Dynamic Address with SETDASA CCC.

The operation of I3C during SETDASA CCC Dynamic Address Assign Procedure is described below.

1. Initial setting (For details, see section 33.3.3.1.2. I3C Initial Setting Flow)
2. When SETDASA CCC which agrees with its own Static Address is received, the SDDYAD [7:0] bit of DAT (SDATBAS0 register) is renewed and SDYADV bit of SVDVAD0 register is set in 1. (For details, see "In case of Direct Write CCC" of (6) CCC detection function [I3C mode].)
3. When SETDASA CCC is completed and a STOP condition is detected, Receive Status Descriptor is stored in Receive Status Queue.
4. Read Receive Status Descriptor via NRSQP register and check the status.

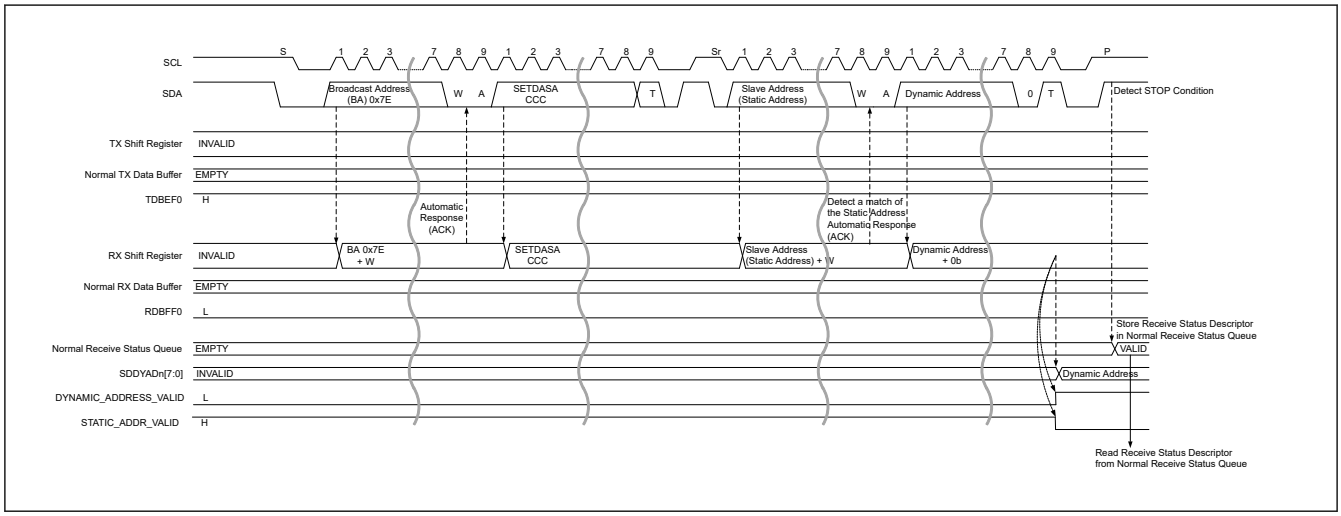


Figure 33.44 Dynamic address assign procedure (SETDASA CCC) timing

(b) SDR Data Write Transfer

1. When Transaction is issued from the I3C Master, it compares the Slave Address of Address Header with its own Slave Address, and if it matches, I3C responds with ACK.
   When a Transaction is received, if the Receive Data Buffer is full, the I3C Slave will respond with NACK in the Address Header.
   In preparation for retrying the I3C Master, read the data from the Receive Data Buffer via the NTDTBPn register, and empty the Receive Data Buffer.

2. Data received from I3C Master is stored in the Receive Data Buffer.
3. With the RDBFF0 = 1 interrupt, the received data is read from the Receive Data Buffer via the NTDTBPn register.
4. When Repeated START condition or STOP condition is detected, the Receive Status Descriptor is stored in the Receive Status Queue.
5. Read Receive Status Descriptor via NRSQP register and check the status.

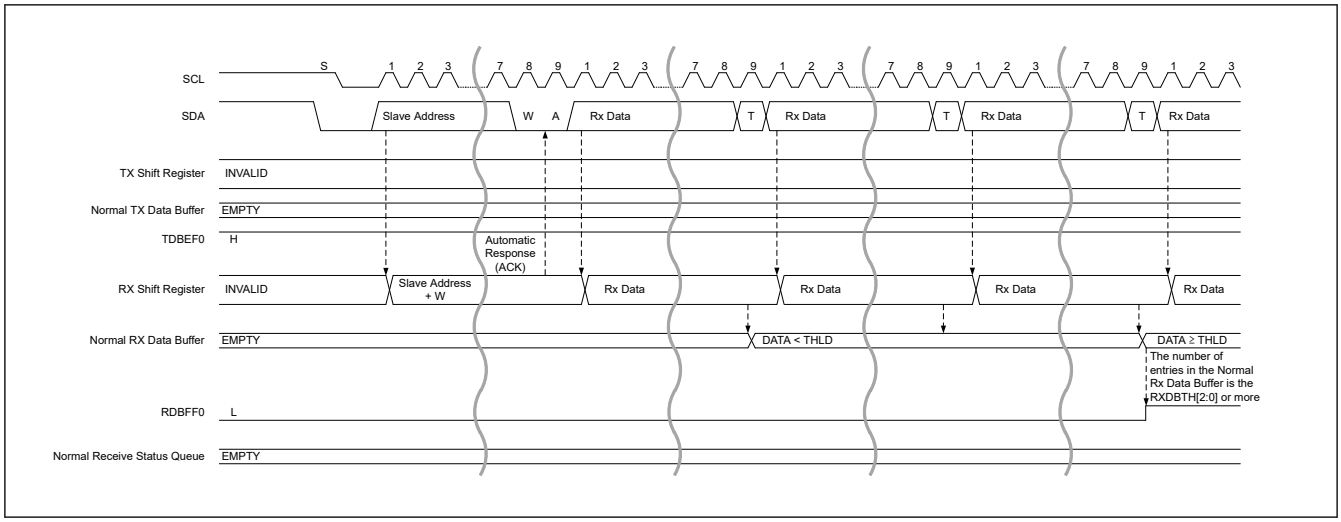


Figure 33.45 SDR data write transfer timing (1/2)

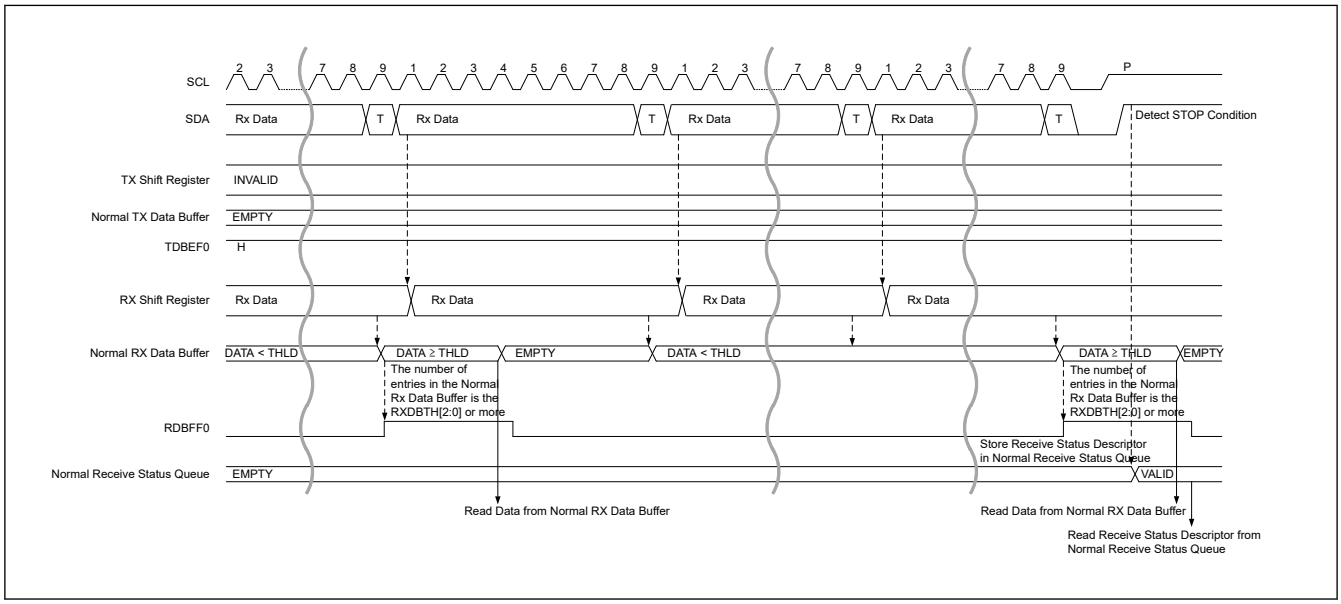


Figure 33.46 SDR data write transfer timing (2/2)

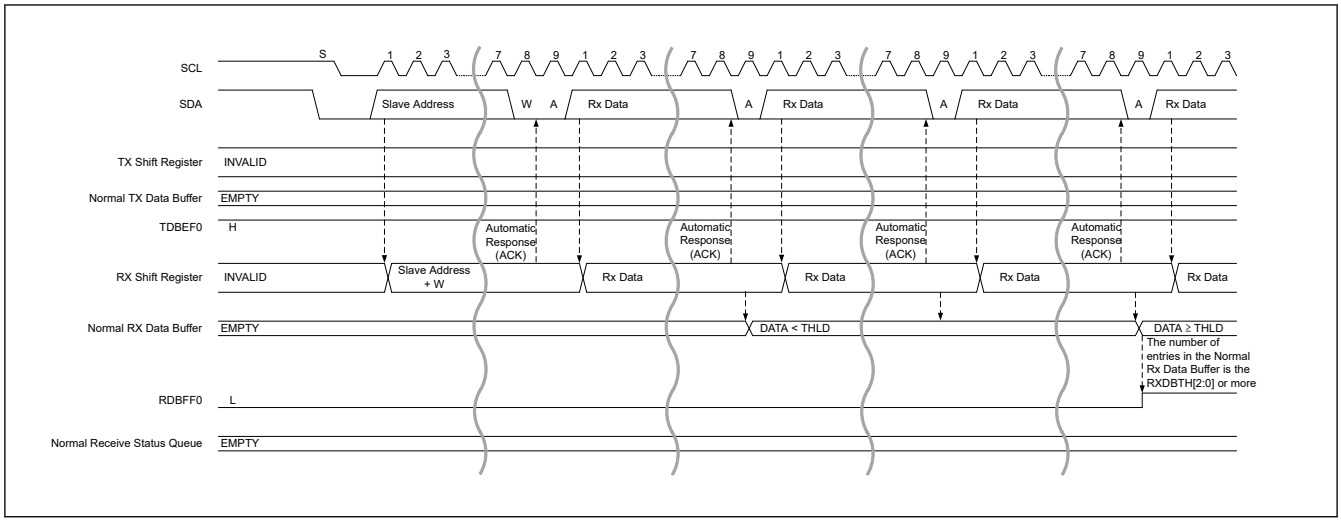


Figure 33.47 Legacy I²C message data write transfer timing (1/2)

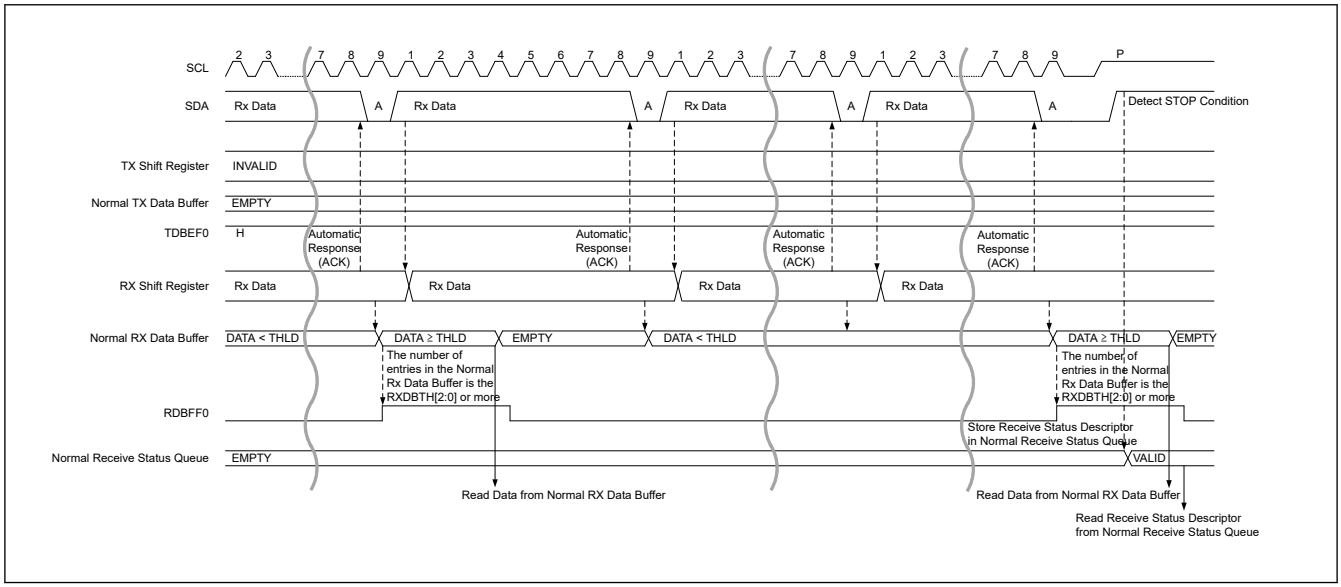


Figure 33.48 Legacy I²C message data write transfer timing (2/2)

(c) SDR Data Read Transfer

1. Write the data requested from the I3C Master to the Tx Data Buffer via the NTDTBPn register.
2. When Transaction is issued from the I3C Master, it compares the Slave Address of Address Header with its own Slave Address, and if it matches, I3C responds with ACK.
   When a Transaction is received, if the Tx Data Buffer is EMPTY, I3C Slave responds with NACK with the Address Header.
   In preparation for retrying the I3C Master, write data to the Tx Data Buffer via the NTDTBPn register.
3. Transmit the data stored in the Tx Data Buffer.
4. If data to be transmitted still remains, write the data to be transmitted with an interrupt by TDBEF0 = 1 to the Tx Data Buffer via the NTDTBPn register.
5. SDR:
   When the transmission of the data stored in the Tx Data Buffer is completed, Low is output to the T-bit following Data, and it is notified to the I3C Master that it is the final data.
   Legacy I²C Message:
   When NACK is detected, data transmission is terminated.
6. When a Repeated START condition or STOP condition is detected, the Receive Status Descriptor is stored in the Receive Status Queue.

7. Read the Receive Status Descriptor via NRSQP and check the status.

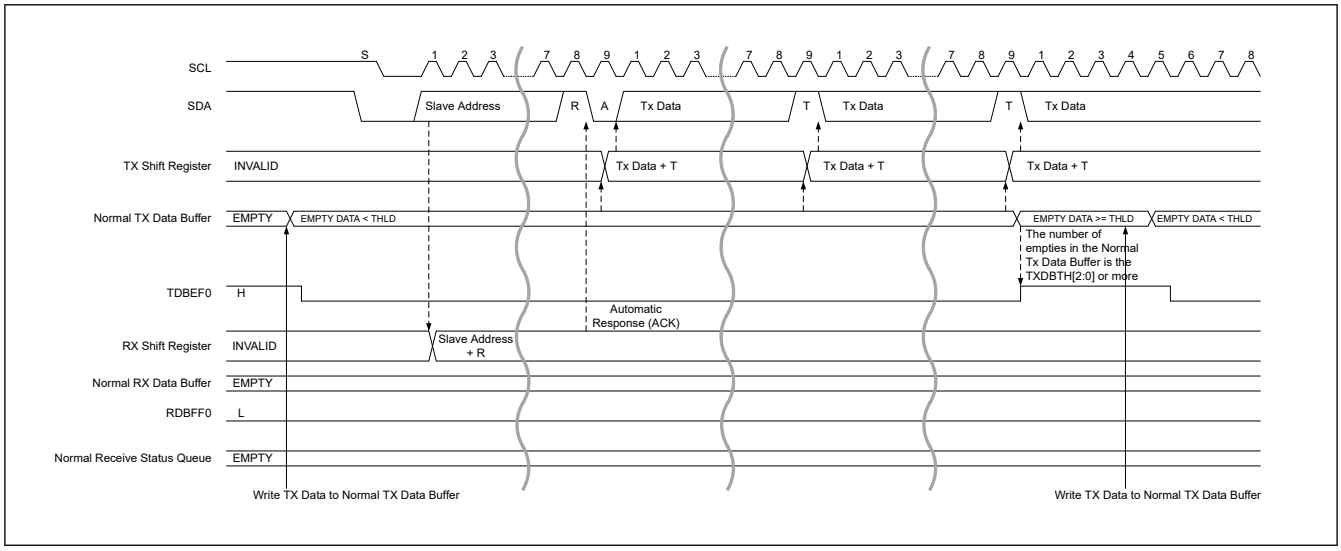


Figure 33.49 SDR data read transfer timing (1/2)

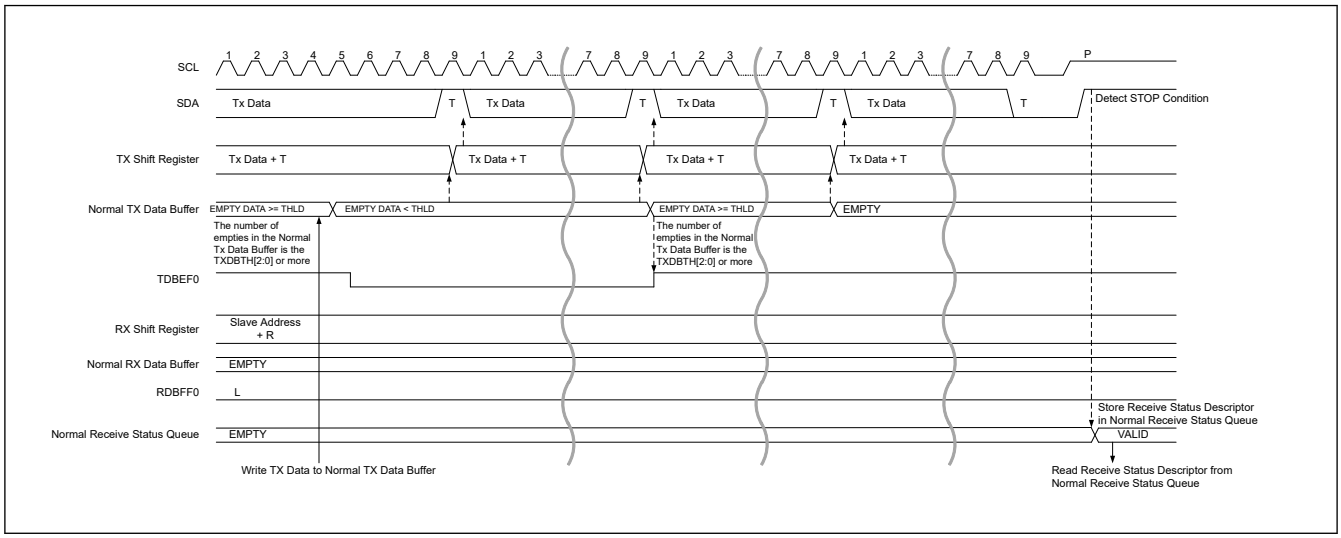


Figure 33.50 SDR data read transfer timing (2/2)

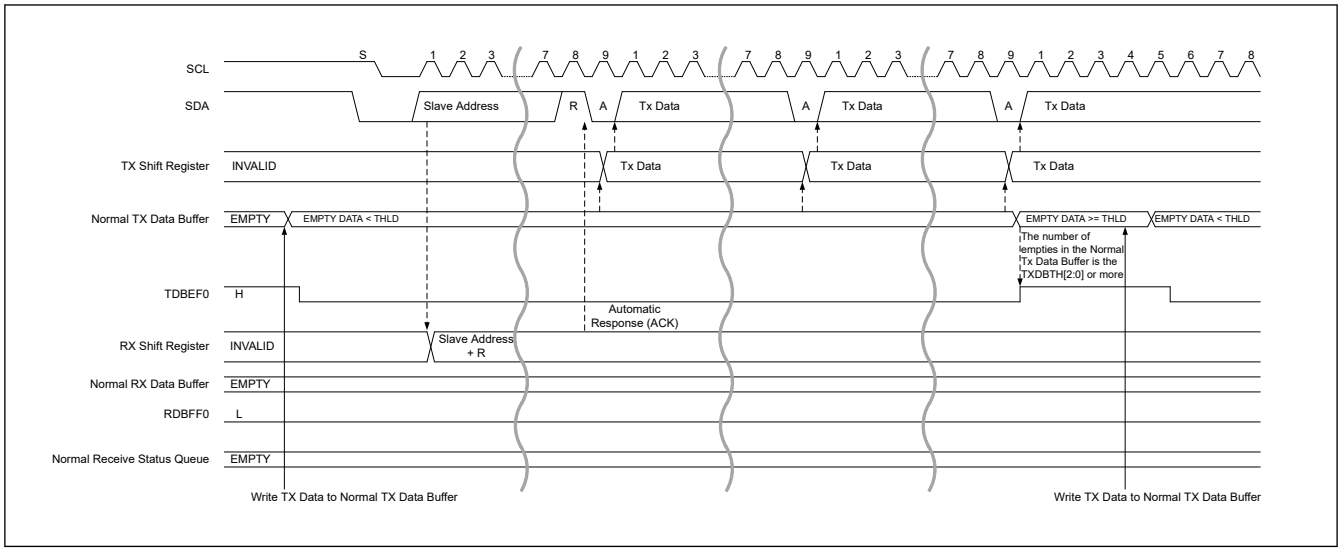


Figure 33.51 Legacy I²C message data read transfer timing (1/2)

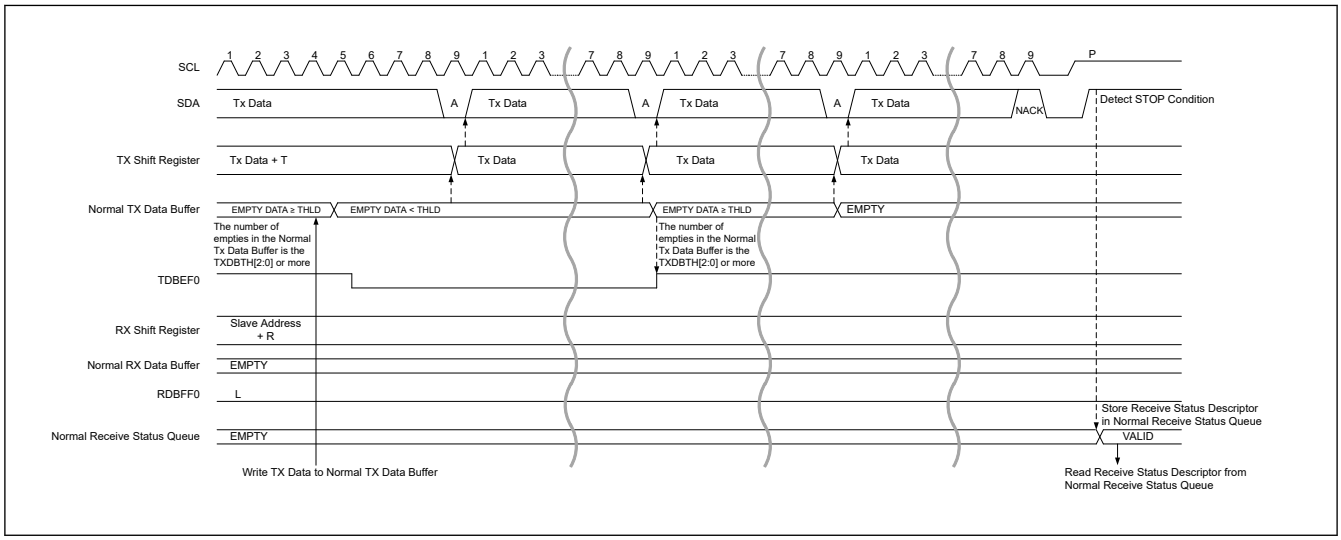


Figure 33.52 Legacy I²C message data read transfer timing (2/2)

(d) HDR Data Write Transfer

1. Upon receipt of ENTHDR * CCC from the I3C Master, it transits to HDR mode (For details, see "In case of Broadcast CCC (ENTHDR *)" of (6) CCC detection function [I3C mode]).
2. Compare the Slave Address of the HDR Command Word issued from the I3C Master with its own Slave Address, and if it matches, receive the following Data Word.
3. Data received from the I3C Master is stored in the Receive Data Buffer.
4. With the RDBFF0 = 1 interrupt, the received data is read from the Receive Data Buffer via the NTDTBPn register.
5. When detecting HDR Restart Pattern or HDR Exit Pattern + STOP condition, it stores the Receive Status Descriptor into the Receive Status Queue.
6. Read the Receive Status Descriptor via NRSQP and check the status.

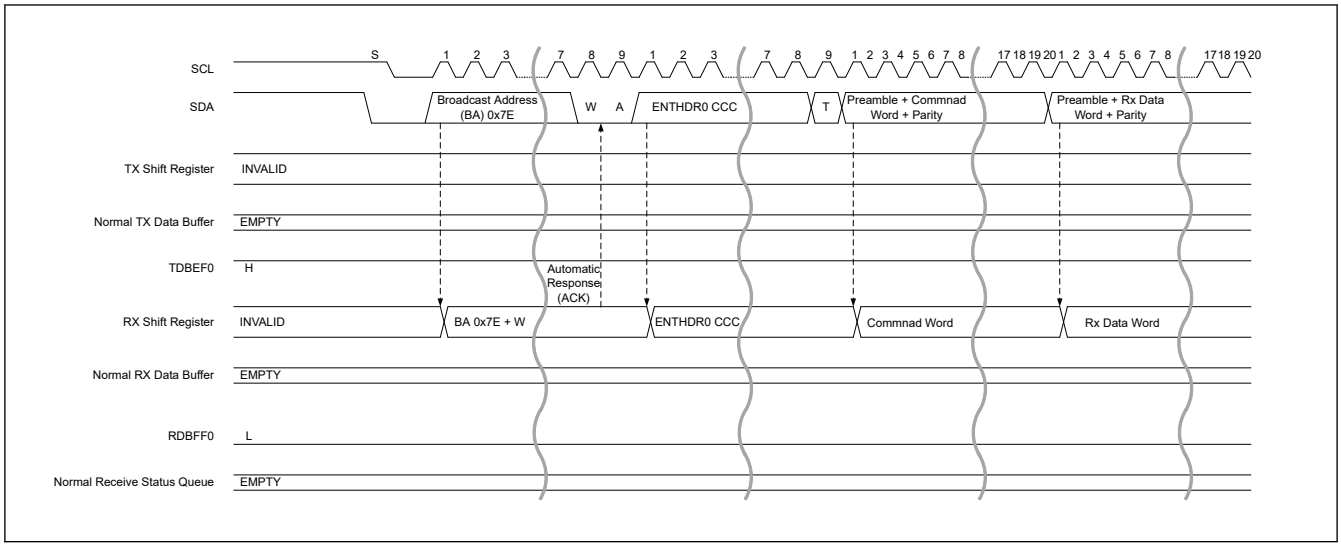


Figure 33.53 HDR data write transfer (HDR-DDR) timing (1/3)

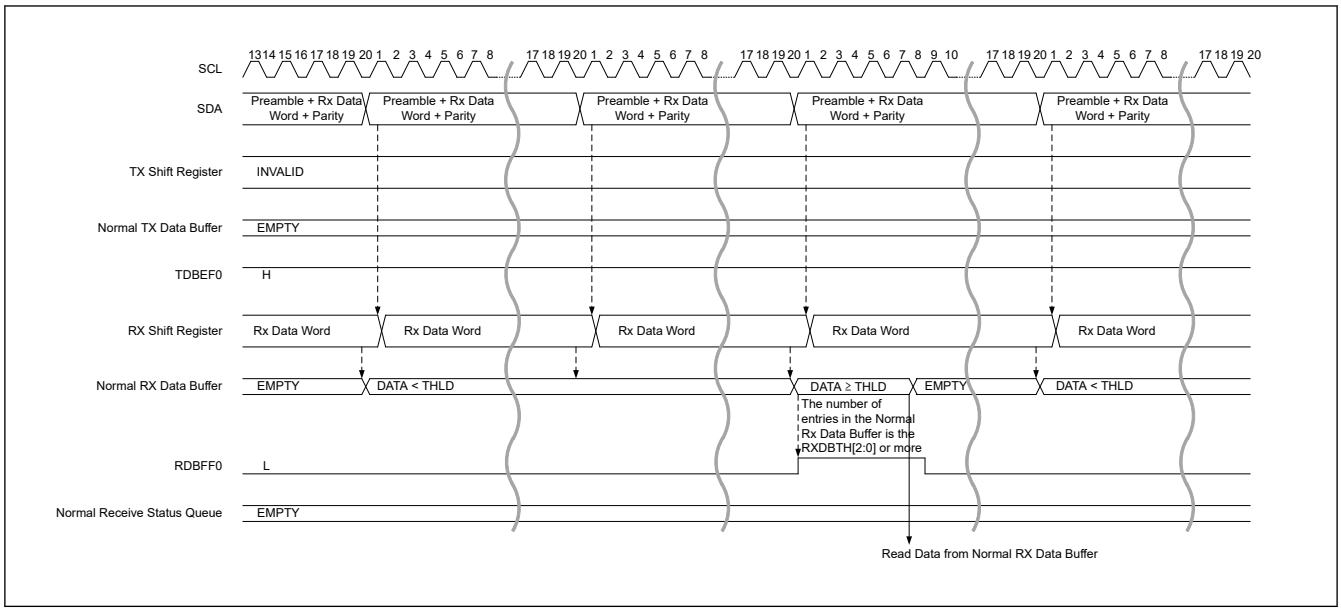


Figure 33.54 HDR data write transfer (HDR-DDR) timing (2/3)

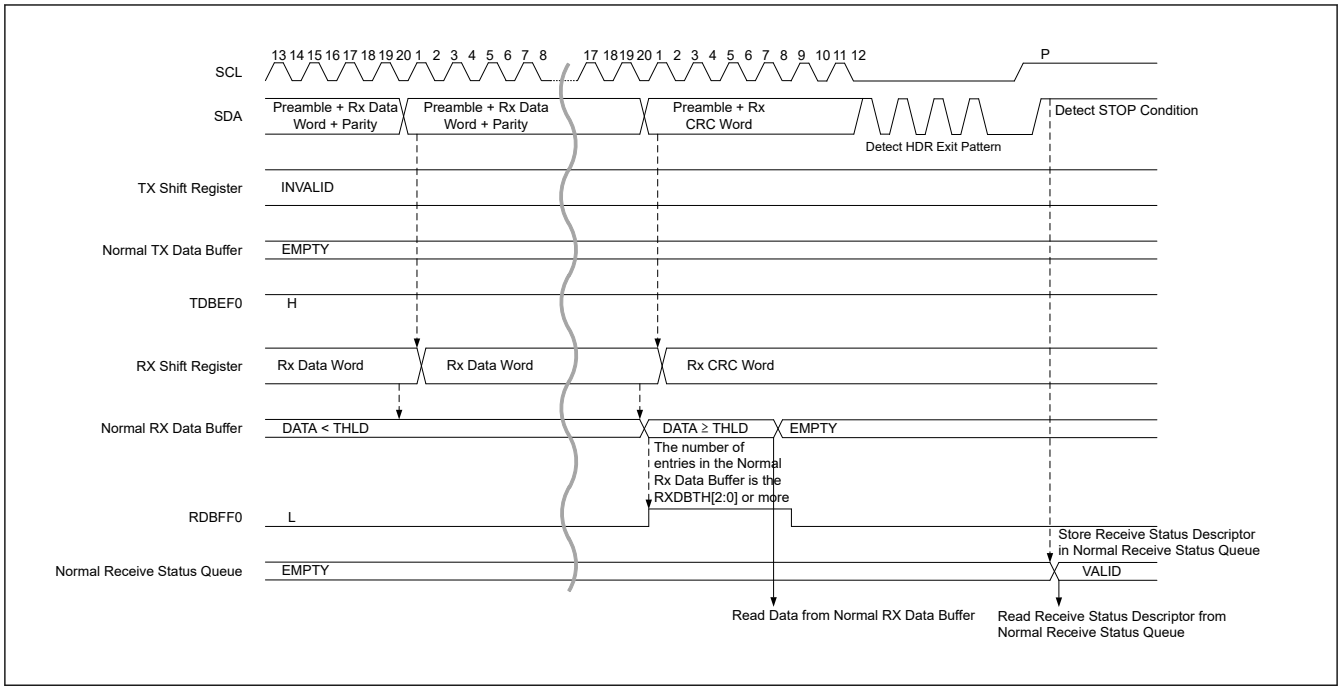


Figure 33.55 HDR data write transfer (HDR-DDR) timing (3/3)

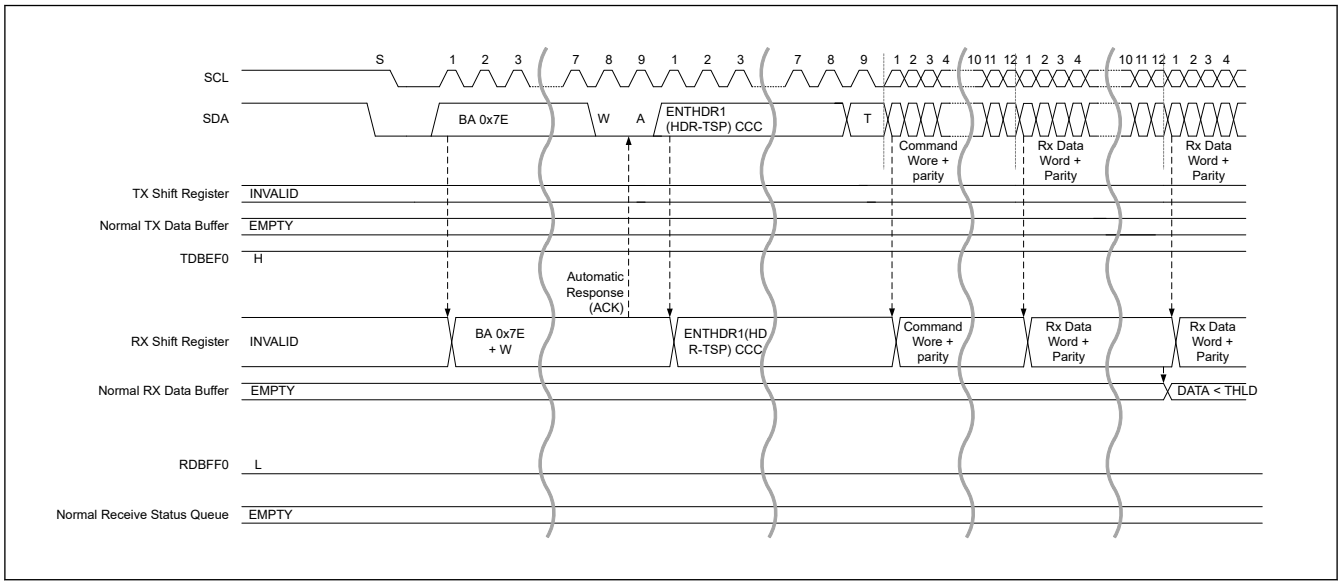


Figure 33.56 HDR data write transfer (HDR-TSP, TSL) timing (1/2)

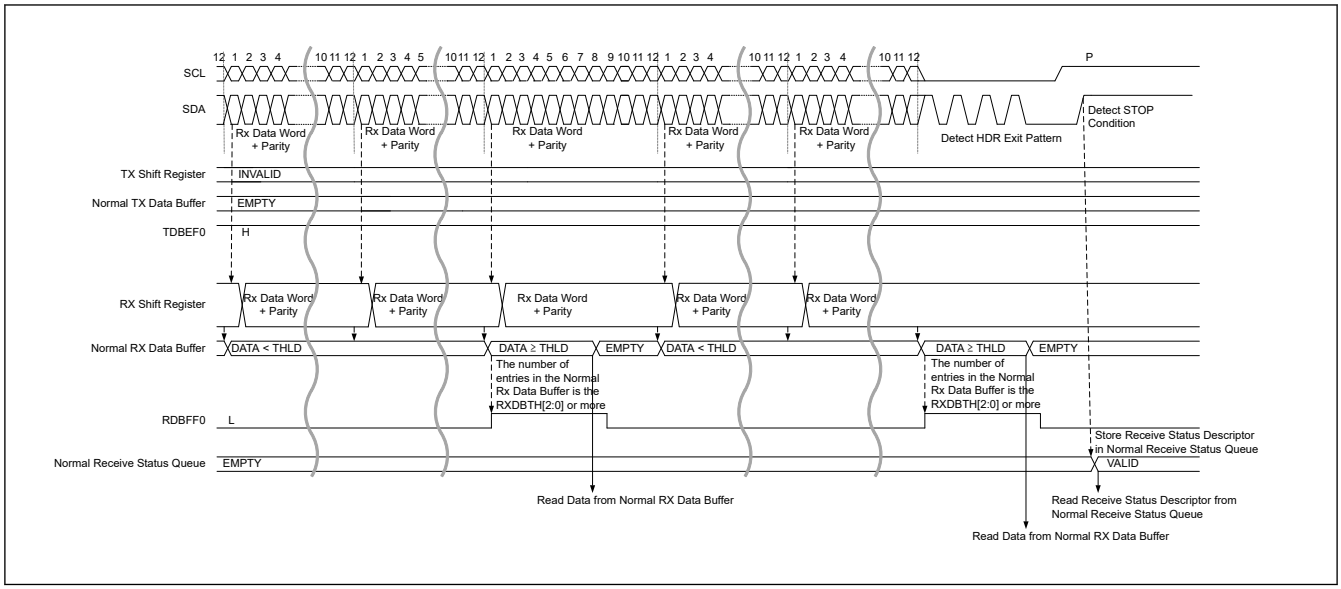


Figure 33.57 HDR data write transfer (HDR-TSP, TSL) timing (2/2)

(e) HDR Data Read Transfer

1. Write the data requested by the I3C Master to the Tx Data Buffer via the NTDTBPn register.
2. Upon receipt of ENTHDR * CCC from the I3C Master, it transits to HDR mode (For details, see "In case of Broadcast CCC (ENTHDR *)" of (6) CCC detection function [I3C mode]).
3. When comparing the Slave Address of the HDR Command Word issued from the I3C Master with its own Slave Address, if it matches, the Data stored in the Tx Data Buffer is transmitted.
   If the Tx Data Buffer is EMPTY when the Slave Address of the HDR Command Word matches its own Slave Address, a NACK response will be sent.
   NACK response :
   - HDR-DDR :
     Send 11 followed by Command Word followed by Preamble.
   - HDR-Ternary :
     Send the beginning of HDR Restart and HDR Exit Pattern (in English translation: "start of the HDR Restart Pattern and HDR Exit Pattern." In MIPI Spec)
4. If data to be transmitted still remains, Write the data to be transmitted with an interrupt by TDBEF0 = 1 to the Tx Data Buffer via the NTDTBPn register.
5. HDR-DDR:
   When the transmission of the data stored in the Tx Data Buffer is completed, the CRC Word is sent continuously to notify the I3C Master that it is the final data.
   HDR-Ternary:
   When the transmission of the data stored in the Tx Data Buffer is completed, the beginning of the HDR Restart or HDR Exit Pattern is sent continuously to notify the I3C Master that it is the final data.
6. When detecting HDR Restart Pattern and HDR Exit Pattern + STOP condition, it stores the Receive Status Descriptor into the Receive Status Queue.
7. Read the Receive Status Descriptor via NRSQP and check the status.

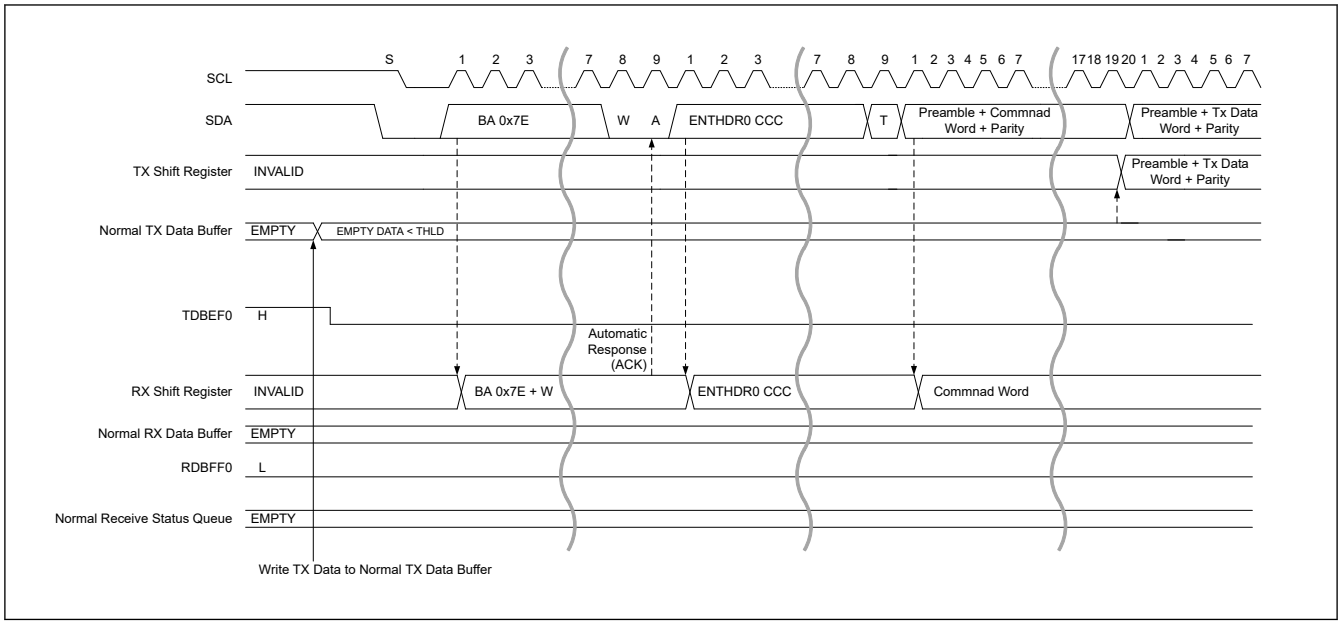


Figure 33.58 HDR data read transfer (HDR-DDR) timing (1/3)

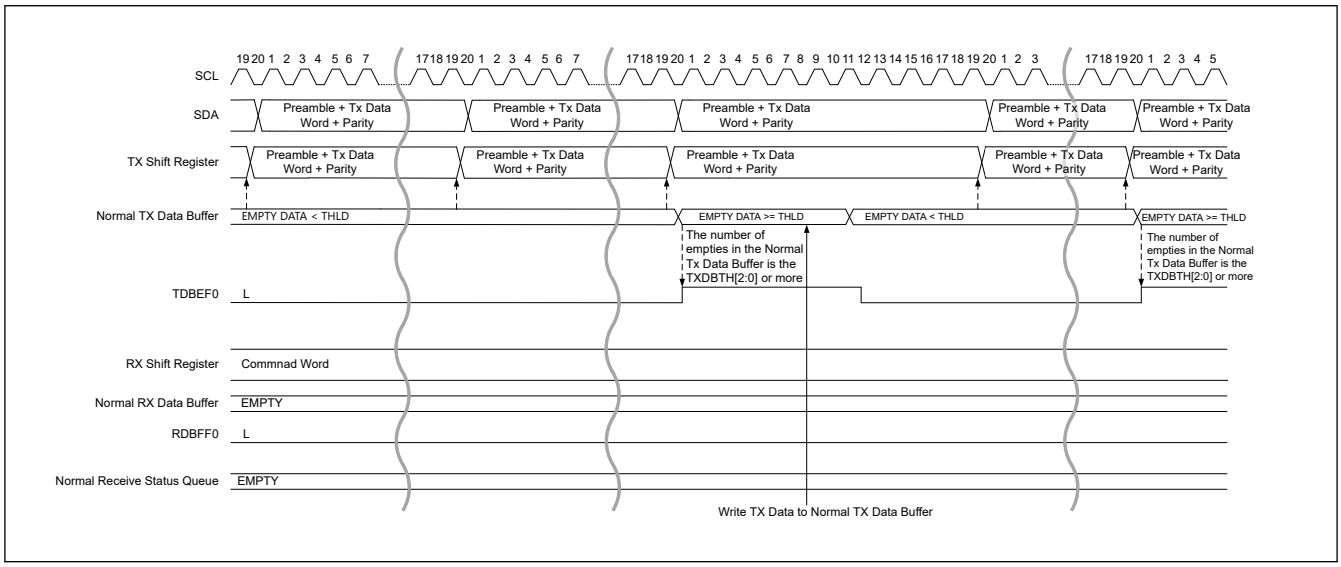


Figure 33.59 HDR data read transfer (HDR-DDR) timing (2/3)

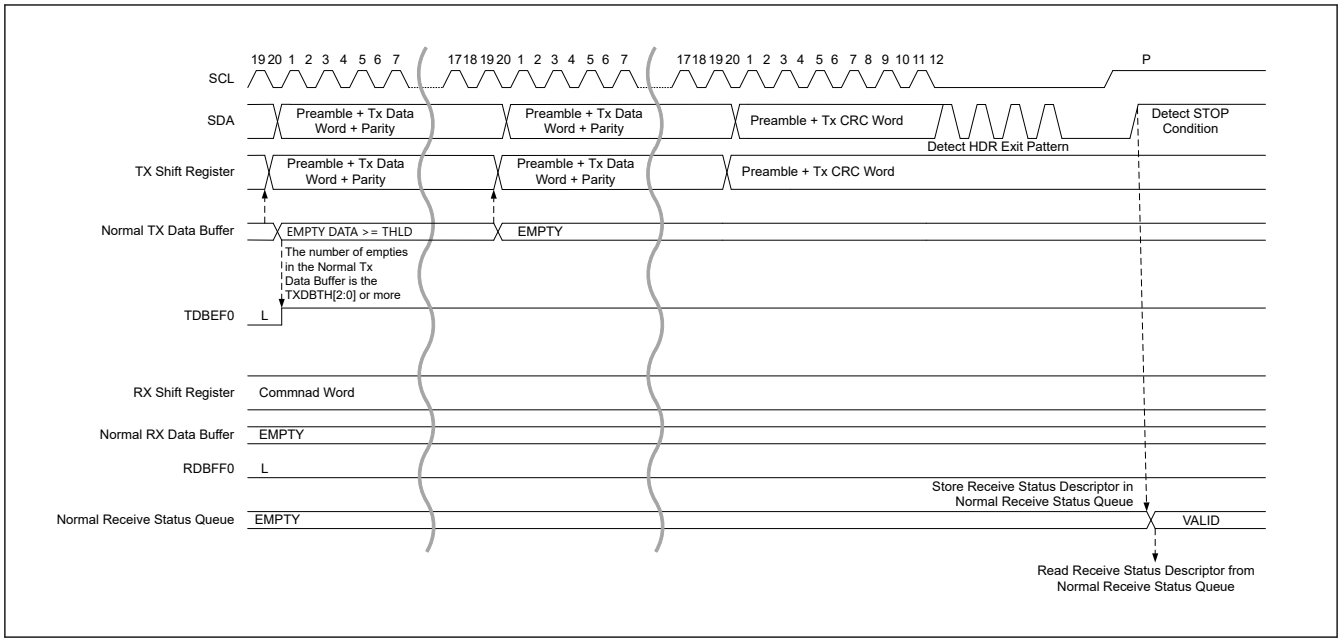


Figure 33.60 HDR data read transfer (HDR-DDR) timing (3/3)

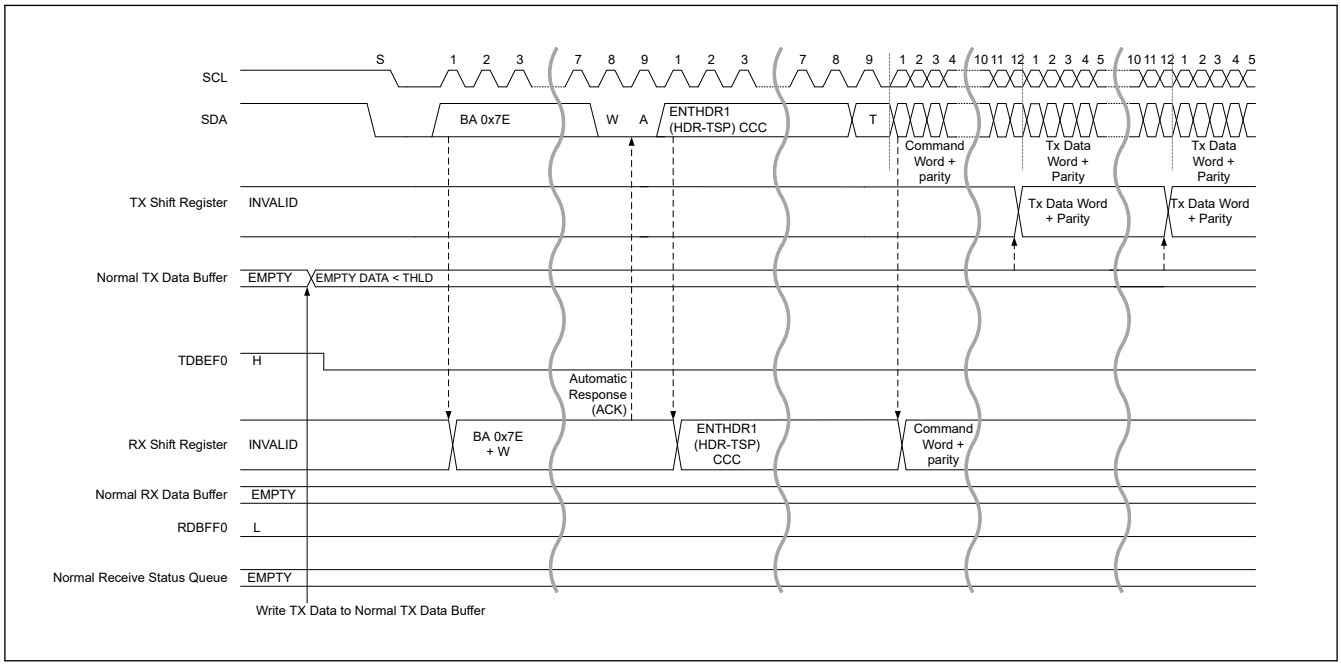


Figure 33.61 HDR data read transfer (HDR-TSP, TSL) timing (1/2)

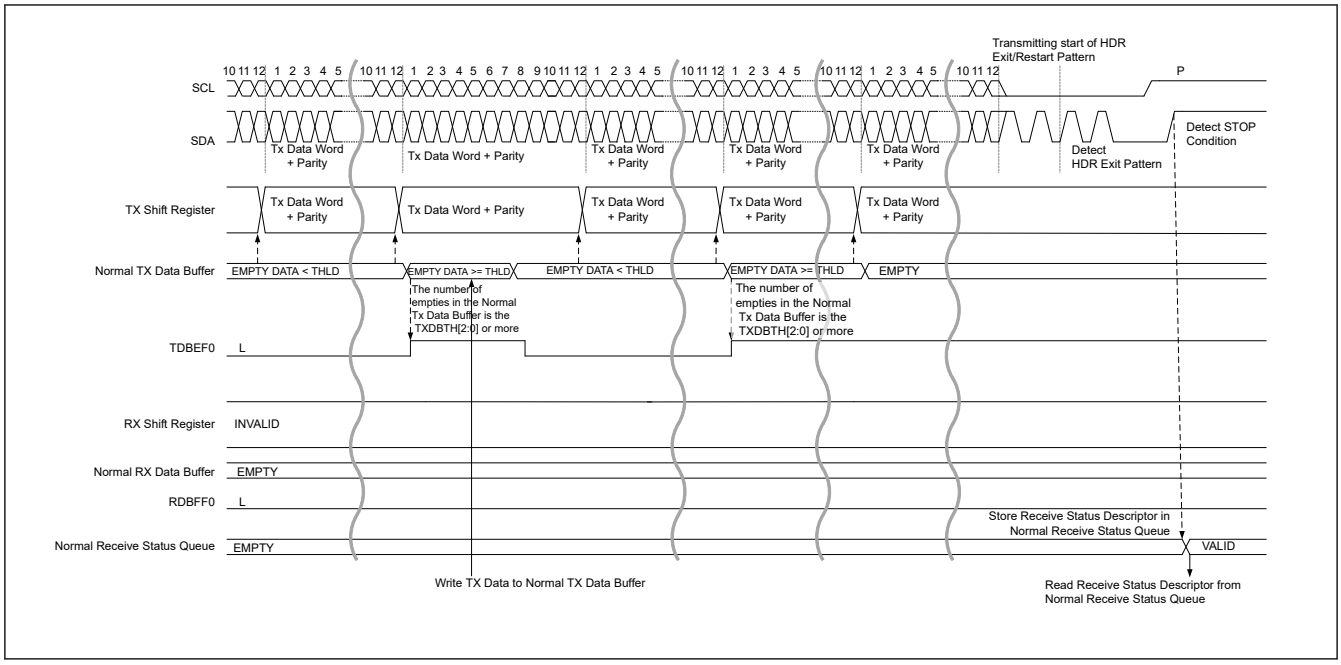


Figure 33.62 HDR data read transfer (HDR-TSP, TSL) timing (2/2)

(f) IBI Transfer

1. When sending Slave Interrupt Request.
   When transmitting IBI Data, write IBI Data to the IBI Data Buffer via the NIBIQP register.
2. Write Command Descriptor (Immediate Transfer Command or Regular Transfer Command) to the Command Buffer for IBI Transfer via the NCMDQP register.
3. When Command Descriptor is written to Command Buffer, IBI Transaction is issued under the following conditions.
   - When START condition is detected in Slave Interrupt Request or Mastership Request. (Does not apply a Repeated START condition)
   - If no START is forthcoming within the following Bus Condition, then this module issue a START Request by pulling the I3C_SDA line Low.
   - (a) Slave Interrupt Request, Mastership Request : Bus Available
4. In Slave Address with RnW of the Address Header, if losing Arbitration by issuing a Transaction from I3C Master, stop issuing Transaction.
   When detecting Repeated START condition or STOP condition, store the Response Descriptor into the Response Buffer.
5. When sending Slave Interrupt Request:
   - When IBI data for transmission still remain, write IBI data with an interrupt by IBIQEFF = 1 to the IBI Data Buffer via the NIBIQP register.
   - When the transmission of IBI Data for the number of Data Length specified by the DATA_LENGTH[15:0] bits of the Command Descriptor is completed, output Low to the T-bit following IBI Data and notify the I3C Master that it is the final IBI Data.
6. When detecting Repeated START condition or STOP condition, store the Response Descriptor into the Response Buffer.
7. Read the Response Descriptor form the Response Buffer with the NRSPQP register and check the status. If NACK is responded, repeat steps 1 to 7.
8. When sending Slave Interrupt Request:
   Check that the value of the DATA_LENGTH[15:0] bit of the Response Descriptor is 0.

The Mastership processing flow is shown in Figure 33.65.

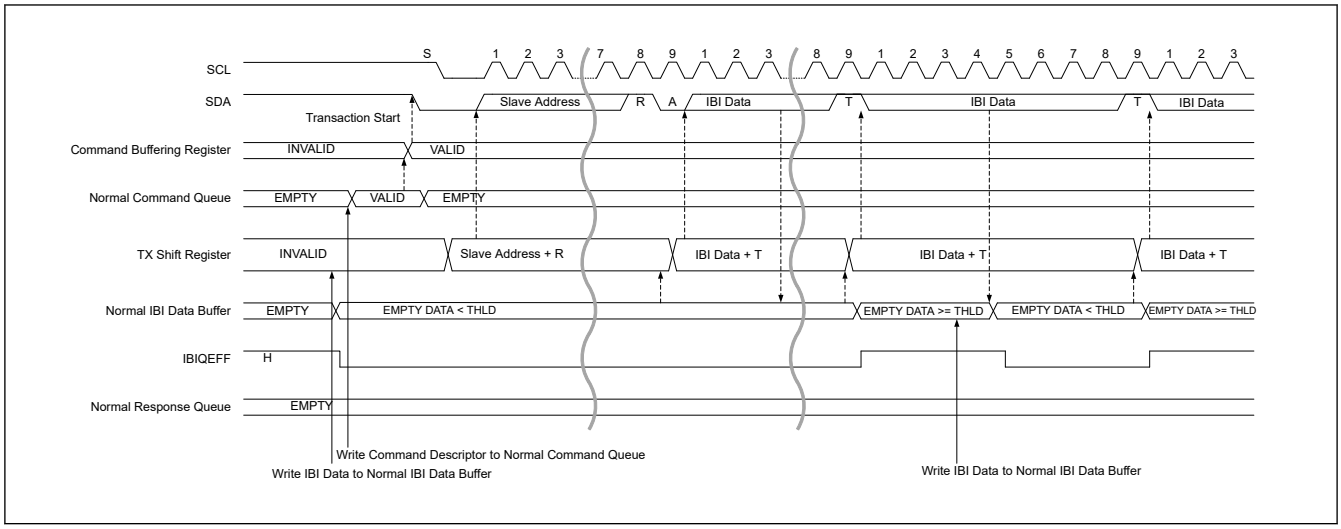


Figure 33.63 I3C slave IBI transfer timing (1/2)

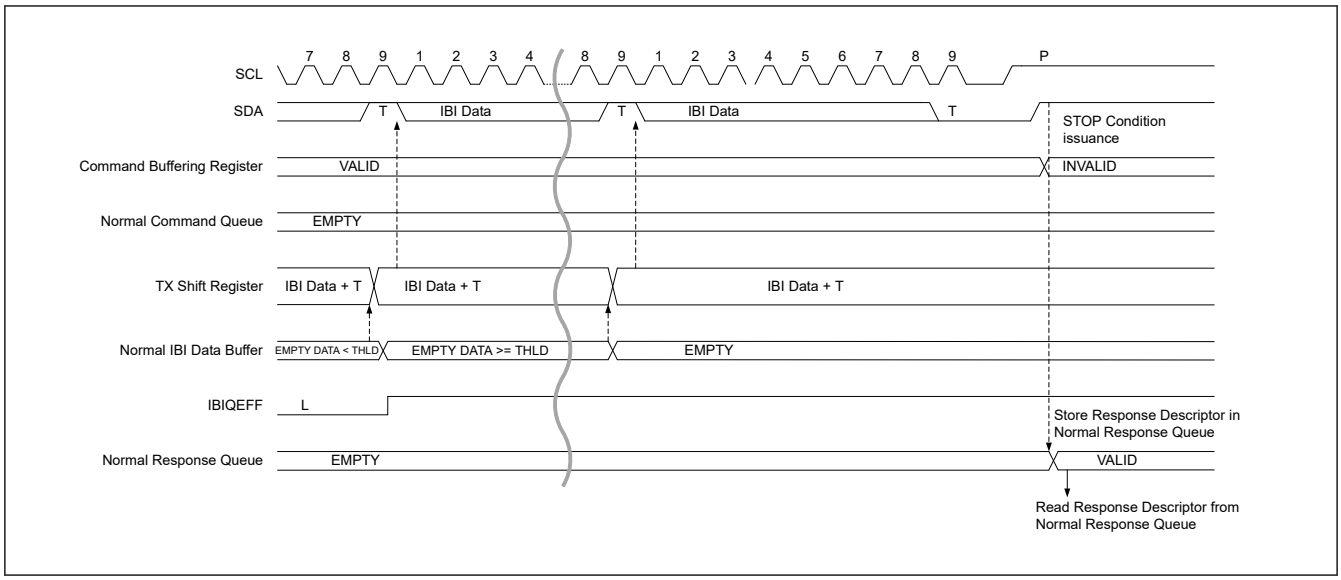


Figure 33.64 I3C slave IBI transfer timing (2/2)

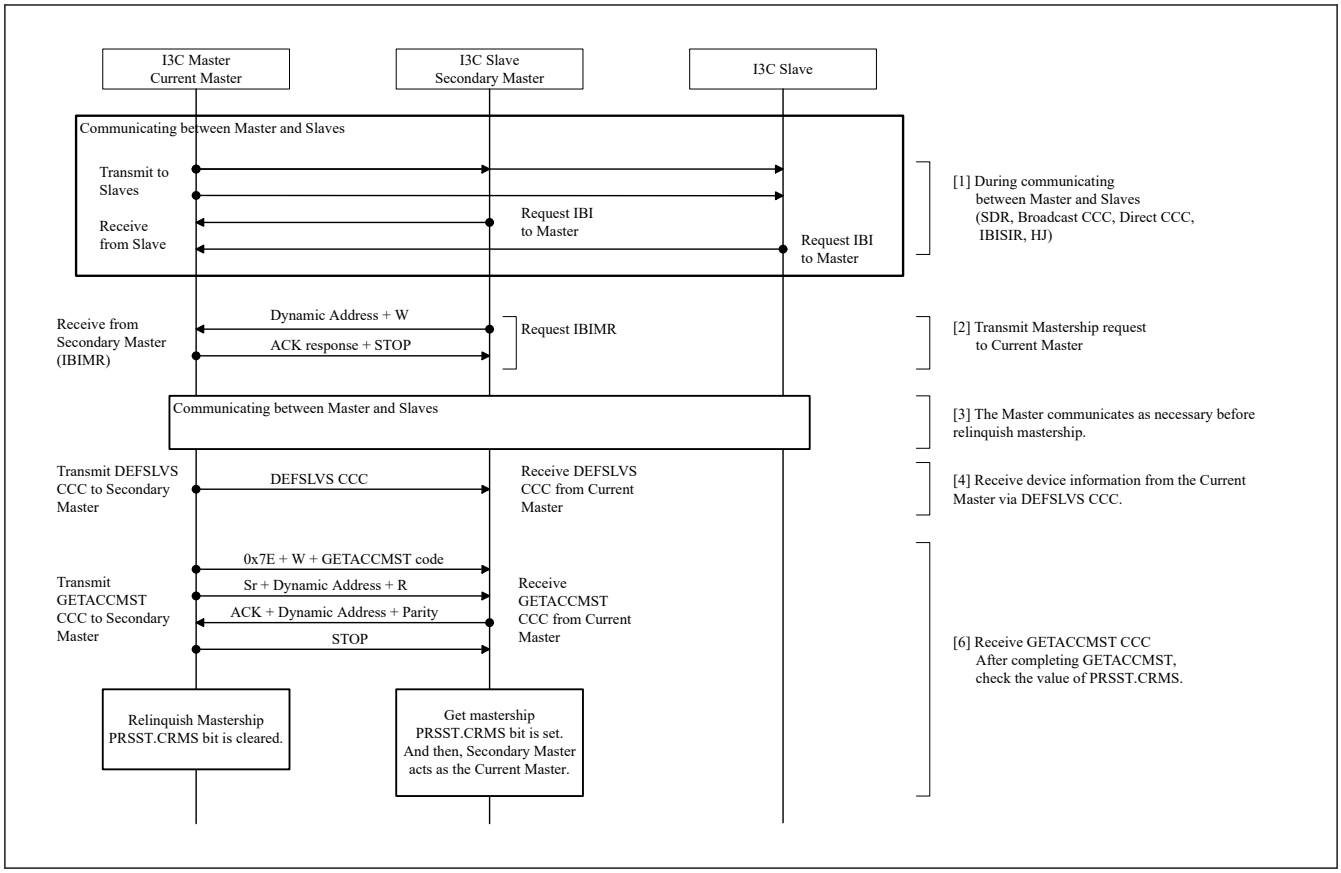


Figure 33.65 I3C Slave Mastership processing flow

33.3.2.2 Data Handler

The relationship between the transfer method and the queue is shown in Table 33.10.

Table 33.10 Transfer method and queue

Protocol	Transfer method	Queue/Buffer	size	Master	Slave	Secondary Master
I²C Mode	Single buffer transfer	Normal Tx Data	1 byte	✓	✓	—
		Normal Rx Data	1 byte	✓	✓	—
I3C Mode	Normal FIFO buffer transfer	Normal Command	4 QUEUEs	✓	✓	✓
		Normal Response	4 QUEUEs	✓	✓	✓
		Normal Tx Data	16 DWORDs	✓	✓	✓
		Normal Rx Data	16 DWORDs	✓	✓	✓
		Normal Receive Status	2 QUEUEs	—	✓	✓
		Normal IBI Status	2 QUEUEs	✓	—	✓
		Normal IBI Data	8 DWORDs	✓	✓	✓
	High Priority FIFO buffer transfer (in Master Mode only)	High Priority Command	2 QUEUEs	✓	—	✓
		High Priority Response	2 QUEUEs	✓	—	✓
		High Priority Tx Data	2 DWORDs	✓	—	✓
		High Priority Rx Data	2 DWORDs	✓	—	✓

33.3.2.2.1 Transfer Method in I²C Mode

(1) Single Buffer transfer

Each process (condition issue, data transfer, ACK / NACK response) is controlled by software.

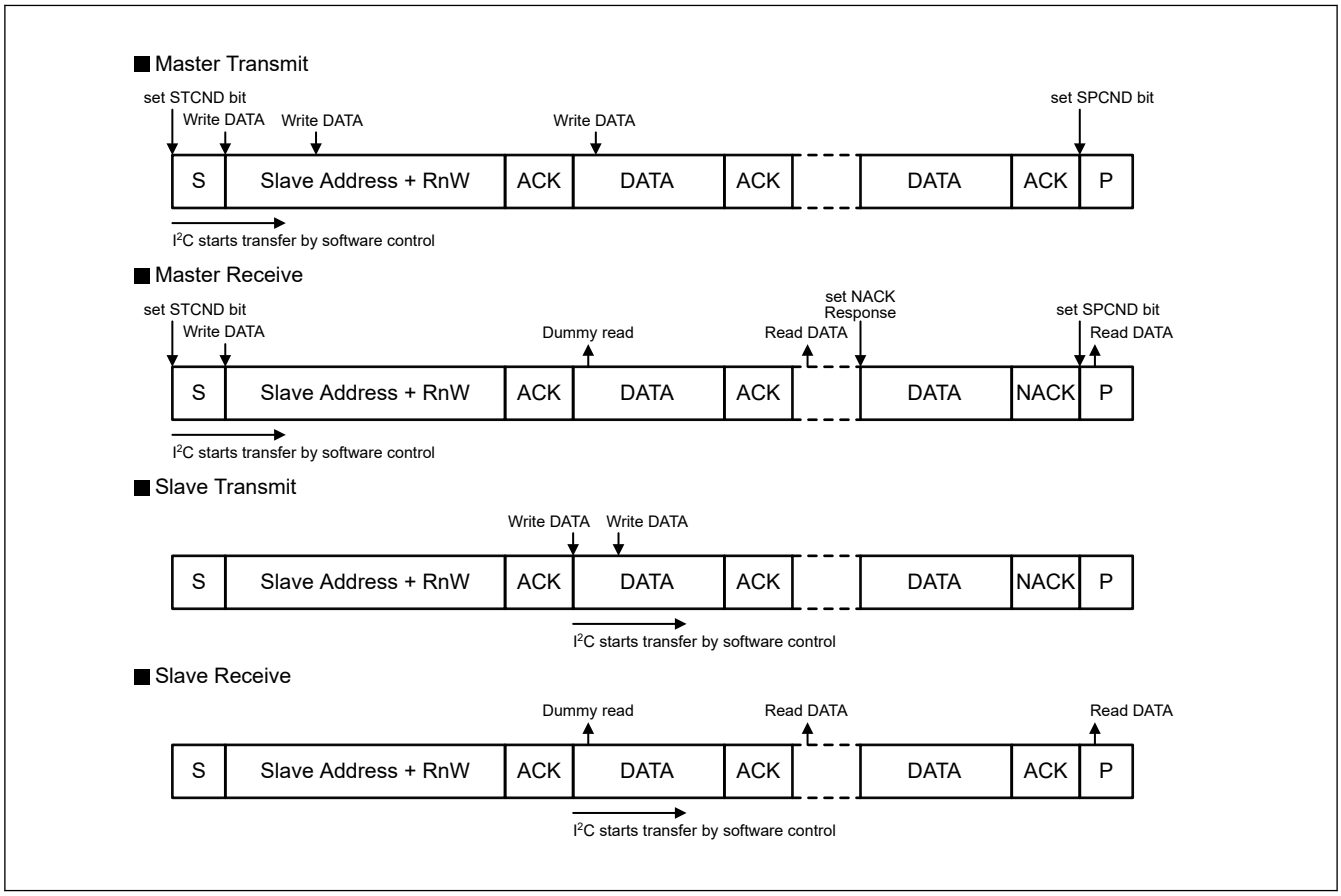


Figure 33.66 Data handler with single buffer transfer

33.3.2.2.2 Transfer Method in I3C Mode

(1) Normal FIFO Buffer Transfer

I3C autonomously starts transfer when data and command are written.

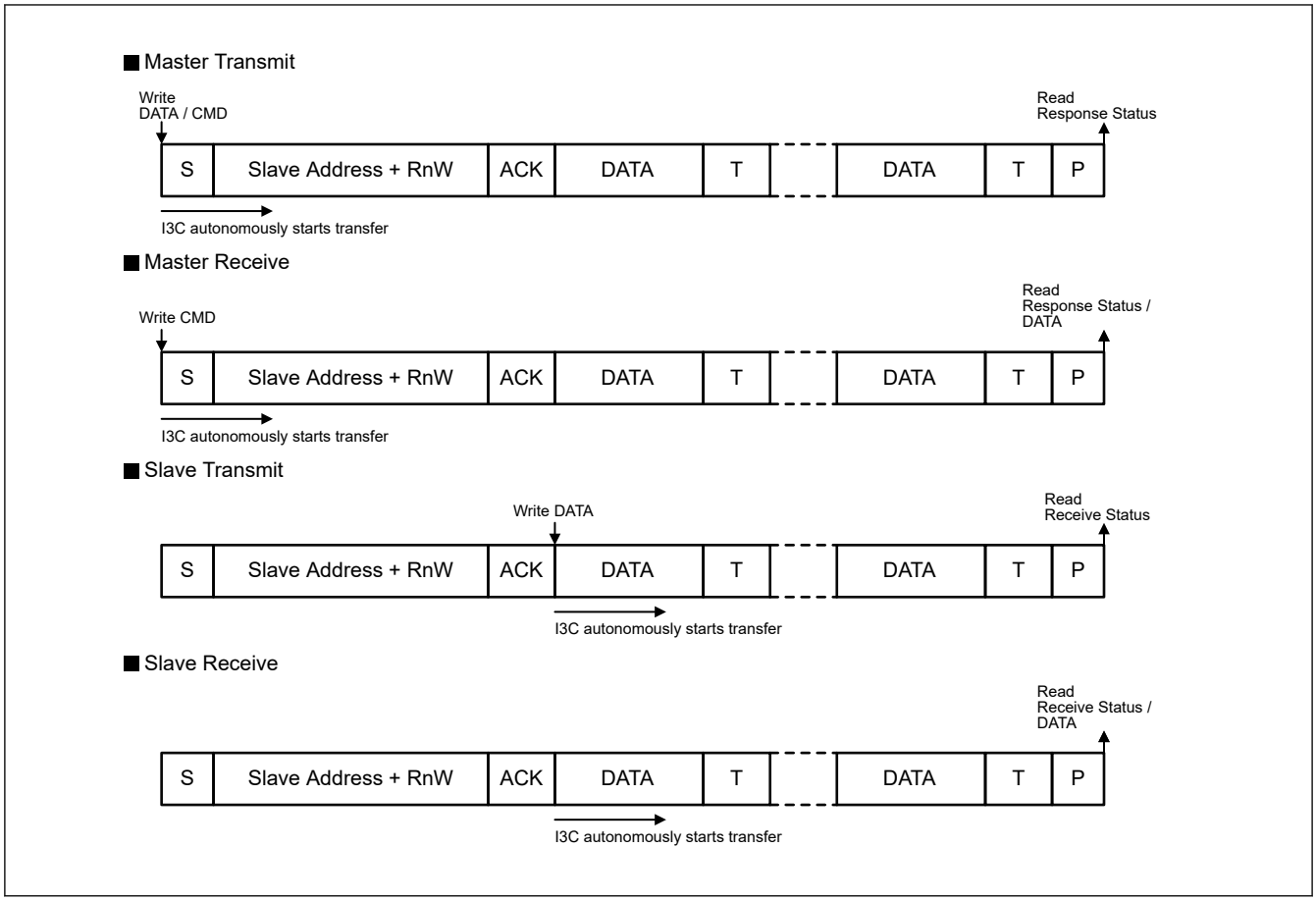


Figure 33.67 Data handler with normal FIFO buffer transfer

(2) High Priority FIFO Buffer Transfer

I3C handles the command of the High Priority FIFO buffer transfer higher priority than the command of the normal FIFO buffer transfer.

If data and commands are written to the High Priority FIFO buffer during normal FIFO buffer transfer, I3C waits for the STOP condition and then processes the command in the High Priority FIFO buffer.

After the command processing in the High Priority FIFO buffer is completed, if the command remains in the normal FIFO buffer, the I3C resumes processing the command in the normal FIFO buffer.

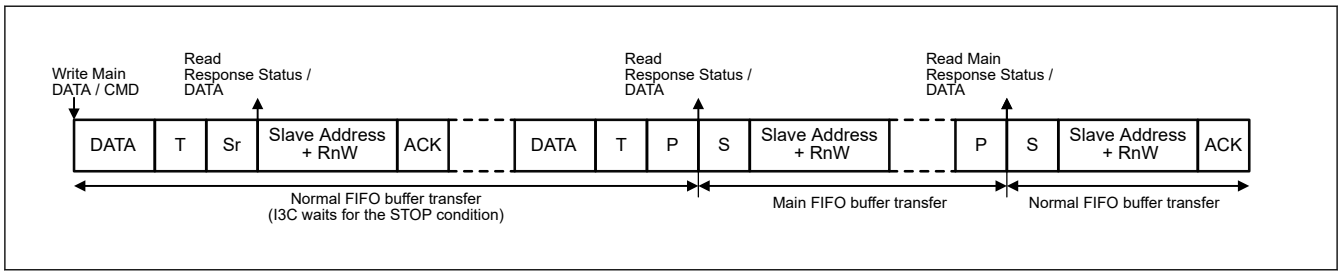


Figure 33.68 Data handler with high priority FIFO buffer transfer

33.3.2.3 I²C/I3C Protocol

33.3.2.3.1 Communication Protocol

(1) I²C Communication Data Format

The I²C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a START condition or Repeated START condition is an address frame used to specify a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a STOP condition is issued.

Figure 33.69 shows the I²C bus format, and Figure 33.70 shows the I²C bus timing.

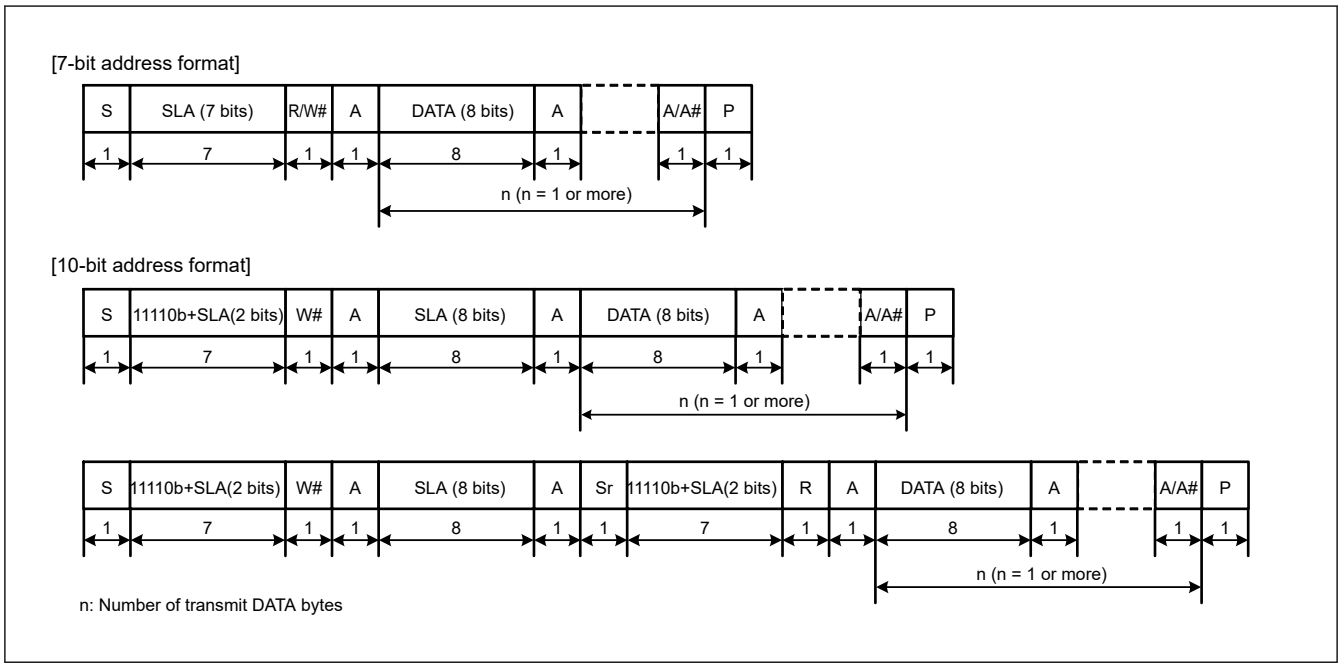


Figure 33.69 I²C bus format

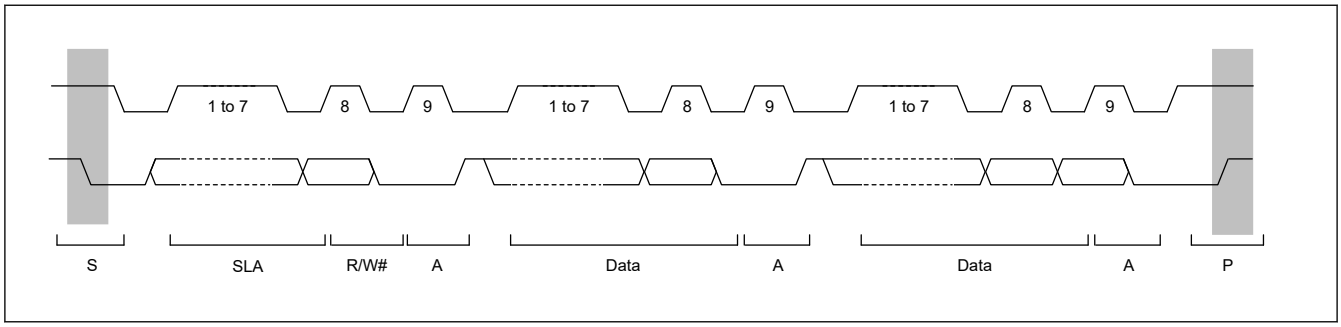


Figure 33.70 I²C bus timing (SLA = 7 bits)

S:	START condition. The master device drives the I3C_SDA line low from high level while the I3C_SCL line is at a high level.
SLA:	Slave address, by which the master device selects a slave device.
R/W#:	Indicates the direction of data transfer: from the slave device to the master device when R/W = 1, or from the master device to the slave device when R/W = 0.
A:	Acknowledge. The receive device drives the I3C_SDA line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)
A#:	Not Acknowledge. The receive device drives the I3C_SDA line high.
Sr:	Repeated START condition. The master device drives the I3C_SDA line low from the high level after the setup time has elapsed with the I3C_SCL line at the high level.

DATA:	Transmitted or received data
P:	STOP condition. The master device drives the I3C_SDA line high from low level while the I3C_SCL line is at a high level.

(2) I3C Communication Data Format

Figure 33.71 through Figure 33.77Figure 33.74 illustrate a typical communication for each of the six I3C Protocols. While these diagrams do not exhaustively illustrate all possible I3C communications, they do serve as useful introductions to the signaling and transmission formatting used in each I3C Protocol.

Figure 33.71 illustrates example communication using I3C Single Data Rate (SDR) coding with Broadcast (0x7E). It shows the Master reading a byte of data from the Slave at Address 0x2B in SDR Mode. From the Bus Free Condition, the Master issues a START by driving the I3C_SDA line Low while keeping the I3C_SCL line High. It then issues the Broadcast Address (0x7E) followed by RnW (0 for Write). Then the Master turns on a pull-up resistor and goes to Open Drain.

All Slaves ACK by pulling the I3C_SDA line Low (in the Figure, pink fill means the Slave is in control of the I3C_SDA line at this time). The Master then issues a Repeated START, then the Address of the Slave (0x2B) it wants to read followed by RnW (1 for Read). The Master then turns on a pull-up resistor and goes to Open Drain, allowing the Slave to acknowledge by pulling the I3C_SDA line Low. At this point, the Master continues to toggle the I3C_SCL line and release the I3C_SDA line, allowing the Slave to drive SDA to send one byte of data (0x4A) followed by T. T = 1 informs the Master that there is additional data, whereas T = 0 signals the end. Here there is additional data, so the Slave drives SDA High until SCL goes High, at which time it releases SDA. The Master has the option of holding SDA High with a weak pullup, which signals to the Slave that the Master allows another byte to be transmitted, or to pull SDA Low (while SCL is High – hence a Repeated START), which would signal to the Slave that the Master has terminated the Read and is taking over.

SDR Mode is backwards compatible with Legacy I2C Devices, because the High time of an SCL pulse is always less than 50ns and therefore SCL will always appear to be Low because of the I2C 50ns Spike Filter.

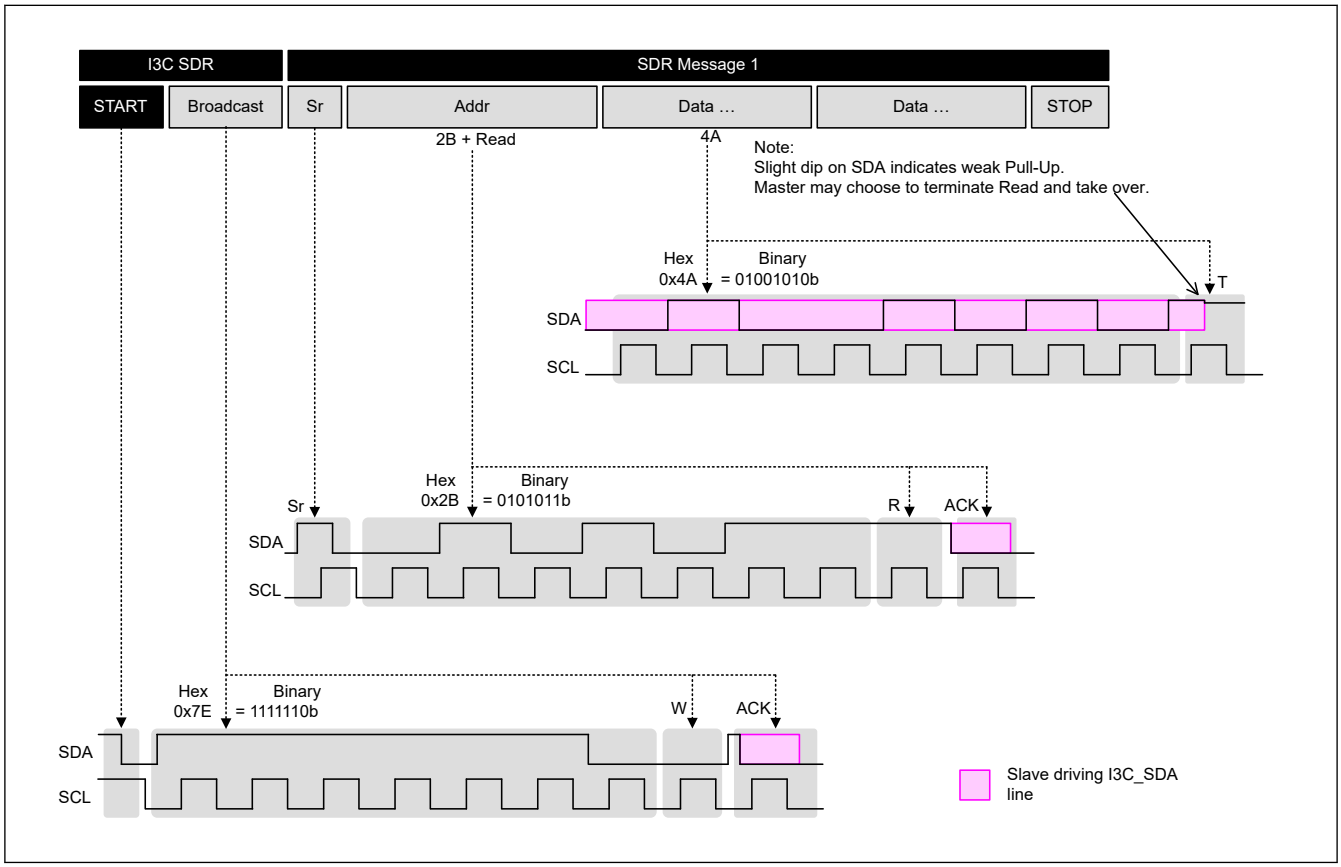


Figure 33.71 Example communication using I3C coding SDR with broadcast (0x7E)

Figure 33.72 illustrates example communication using I3C Single Data Rate (SDR) coding without Broadcast (0x7E). It shows the Master reading a byte of data from the Slave at Address 0x2B in SDR Mode. From the Bus Free Condition, The Master then issues a START, then the Address of the Slave (0x2B) it wants to read followed by RnW (1 for Read).

The Master then turns on a pull-up resistor and goes to Open Drain, allowing the Slave to acknowledge by pulling the I3C_SDA line Low. At this point, the Master continues to toggle the I3C_SCL line and release the I3C_SDA line, allowing the Slave to drive SDA to send one byte of data (0x4A) followed by T. T = 1 informs the Master that there is additional data, whereas T = 0 signals the end. Here there is additional data, so the Slave drives SDA High until SCL goes High, at which time it releases SDA. The Master has the option of holding SDA High with a weak pull-up, which signals to the Slave that the Master allows another byte to be transmitted, or to pull SDA Low (while SCL is High – hence a Repeated START), which would signal to the Slave that the Master has terminated the Read and is taking over.

SDR Mode is backwards compatible with Legacy I2C Devices, because the High time of an SCL pulse is always less than 50ns and therefore SCL will always appear to be Low because of the I2C 50ns Spike Filter.

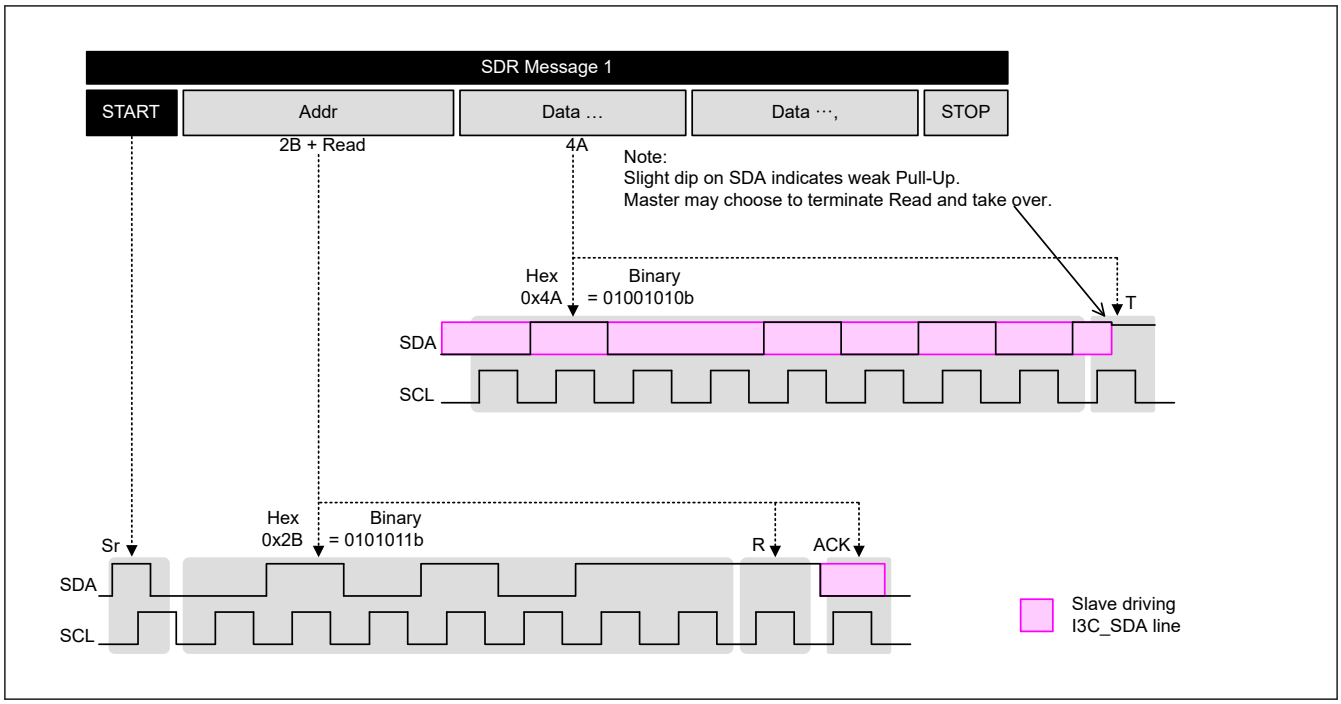


Figure 33.72 Example communication using I3C coding SDR without broadcast (0x7E)

Figure 33.73 shows the Master issuing a CCC Direct Command to a single Slave. This particular command (GETPID) reads the Provisional ID of a Slave.

From the Bus Free Condition, the Master issues a START by driving the I3C_SDA line Low while keeping the I3C_SCL line High. It then issues the Broadcast Address (0x7E) followed by RnW (0 for Write). Then the Master turns on a pull-up resistor and goes to Open Drain. All Slaves ACK by pulling SDA Low (in the Figure, pink fill means the Slaves are in control of SDA at this time). The Master then issues the Direct Common Command Code for GETPID (0x8C) followed by parity bit T (odd parity = 0 for 0x8C) then the 7-bit Dynamic Address of the Slave (chosen arbitrarily here to be 0x2B) followed by a RnW bit (1 for Read). Then the Master turns on a pull-up resistor and goes to Open Drain, allowing the Slave at Address 0x2B to ACK by pulling SDA Low, which tells the Master that the Slave Acknowledges the command and will comply. (Alternatively, the Slave may NACK by not pulling SDA Low, which would inform the Master that the Slave will not comply – in this case, that an error occurred.) Following the ACK the Slave outputs its 48-bit PID one byte at a time, and then the Master issues a Repeated START (this part of the waveform sequence is not shown in the Figure).

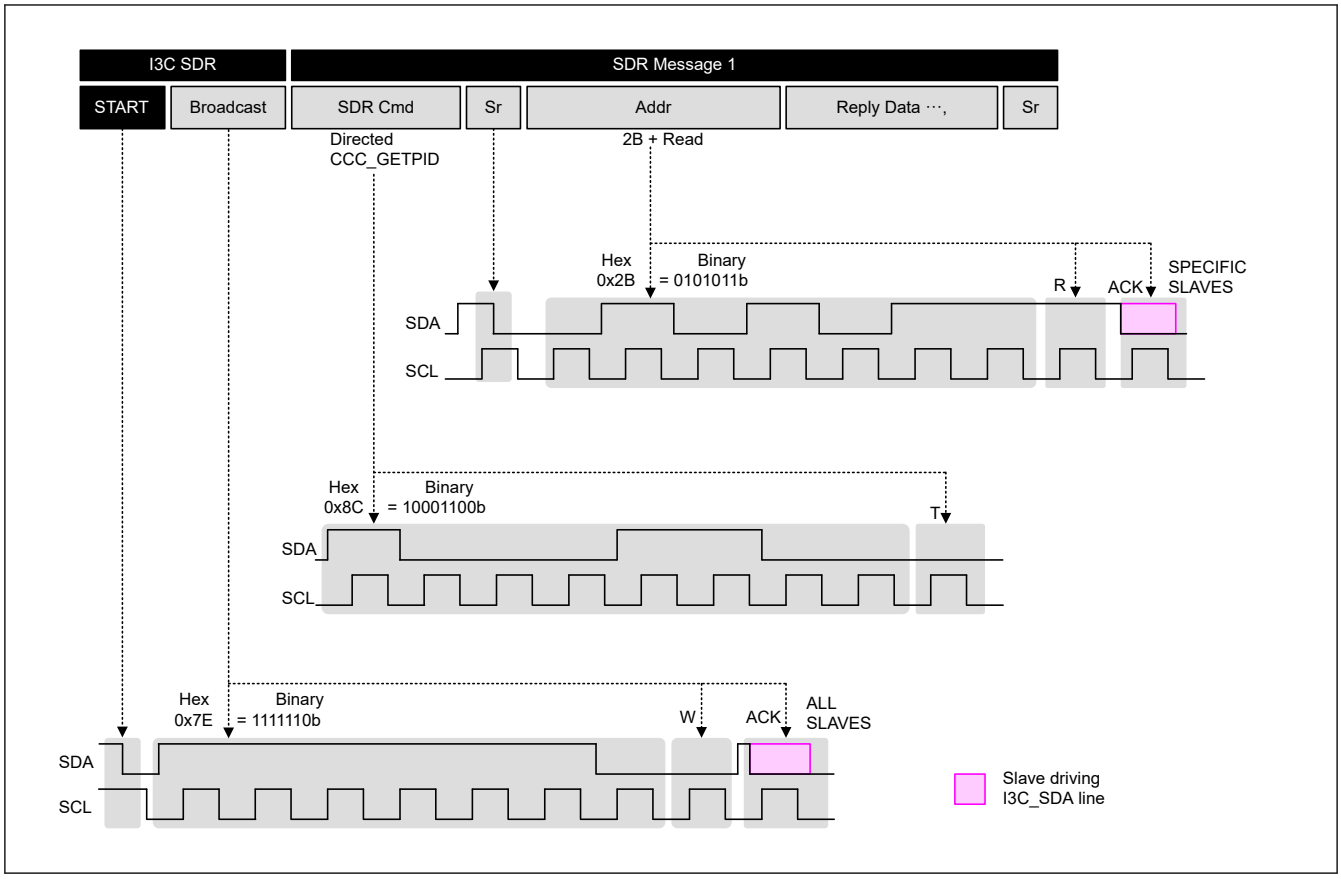


Figure 33.73 Example communication using I3C coding SDR with CCC direct addressing

Figure 33.74 illustrates example SDR communication with a CCC Broadcast command. The command used in this example sets the Maximum Read Length of all Slaves to 43 bytes (0x002B).

From the Bus Free Condition, the Master issues a START by driving the I3C_SDA line Low while keeping the I3C_SCL line High. It then issues the Broadcast Address (0x7E) followed by RnW (0 for Write). Then the Master turns on a pull-up resistor and goes to Open Drain. All Slaves ACK by pulling SDA Low (in the Figure, pink fill means the Slaves are in control of SDA at this time). The Master then issues the Broadcast Common Command Code for SETMRL (0x09) followed by parity bit T (odd parity = 1 for 0x09), and then 2 data bytes (MSB first) to define the maximum number of bytes which can be read from a Slave in a single read operation. Each data byte is followed by a T bit (parity bit – odd parity). After this the Master issues a Repeated START.

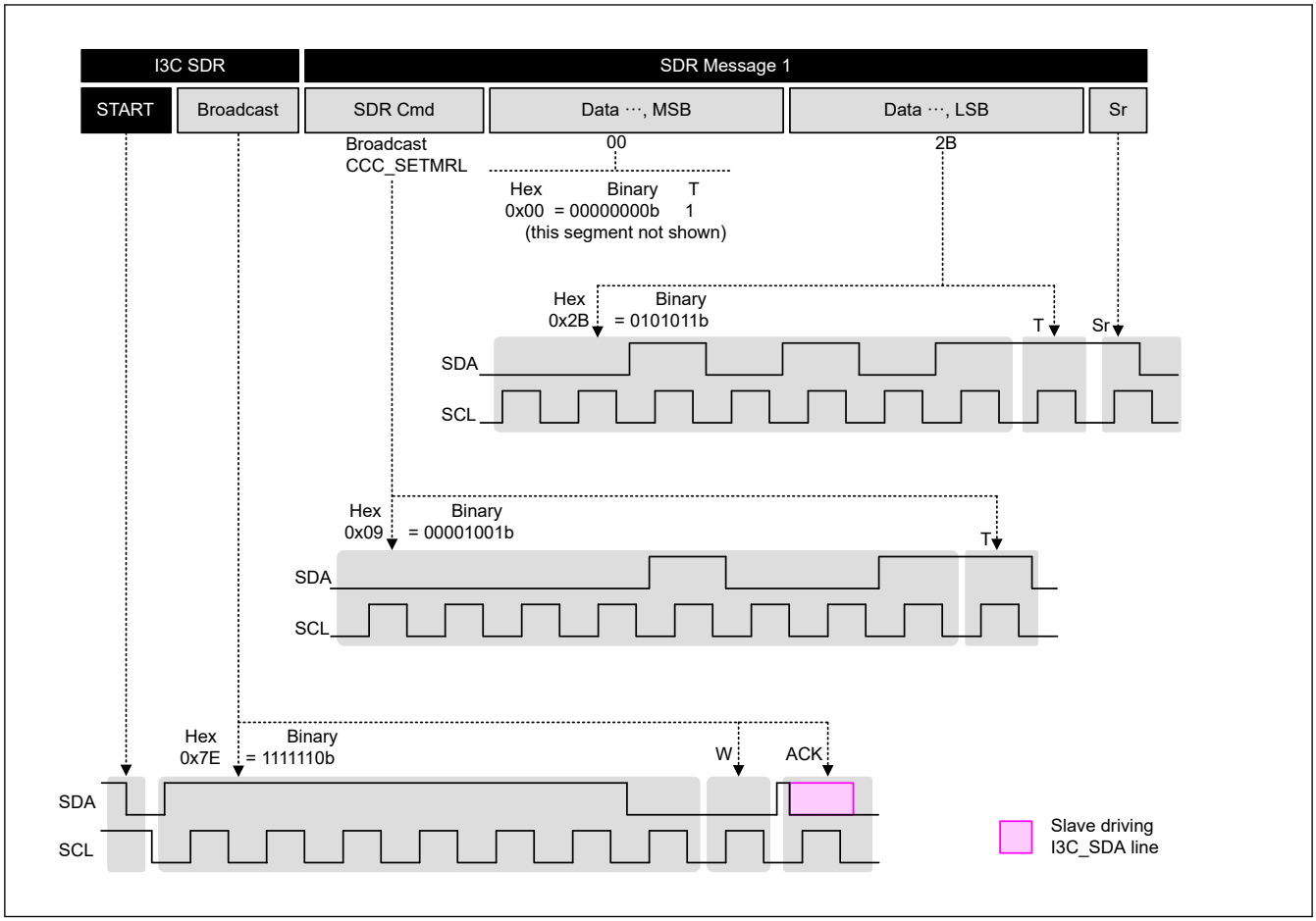


Figure 33.74 Example communication using I3C coding SDR with CCC broadcast

Figure 33.75 illustrates use of the HDR-DDR (Double Data Rate) Mode. It shows how the Master can change the Mode from SDR (Single Data Rate) Mode to HDR-DDR Mode, and a sample of the HDR-DDR data format.

From the Bus Free Condition, the Master issues a START by driving the I3C_SDA line Low while keeping the I3C_SCL line High. The Master then issues the Broadcast Address (0x7E) followed by RnW (0 for Write). Then the Master turns on a pull-up resistor and goes to Open Drain. All Slaves 'ACK' by pulling SDA Low (in the Figure, pink fill means the Slaves are in control of SDA at this time). The Master then issues the Broadcast Common Command Code for ENTHDR0 (0x20), followed by parity bit T (odd parity = 0 for 0x20). At this point, the Bus is in HDR-DDR Mode. In the HDR-DDR protocol, the I3C_SDA line is sampled on every SCL edge (both Low-to-High and High-to-Low transitions of SCL). The HDR-DDR Word consists of a 2-bit Preamble, followed by two bytes of data, followed by two parity bits. The waveform for the 5-bit CRC and following traffic is not shown in the Figure.

HDR-DDR Mode is backwards compatible with Legacy I^2C Devices, because the High time of an SCL pulse is always less than 50ns and as a result SCL will always appear to be Low because of the I^2C 50ns Spike Filter.

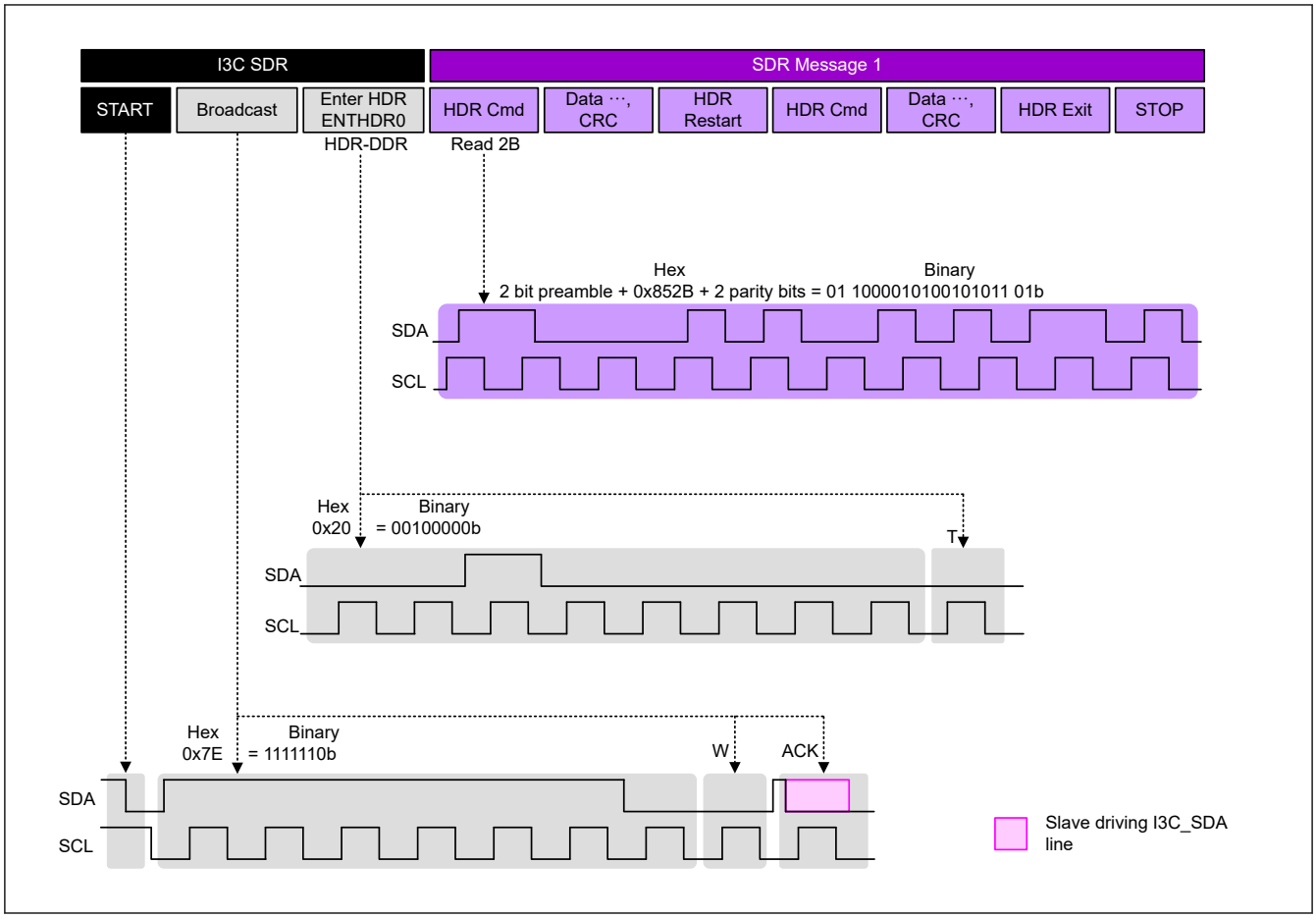


Figure 33.75 Example communication using HDR-DDR protocol

Figure 33.76 illustrates use of the HDR-TSL (Ternary Symbol Legacy) Mode, which is only used if a Legacy I^{2}C Device is present on the Bus. It shows how the Master can change the Mode from SDR (Single Data Rate) Mode to HDR-TSL Mode, and a sample of the HDR-TSL data format.

From the Bus Free Condition, the Master issues a START by driving the I3C_SDA line Low while keeping the I3C_SCL line High. It then issues the Broadcast Address (0x7E) followed by RnW (0 for Write). Then the Master turns on a pull-up resistor and goes to Open Drain. All Slaves ACK by pulling SDA Low (in the Figure, pink fill means the Slaves are in control of SDA at this time). The Master then issues the Broadcast Common Command Code for ENTHDR2 (0x22) followed by parity bit 'T' (odd parity = 1 for 0x22). At this point, the Bus is in HDR-TSL Mode. In this protocol, the I3C_SCL line is more than just the clock. Both SDA transitions and SCL transitions carry data. The ternary conversion can be understood by considering that each octal digit is equivalent to two ternary Symbols. For example, for octal sequence 412255 we have: 4 = 3 + 1, 1 = 0 + 1, 2 = 0 + 2, 2 = 0 + 2, 5 = 3 + 2, 5 = 3 + 2. The clock pulse occurs when SCL transitions, when SDA transitions, or when both lines transition simultaneously.

HDR-TSL Mode is backwards compatible with Legacy I^{2}C Devices, because the High time of an SCL pulse is always less than 50ns, and as a result SCL will always appear to be Low due to the I^{2}C 50ns Spike Filter. Note that in the Figure, the thick blue High-to-Low transitions of SCL are dummy edges added in order to make sure that SCL pulses are always less than 50ns. High-to-Low SCL transitions that occur without a simultaneous SDA transition are always dummy transitions.

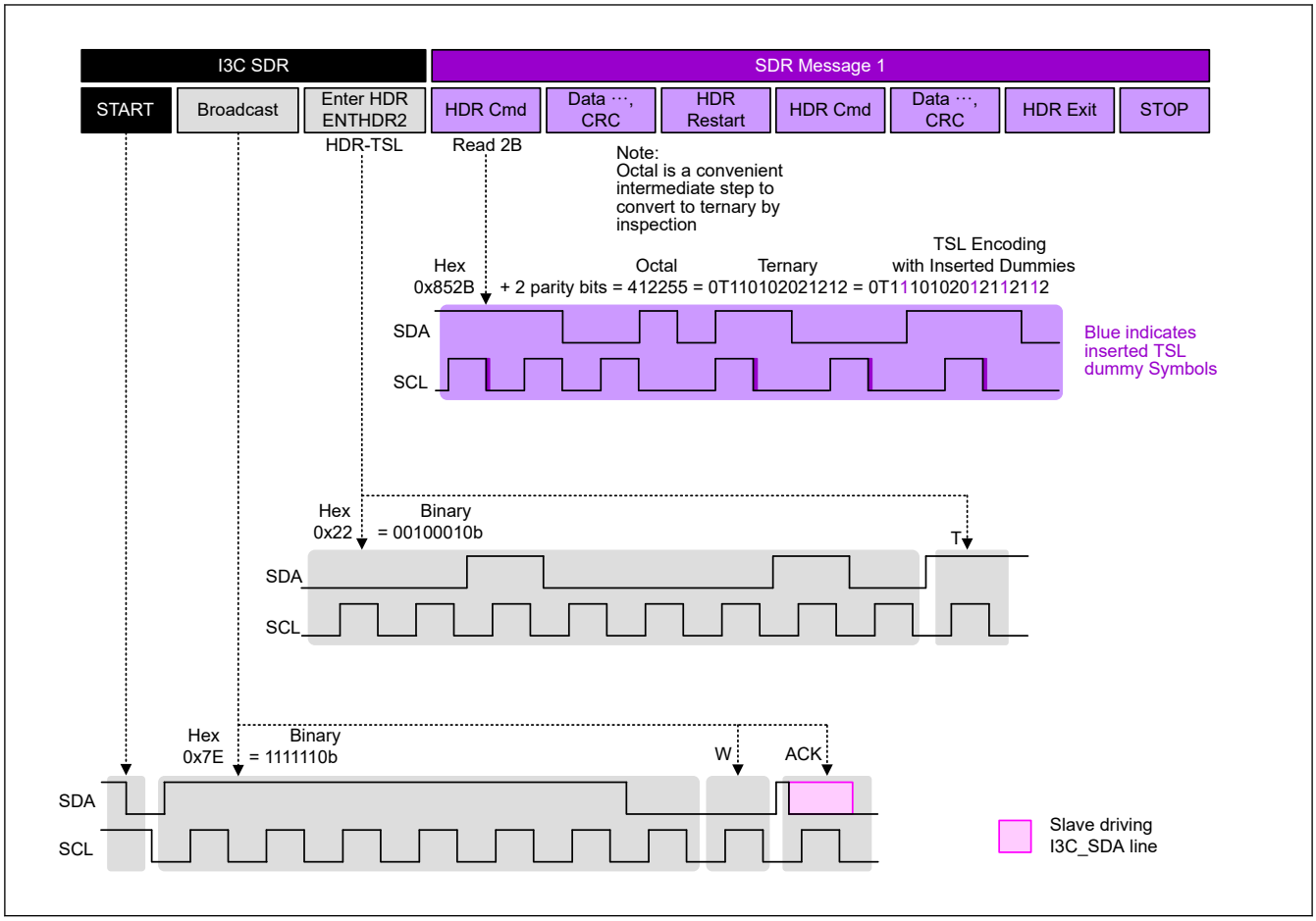


Figure 33.76 Example communication using HDR-TSL protocol

Figure 33.77 illustrates use of HDR-TSP (High Data Rate – Ternary Symbol for Pure Bus) Mode. This protocol is only used when there are no I^2C Devices on the Bus. The Figure shows how the Master can change the Mode from SDR (Single Data Rate) to HDR-TSP Mode, and a sample of the TSL data format.

From the Bus Free Condition, the Master issues a START by driving the I3C_SDA line Low while keeping the I3C_SCL line High. The Master then issues the Broadcast Address (0x7E) followed by RnW (0 for Write). Then the Master turns on a pull-up resistor and goes to Open Drain. All Slaves ACK by pulling SDA Low (in the Figure, pink fill means the Slaves are in control of SDA at this time). The Master then issues the Broadcast Common Command Code for ENTHDR1 (0x21) followed by parity bit 'T' (odd parity = 1 for 0x21). At this point, the Bus is in HDR-TSP Mode. In the HDRTSP protocol, the SCL is more than just the clock. Both SDA transitions and SCL transitions carry data. The ternary conversion can be understood by considering that each octal digit is equivalent to two ternary Symbols. For example, for octal sequence 412255 (4 = 3 + 1, 1 = 0 + 1, 2 = 0 + 2, 2 = 0 + 2, 5 = 3 + 2, 5 = 3 + 2). The clock pulse occurs when SCL transitions, when SDA transitions, or when both lines transition simultaneously.

HDR-TSP Mode supports the highest data-rate and uses the least energy per byte, however it is not backwards compatible with Legacy I^2C Devices since the High duration of an SCL pulse can exceed 50ns.

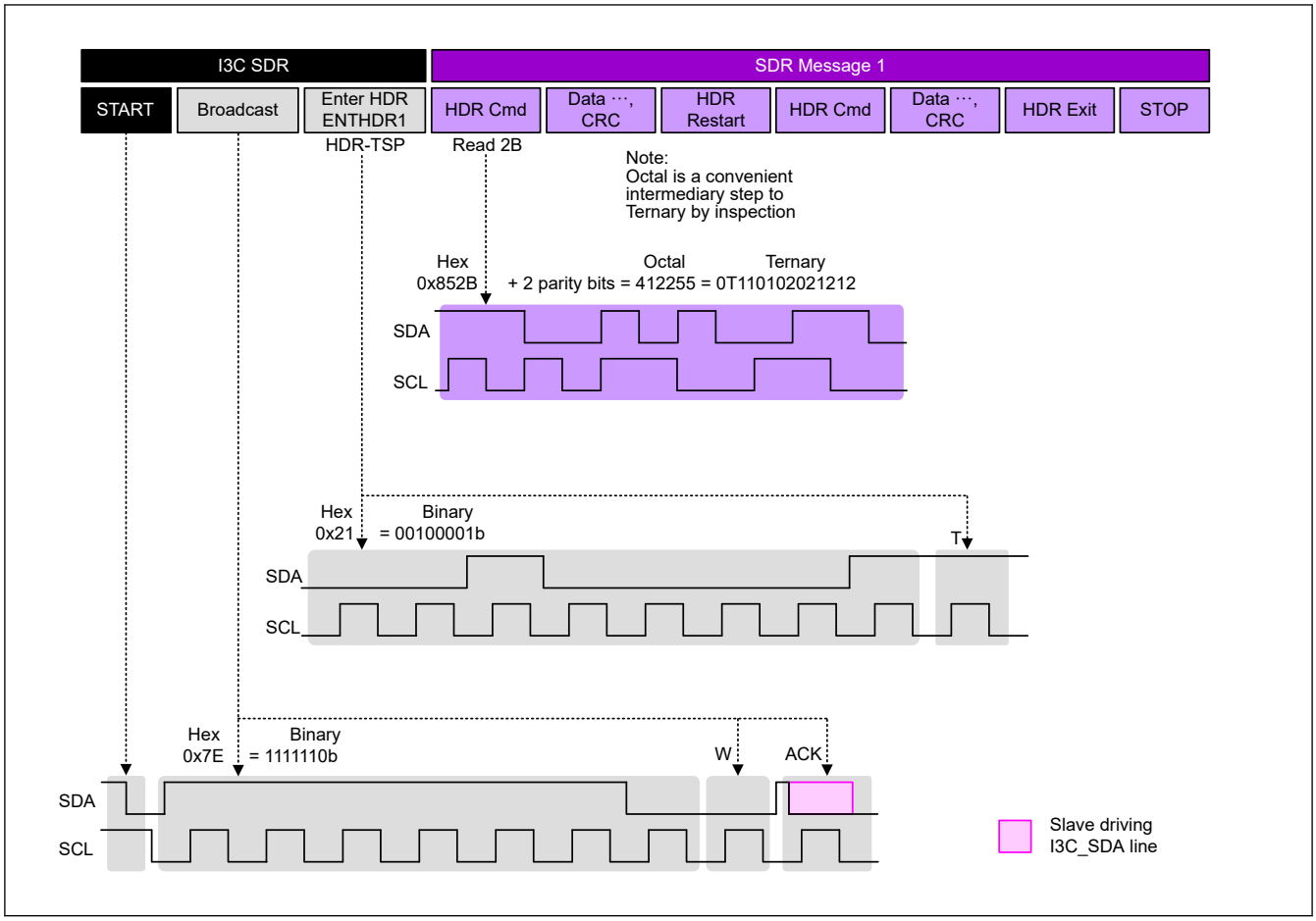


Figure 33.77 Example communication using HDR-TSP protocol

33.3.2.3.2 Bus Conditions

I3C defines three distinct conditions in which the I3C Bus shall be considered inactive: Bus Free, Bus Available, and Bus Idle (see Figure 33.78).

(1) Bus Free Condition

State on the I3C Bus where both the I3C_SCL line and the I3C_SDA line are High for at least the period set by BFRECDT.FRECYC[8:0] bit.

(2) Bus Available Condition [I3C mode]

State on the I3C Bus where both the I3C_SCL line and the I3C_SDA line are High for at least the period set by BAVLCDT.AVLCYC[8:0] bit.

A Slave may only issue a START Request (For example, for an In-Band Interrupt, or for a Master Handoff Request) after a Bus Available Condition.

(3) Bus Idle Condition [I3C mode]

State on the I3C Bus where both the I3C_SCL line and the I3C_SDA line are High for at least the period set by BIDLCDT.IDLCYC[17:0] bit.

A Slave may only issue a START Request after a Bus Idle Condition.

Specifications are as follows. IDLE needs to be the largest.

BFRECDT.FRECYC[8:0] < BAVLCDT.AVLCYC[8:0] < BIDLCDT.IDLCYC[17:0]

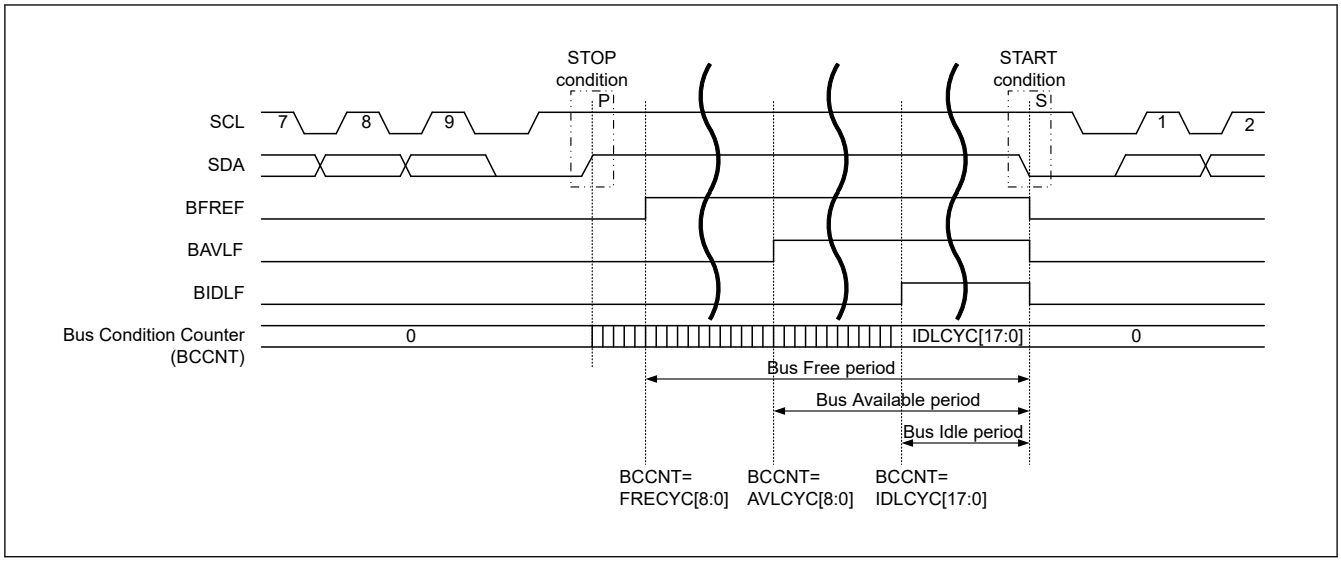


Figure 33.78 Bus conditions

33.3.2.3.3 START Condition / Repeated START Condition / STOP Condition Issuing Function

(1) Issuing a START Condition

I3C issues a START condition when the CNDCTL.STCND bit is set to 1.

Set the STCND bit to 1 (START condition issuance request) when the BCST.BFREF flag is set to 1 (bus free state).

I3C issues a START condition.

When a START condition is issued normally, I3C automatically shifts to the master transmit mode. A START condition is issued in the following sequence.

[START condition issuance]

- Drive the I3C_SDA line low (high level to low level).
- Ensure the time set in STDBR.SBRHO[7:0] and the START condition hold time.
- Drive the I3C_SCL line low (high level to low level).
- Detect low level of the I3C_SCL line and ensure the low-level period of I3C_SCL line set in STDBR.SBRLO[7:0].

(2) Issuing a Repeated START Condition

I3C issues a Repeated START condition when the CNDCTL.SRCND bit is set to 1.

When the SRCND bit is set to 1, a Repeated START condition issuance request is made and I3C issues a Repeated START condition when the BCST.BFREF flag = 0 (bus busy state) and the PRSST.CRMS bit = 1 (master mode).

A Repeated START condition is issued in the following sequence.

[Repeated START condition issuance]

- Release the I3C_SDA line.
- Ensure the low-level period of I3C_SCL line set in STDBR.SBRLO[7:0] or EXTBR.EBRLO[7:0] .
- Release the I3C_SCL line (low level to high level).
- Detect a high level of the I3C_SCL line and ensure the time set in STDBR.SBRLO[7:0] or EXTBR.EBRLO[7:0] and the Repeated START condition setup time.
- Drive the I3C_SDA line low (high level to low level).
- Ensure the time set in STDBR.SBRHO[7:0] or EXTBR.EBRHO[7:0] or EXTBR.EBRHO[7:0] and the Repeated START condition hold time.
- Drive the I3C_SCL line low (high level to low level).

- Detect a low level of the I3C_SCL line and ensure the low-level period of I3C_SCL line set in STDBR.SBRLO[7:0] or EXTBR.EBRLO[7:0] .

Note: When issuing Repeated START conditions request, write the slave address to NTDTBP0 after confirming CNDCTL.SRCND = 0. Data written in the period of CNDCTL.SRCND = 1 is not forwarded because retransmission condition before the occurrence.

To issue a Repeated START condition in Hs-mode, follow the steps below.

1. Wait for PRSTDBG.SCOLV=0.
2. Set EXTBR.EBRHO[7:0] to satisfy the hold time of the Repeated START condition.
3. Set the CNDCTL.SRCND bit to 1.
4. After confirming CNDCTL.SRCND=0, wait for PRSTDBG.SCOLV=0.
5. Set EXTBR.EBRHO [7: 0] according to the High period of the SCL clock in Hs-mode.
6. Write the slave address to NTDTBP0.

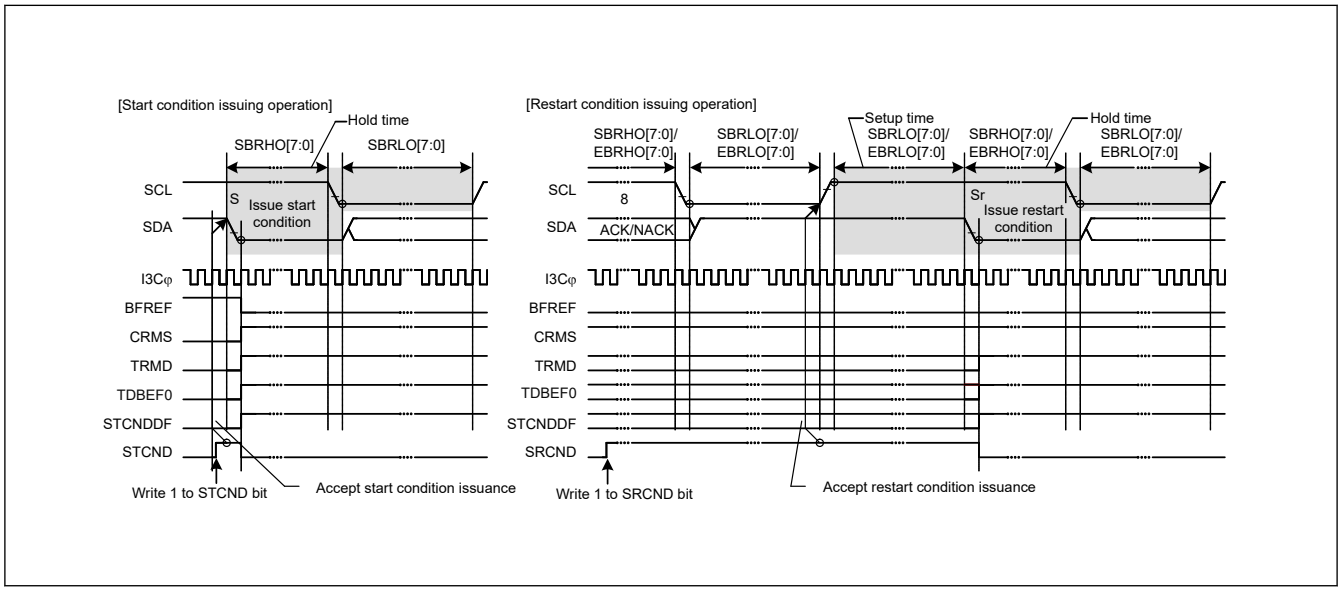


Figure 33.79 START condition / repeated START condition issue timing (STCND and SRCND bits)

Figure 33.80 shows the operation to issue a Repeated START condition after the master transmission.

[Repeated START condition issuance after the master transmission]

- Initial setting. For details, see section 33.3.3.1. Initial Setting Flow.
- Read the BFREF flag in BCST to check that the bus is open, and then set the STCND bit in CNDCTL to 1 (START condition issuance request). Upon receiving the request, I3C issues a START condition. At the same time, the BFREF flag is automatically set to 0 and the STCNDDF flag in BST is automatically set to 1 and the STCND bit is automatically set to 0. At this time, if the START condition is detected and the internal levels for the SDA output state and the levels on the I3C_SDA line have matched while the STCND bit = 1, I3C recognizes that issuing of the START condition as requested by the STCND bit has been successfully completed, and CRMS and TRMD bits in PRSST is automatically set to 1, placing I3C in master transmit mode. The NTST.TDBEF0 flag is also automatically set to 1 in response to setting of the TRMD bit to 1.
- Check that the NTST.TDBEF0 flag = 1, and then write the value for transmission (the slave address and the R/W# bit) to NTDTBP0. Once the data for transmission are written to NTDTBP0, the TDBEF0 flag is automatically set to 0, the data are transferred from NTDTBP0, and the TDBEF0 flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRMD bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, I3C continues in master transmit mode. Since the BST.NACKDF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to CNDCTL.SPCND bit to issue a STOP condition. For data transmission with an address in the 10-bit format, start by writing 1111 0, the 2 higher-order bits of the slave address, and W to NTDTBP0 as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to NTDTBP0.

- After confirming that the NTST.TDBEF0 flag = 1, write the data for transmission to the NTDTBP0 register. I3C automatically holds the I3C_SCL line low until the data for transmission are ready, a Repeated START condition is issued or a STOP condition is issued.
- After all bytes of data for transmission have been written to the NTDTBP0 register, wait until the value of the BST.TENDF flag returns to 1, and then, after check that the BST.STCNDDF flag = 1, set the BST.STCNDDF flag to 0.
- Set the SRCND bit in CNDCTL to 1 (Repeated START condition issuance request). Upon receiving the request, I3C issues a Repeated START condition.
- After check that the BST.STCNDDF flag = 1, write the value for transmission (the slave address and the R/W# bit) to NTDTBP0.

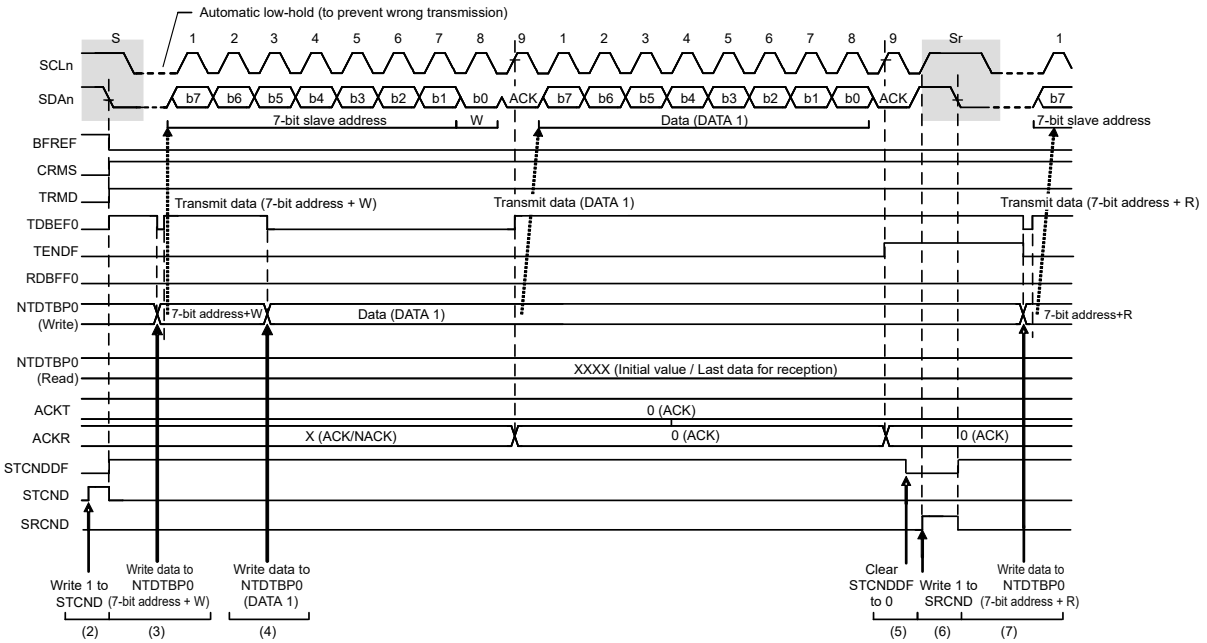


Figure 33.80 Repeated START condition issuance after the master transmission timing

(3) Issuing a STOP Condition

I3C issues a STOP condition when the SPCND bit in CNDCTL is set to 1.

When the SPCND bit is set to 1, a STOP condition issuance request is made and I3C issues a STOP condition when the BCST.BFREF flag = 0 (bus busy state) and the PRSST.MST bit = 1 (master mode).

A STOP condition is issued in the following sequence.

[STOP condition issuance]

- Drive the I3C_SDA line low (high level to low level).
- Ensure the low-level period of I3C_SCL line set in STDBR.SBRLO[7:0] or EXTBR.EBRLO[7:0] .
- Release the I3C_SCL line (low level to high level).
- Detect a high level of the I3C_SCL line and ensure the time set in STDBR.SBRHO[7:0] or EXTBR.EBRHO[7:0] and the STOP condition setup time.
- Release the I3C_SDA line (low level to high level).
- Ensure the time set in BFRECDT.FRECYC[8:0] and the bus free time.
- Set the BFREF flag to 1 (to release the bus mastership).

Note: To issue a STOP condition in Hs-mode, follow the steps below.

1. Wait for PRSTDBG.SCOLV=0.

2. Set EXTBR.EBRHO[7:0] to satisfy the setup time for the STOP condition.
3. Set the CNDCTL.SPCND bit to 1.
4. Wait for CNDCTL.SPCND=0.
5. Set EXTBR.EBRHO [7:0] according to the High period of the SCL clock in Hs-mode.

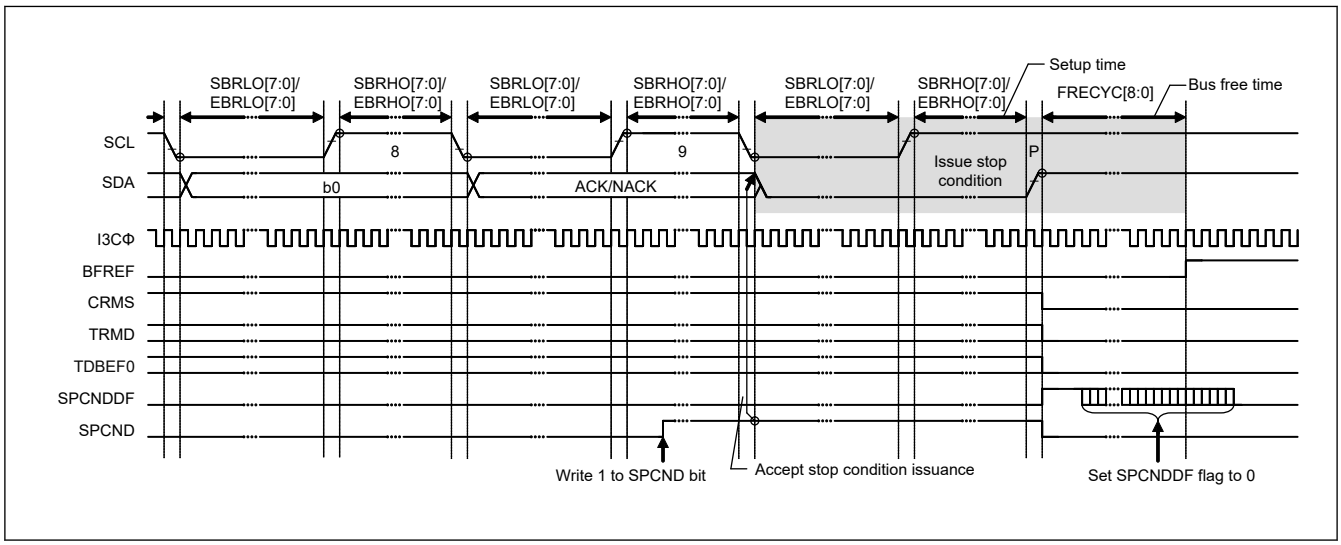


Figure 33.81 STOP condition issue timing (SPCND bit)

33.3.2.3.4 Address Match Detection

I3C can set three unique slave addresses in addition to the general call address and host address, and also can set 7- bit or 10- bit slave addresses.

(1) Slave-Address Match Detection [I^2C mode]

I3C can set three unique slave addresses, and has a slave address detection function for each unique slave address.

When the SVCTL.SVAEy bit (y = 0 to 2) is set to 1, the slave addresses set in the SVDVADy register (y = 0 to 2) can be detected.

When I3C detects a match of the set slave address, the corresponding SVST.SVAF[y] flag (y = 0 to 2) is set to 1 at the rising edge of the ninth SCL clock cycle, and the NTST.RDBFF0 flag or the NTST.TDBEF0 flag is set to 1 by the following R/W# bit. This causes a Normal Rx Data buffer full interrupt (I3C_RX) or Normal Tx Data buffer empty interrupt (I3C_TX) to be generated. The SVAF[y] flag is used to identify which slave address has been specified.

Figure 33.82 to Figure 33.84 show the SVAF[y] flag set timing in three cases.

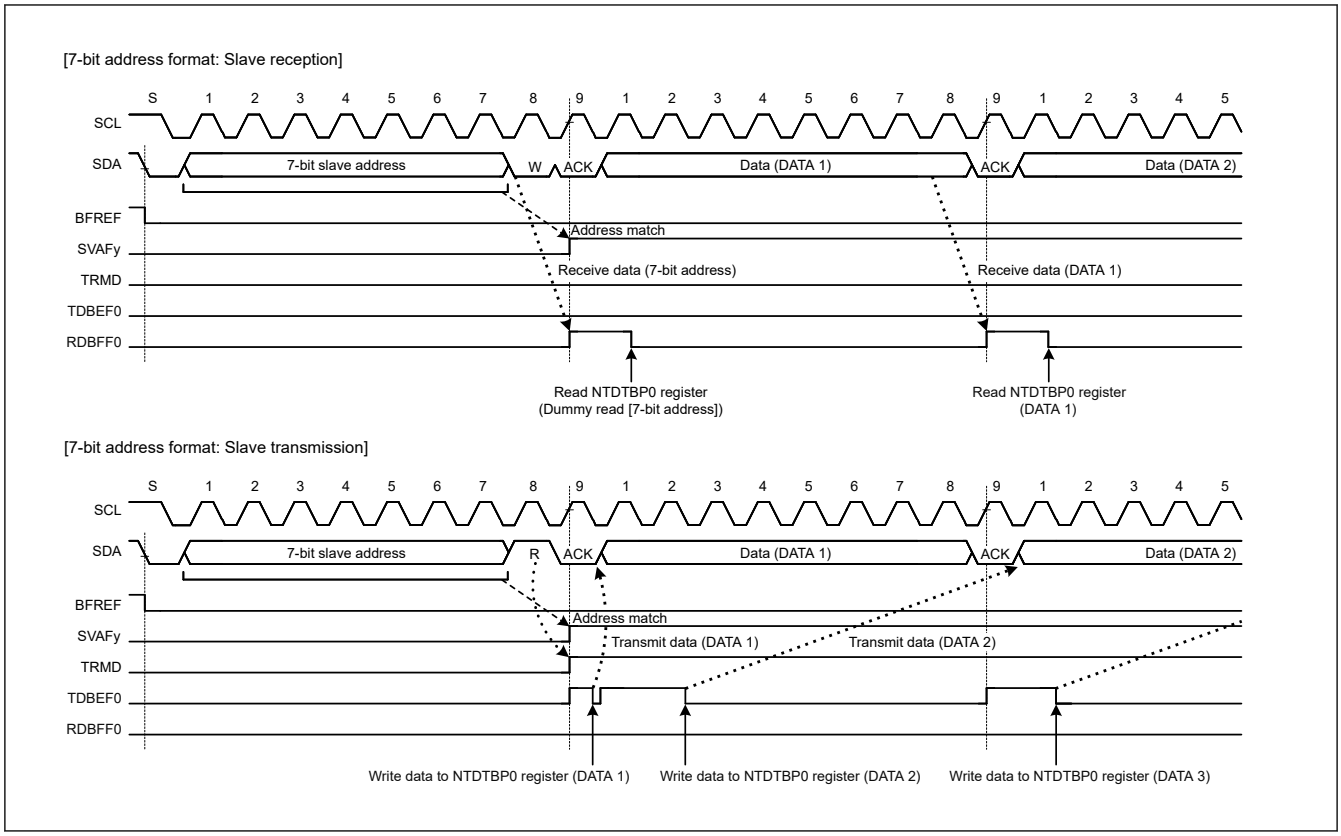


Figure 33.82 SVAFy flag set timing with 7-bit address format selected

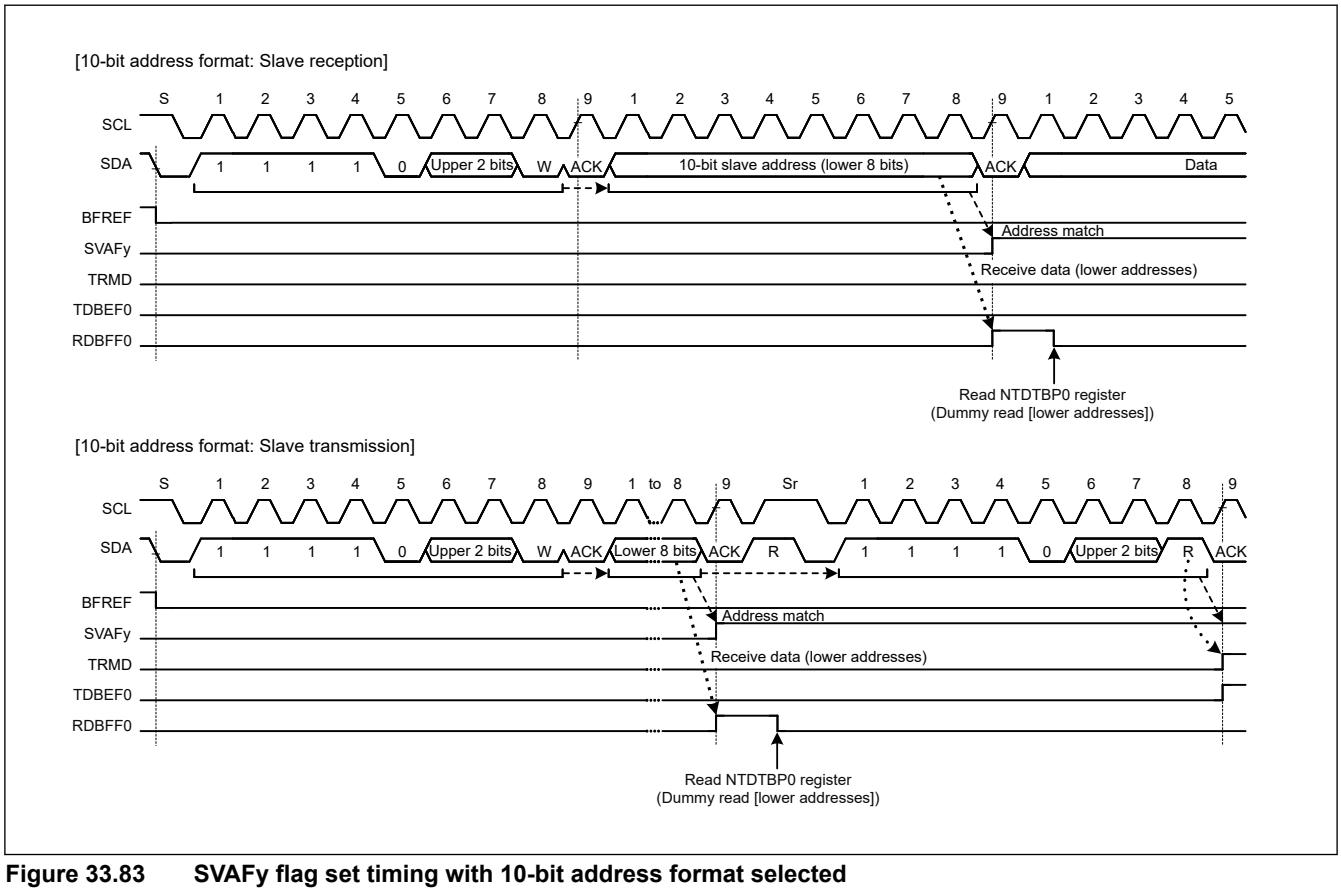


Figure 33.83 SVAFy flag set timing with 10-bit address format selected

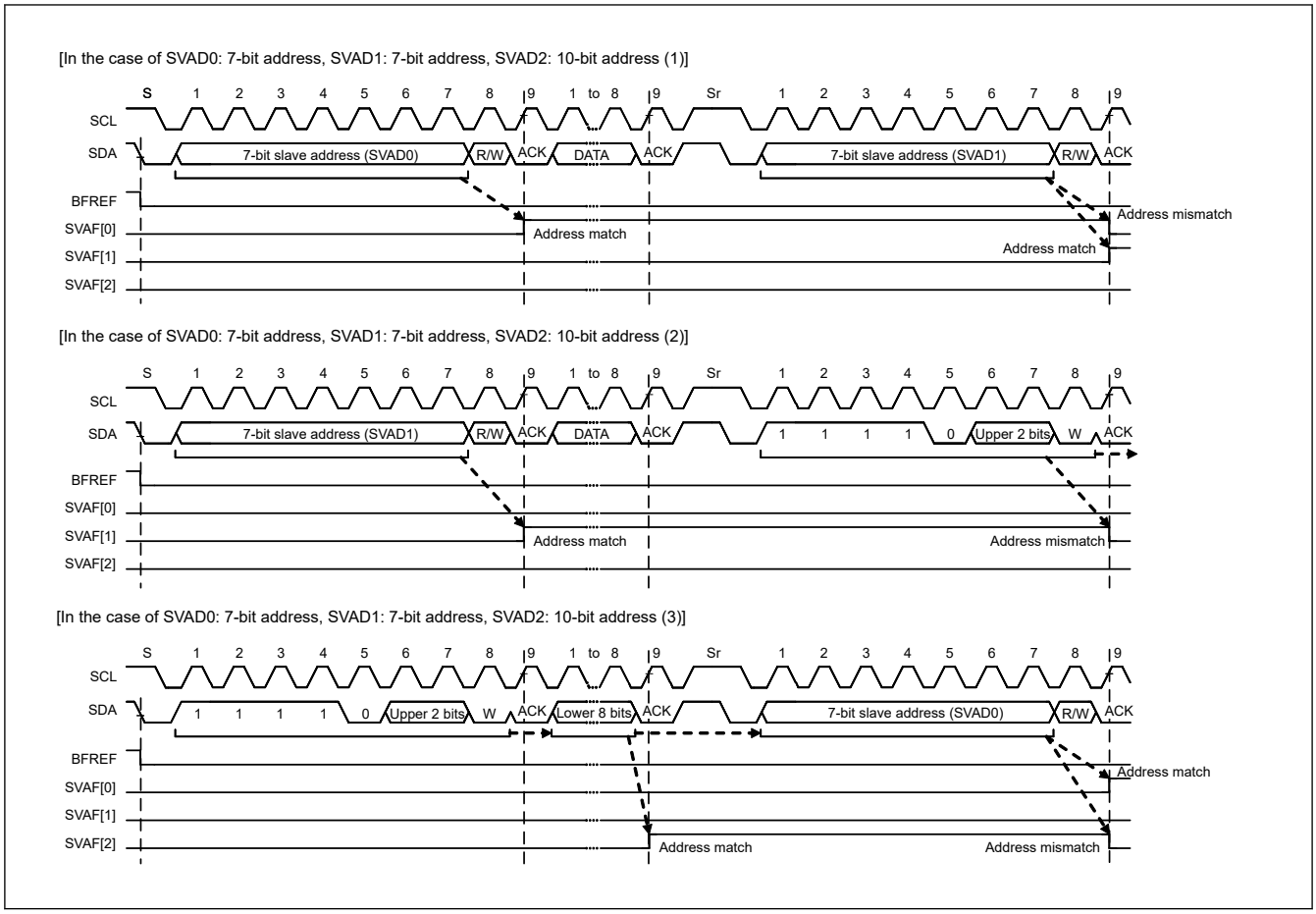


Figure 33.84 SVAFy flag set/clear timing with 7-bit/10-bit address formats mixed

(2) Detection of the General Call Address [I2C mode]

I3C has a facility for detecting the general call address (0000 000 + 0 (write)). This is enabled by setting the SVCTL.GCAE bit to 1.

If the address received after a START or Repeated START condition is issued is 0000 000 + 1 (read) (start byte), I3C recognizes this as the address of a slave device with an all-zero address but not as the general call address.

When I3C detects the general call address, both the SVST.GCAF flag and the NTST.RDBFF0 flag are set to 1 on the rising edge of the ninth cycle of SCL clock. This leads to the generation of a Normal Rx Data buffer full interrupt (I3C_RX). The value of the GCAF flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

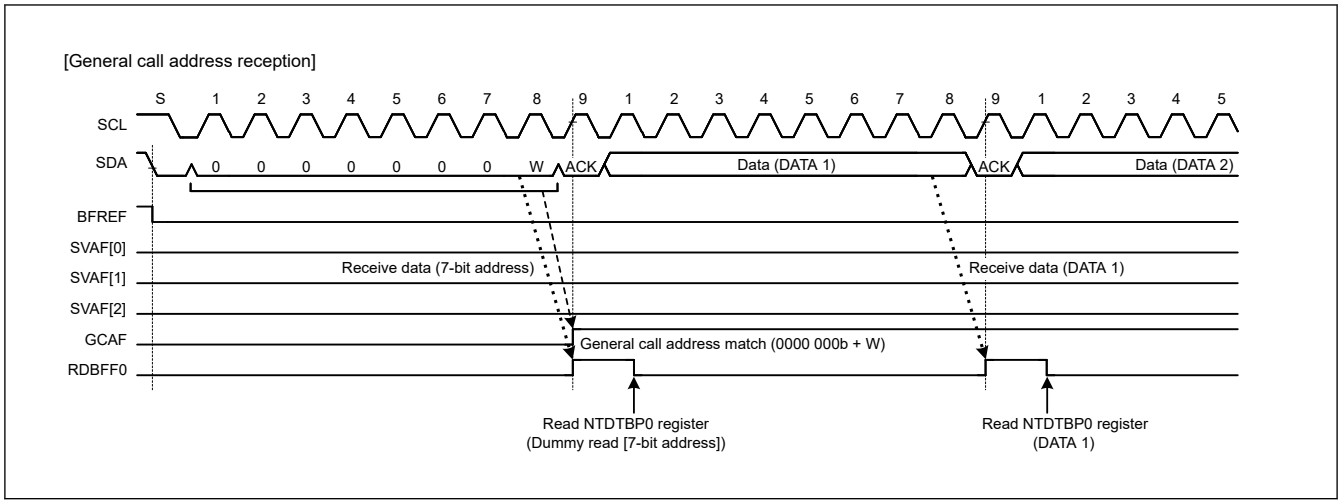


Figure 33.85 Timing of GCAF flag setting during reception of general call address

(3) Device-ID Address Detection [I2C mode]

I3C module has a facility for detecting device-ID addresses conformant with the I2C-bus specification (Rev.03). When I3C receives 1111 100 as the first byte after a START condition or Repeated START condition was issued with the SVCTL.DVIDE bit set to 1, I3C recognizes the address as a device ID, sets the SVST.DVIDF flag to 1 on the rising edge of the ninth SCL clock cycle when the following R/W# bit = 0, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, I3C sets the corresponding SVST.SVAF[y] flag (y = 0 to 2) to 1.

After that, when the first byte received after a START or Repeated START condition is issued matches the device ID address (1111 100) again and the following R/W# bit = 1, I3C does not compare the second and subsequent bytes and sets the NTST.TDBEF0 flag to 1.

In the device-ID address detection function, I3C sets the DVIDF flag to 0 if a match with I3C's own slave address is not obtained or a match with the device ID address is not obtained after a match with I3C's own slave address and the detection of a Repeated START condition. If the first byte after detection of a START or Repeated START condition matches the device ID address (1111 100) and the R/W# bit = 0, I3C sets the DVIDF flag to 1 and compares the second and subsequent bytes with I3C's slave address. If the R/W# bit = 1, the DVIDF flag holds the previous value and I3C does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DVIDF flag after confirming that TDBEF0 flag = 1.

Furthermore, prepare the device-ID fields (3 bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details of the information that must be included in device-ID fields, contact NXP Semiconductors.

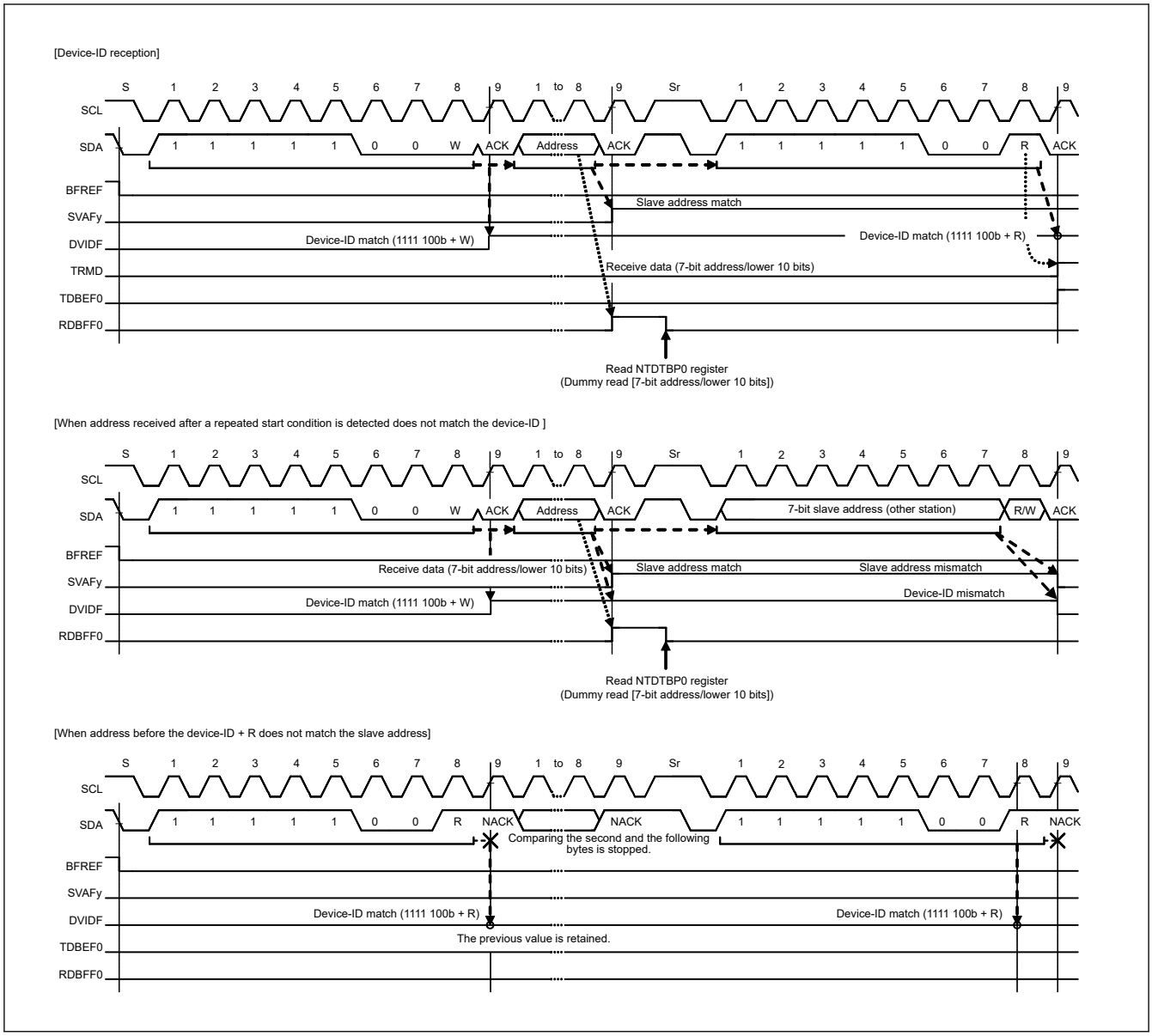


Figure 33.86 SVAFy/DVIDF flag set/clear timing during reception of device-ID

(4) Host Address Detection [I2C mode]

I3C has a function to detect the host address while the SMBus is operating. When the SVCTL.HOAE bit is set to 1 while the BFCTL.SMBS bit = 1, I3C can detect the host address (0001 000) in slave receive mode (bits CRMS and TRMD in the PRSST register = 00).

When I3C detects the host address, the SVST.HOAF flag is set to 1 at the rising edge of the ninth SCL clock cycle, and at the same time, the NTST.RDBFF0 flag is set to 1 when the R/W# bit = 0 (Wr bit). This causes a Normal Rx Data buffer full interrupt (I3C_RX) to be generated. The HOAF flag is used to recognize that the host address was sent from the smart battery or other devices.

If the bit following the host address (0001 000) is an Rd bit (R/W# bit = 1), I3C can also detect the host address. After the host address is detected, I3C operates in the same manner as normal slave operation.

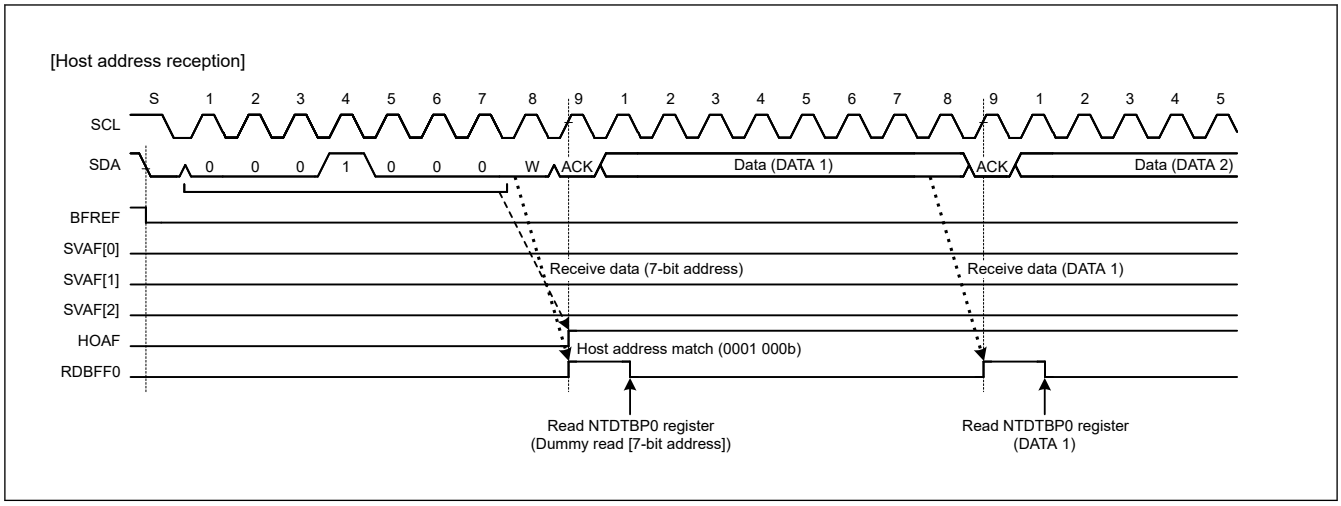


Figure 33.87 HOAF flag set timing during reception of host address

(5) Hs-mode master code Detection [I^2C mode]

IIC has a facility for detecting the Hs-mode master code (0000 1XXXb). When IIC receives the Hs-mode master code (0000 1XXXb) as the first byte after a START condition was issued with the SVCTL.HSMCE bit set to 1, this module recognizes the address as the Hs-mode master code, sets the SVST.HSMCF flag to 1 on the rising edge of the ninth SCL clock cycle. The first byte after Repeated START after NACK response to Hs-mode master code is recognized as a slave address and compared with the slave address set by SVDVADy.SVAD[9:0] (y = 0 to 2). When IIC detects a match of the set slave address, the corresponding SVST.SVAF[y] flag (y = 0 to 2) is set to 1 at the rising edge of the ninth SCL clock cycle, and the NTST.RDBFF0 flag or the NTST.TDBEF0 flag is set to 1 by the following R/W# bit. This causes a Normal Rx Data buffer full interrupt (I3C_RX) or Normal Tx Data buffer empty interrupt (I3C_TX) to be generated. The SVAF[y] flag is used to identify which slave address has been specified. The SVST.HSMCF flag is cleared to 0 when the STOP condition is detected.

Note: If the Hs-mode master code (0000 1XXXb) is received with the SVCTL.HSMCE bit set to 0, other patterns are ignored until the STOP condition is detected.

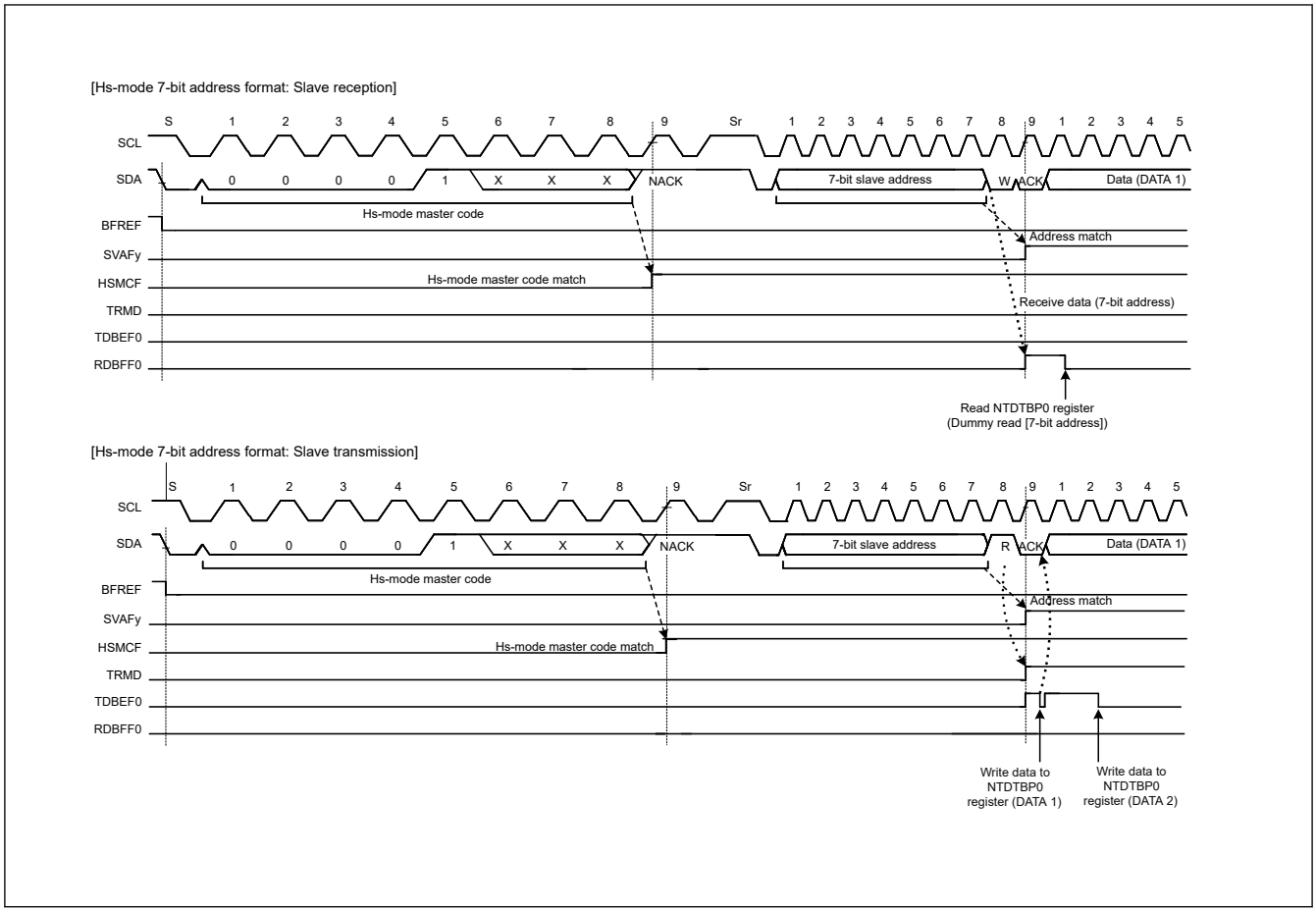


Figure 33.88 SVAFy/HSMCF Flag Set Timing during Reception of Hs-mode master code

(6) CCC detection function [I3C mode]

- In case of Broadcast CCC
  1. It receives Broadcast Address (0x7E) and R/W# = 0 after START condition or Repeated START.
  2. Respond to ACK.
  3. Receive Common Command Code (CCC).
  4. In accordance with the CCC, the following data is stored. (Storage destination: see Table 33.11)
  5. Store the Receive Status Descriptor into the Receive Status Queue.
- In case of Broadcast CCC (ENTDAA)
  1. It receives Broadcast Address (0x7E) and R/W# = 0 after START condition.
  2. Respond to ACK.
  3. Receive ENTDAA.
  4. If receives Broadcast Address (0x7E) and R/W# = 1 after Repeated START.
  5. When the Dynamic Address is not assigned, ACK response is done.
  6. This Provisional ID (SDCTPIDH[31:0], SDCTPIDL[15:0]), BCR (SVDCT.TBCRn) and DCR (SVDCT.TDCR[7:0]) are transmitted.
  7. When winning the arbitration in a transmission of the above Step 6, the dynamic address following that is received. When losing arbitration in a transmission of the above Step 6, processing of Step 6 is repeated from Step 4.
  8. When parity of the Dynamic Address is valid, ACK response is done.
  9. When parity of the Dynamic Address is invalid, NACK replies, and repeat the process from Steps 4 to 7.
  10. SDATBAS0.SDDYAD[7:0] is renewed and the SVDVAD0.SDYADV bit is set to 1.

11. Upon detecting the STOP condition, Store the Receive Status Descriptor into the Receive Status Queue.

- In case of Broadcast CCC (ENTHDR*)
  1. It receives Broadcast Address (0x7E) and R/W# = 0 after START condition or Repeated START.
  2. Respond to ACK.
  3. Receive ENTHDR*.
  4. Transit to HDR mode according to ENTHDR*.
  5. Receives the HDR command word.
  6. Compare the Dynamic Address in the HDR command word with the assigned Dynamic Address, and if they match, transmit / receive according to the Read / Write bit.
     If they do not match, wait for HDR Restart Pattern or HDR Exit Pattern + STOP.
  7. When HDR Restart Pattern or HDR Exit Pattern + STOP is received, Receive Status Descriptor is stored in Receive Status Queue.
     For HDR Restart Pattern, repeat Steps 5 to 6.
- In case of Direct Write CCC
  1. It receives Broadcast Address (0x7E) and R/W# = 0 after START condition or Repeated START.
  2. Respond to ACK.
  3. Receive Common Command Code (CCC).
  4. Receive Dynamic Address and R/W# = 0 after Repeated START.
  5. Compare the received Dynamic Address with the assigned Dynamic Address, and if it matches, I3C responds with ACK.
     If they do not match, it responds with NACK and waits for Repeated START or STOP.
  6. In accordance with the CCC, the following data is stored. (Storage destination: see Table 33.11)
  7. Store the Receive Status Descriptor into the Receive Status Queue.
- In case of Direct Read CCC
  1. It receives Broadcast Address (0x7E) and R/W# = 1 after START condition or Repeated START.
  2. Respond to ACK.
  3. Receive Common Command Code (CCC).
  4. Receive Dynamic Address and R/W# = 1 after Repeated START.
  5. Compare the received Dynamic Address with the assigned Dynamic Address, and if it matches, I3C responds with ACK.
     If they do not match, it responds with NACK and waits for Repeated START or STOP.
  6. Respond from SFR according to CCC. (Responding CCC: see Table 33.11)
  7. Store the Receive Status Descriptor into the Receive Status Queue.

Table 33.11 Common command code operation (1 of 2)

Command Code	CCC Type	Command Name	With Data	Auto Response	Storage
0x00	Broadcast	ENEC	Yes	—	SFR
0x01	Broadcast	DISEC	Yes	—	SFR
0x02	Broadcast	ENTAS0	No	—	SFR
0x03	Broadcast	ENTAS1	No	—	SFR
0x04	Broadcast	ENTAS2	No	—	SFR
0x05	Broadcast	ENTAS3	No	—	SFR
0x06	Broadcast	RSTDAA	No	—	SFR
0x07	Broadcast	ENTDAA	Yes	Yes	SFR
0x08	Broadcast	DEFSLVS	Yes	—	FIFO

Table 33.11 Common command code operation (2 of 2)

Command Code	CCC Type	Command Name	With Data	Auto Response	Storage
0x09	Broadcast	SETMWL	Yes	—	SFR
0x0A	Broadcast	SETMRL	Yes	—	SFR
0x0B	Broadcast	ENTTM	Yes	—	SFR
0x20	Broadcast	ENTHDR0	No	—	n/a
0x21	Broadcast	ENTHDR1	No	—	n/a
0x22	Broadcast	ENTHDR2	No	—	n/a
0x28	Broadcast	SETXTIME	Yes	—	FIFO
0x29	Broadcast	SETAASA	No	—	SFR
0x80	Direct Write	ENEC	Yes	—	SFR
0x81	Direct Write	DISEC	Yes	—	SFR
0x82	Direct Write	ENTAS0	No	—	SFR
0x83	Direct Write	ENTAS1	No	—	SFR
0x84	Direct Write	ENTAS2	No	—	SFR
0x85	Direct Write	ENTAS3	No	—	SFR
0x86	Direct Write	RSTDAA	No	—	SFR
0x87	Direct Write	SETDASA	Yes	—	SFR
0x88	Direct Write	SETNEWDA	Yes	—	SFR
0x89	Direct Write	SETMWL	Yes	—	SFR
0x8A	Direct Write	SETMRL	Yes	—	SFR
0x8B	Direct Read	GETMWL	—	Yes	SFR
0x8C	Direct Read	GETMRL	—	Yes	SFR
0x8D	Direct Read	GETPID	—	Yes	SFR
0x8E	Direct Read	GETBCR	—	Yes	SFR
0x8F	Direct Read	GETDCR	—	Yes	SFR
0x90	Direct Read	GETSTATUS	—	Yes	SFR
0x91	Direct Read	GETACCMST	—	Yes	SFR
0x94	Direct Read	GETMXDS	—	Yes	SFR
0x95	Direct Read	GETHDRCAP	—	Yes	SFR
0x98	Direct Write	SETXTIME	Yes	—	FIFO
0x99	Direct Read	GETXTIME	—	Yes	SFR

33.3.2.3.5 Arbitration-Lost Detection [I^2C mode]

In addition to the normal arbitration-lost detection function defined by the I^2C-bus specification, the I3C has functions to prevent double-issue of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

(1) Master Arbitration-Lost Detection (MALE Bit)

The I3C drives the I3C_SDA line low to issue a start condition. However, if the I3C_SDA line has already been driven low by another master device issuing a start condition, this module causes arbitration to be lost, so priority is given to transfer by the other master device. Similarly, if the CNDCTL.STCND bit is set to 1 while the BCST.BFREF flag is 0 (bus busy state), arbitration is lost, so priority is given to transfer by the other master device. No start condition is issued in this case.

When a start condition is issued successfully, if the data for transmission including the address bits (the internal SDA output level) and the level on the I3C_SDA line do not match (the high output as the internal SDA output, that is, the I3C_SDA pin is in the high-impedance state) and the low level is detected on the I3C_SDA line, the I3C loses in arbitration.

I3C detects master arbitration-lost when the following conditions are met while the BSTE.ALE bit = 1 and the BFCTL.MALE bit = 1 (master arbitration-lost detection enabled).

If arbitration of mastership is lost, I3C immediately enters slave receive mode.

If a slave address (including the general call address) matches its own address at this time, I3C continues in slave operation.

[Conditions for master arbitration-lost]

- Non-matching of the internal level for output on SDA and the level on the I3C_SDA line after a START condition was issued by setting the CNDCTL.STCND bit to 1 while the BCST.BFREF flag was set to 1 (erroneous issuing of a START condition)
- Setting of the CNDCTL.STCND bit to 1 (START condition double-issue error) while the BFREF flag is set to 0

  Note: I3C does not issue a START condition.
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the I3C_SDA line in master transmit mode (bits CRMS and TRMD in the PRSST register = 11b)

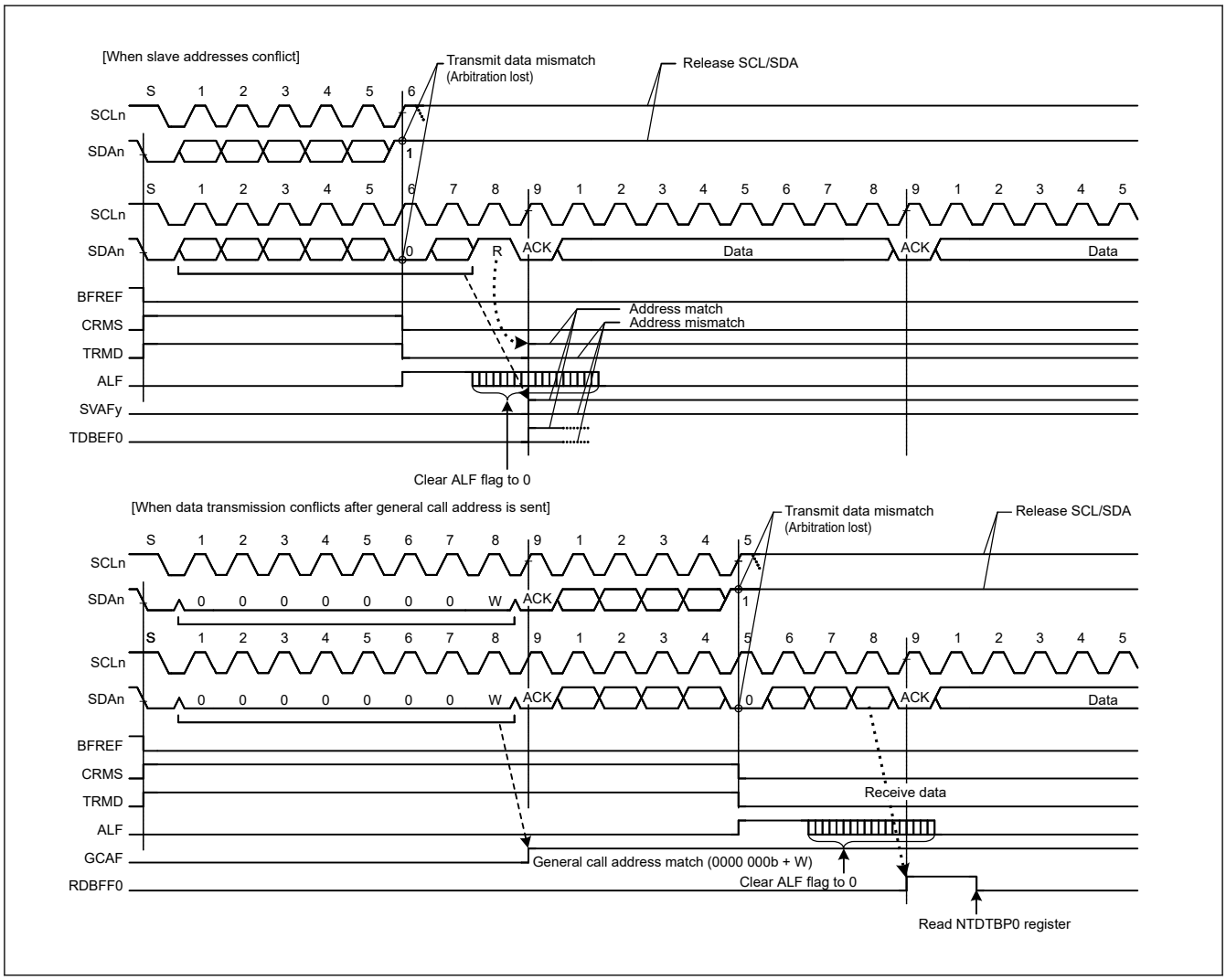


Figure 33.89 Examples of master arbitration-lost detection (MALE = 1)

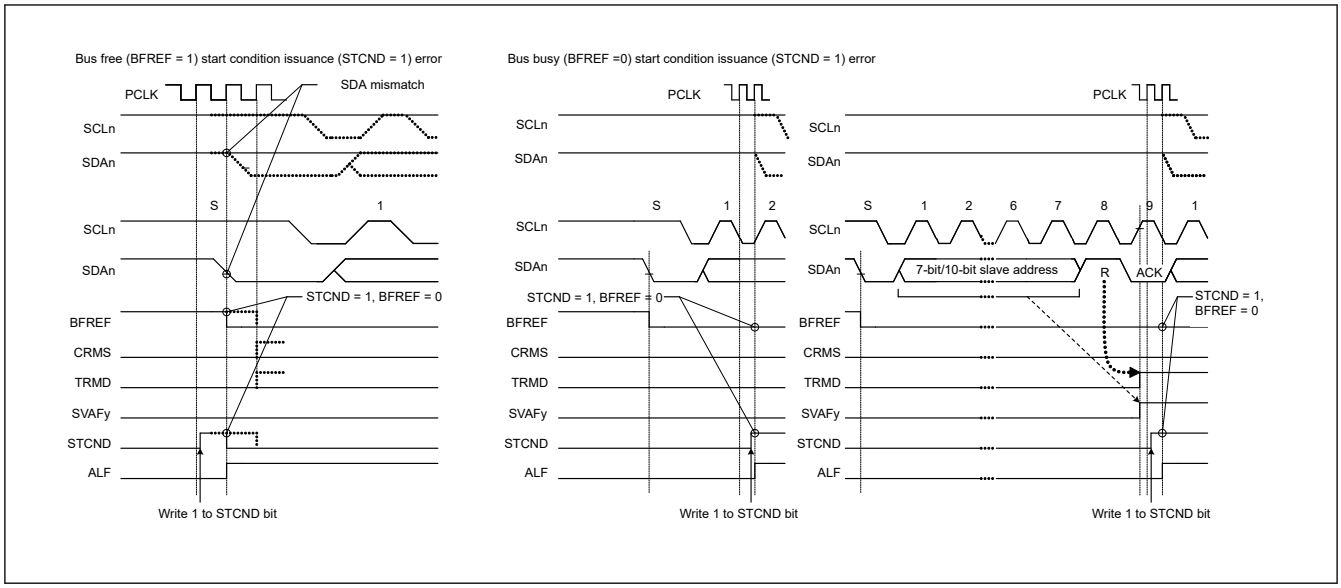


Figure 33.90 Arbitration-lost detection when a START condition is issued (MALE = 1)

(2) Arbitration-Lost Detection during NACK Transmission (NALE Bit)

The I3C has a function to cause arbitration to be lost if the internal SDA output level does not match the level on the I3C_SDA line (the high output as the internal SDA output; i.e. the I3C_SDA pin is in the high-impedance state) and the low level is detected on the I3C_SDA line during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send/receive the same information through a single slave device.

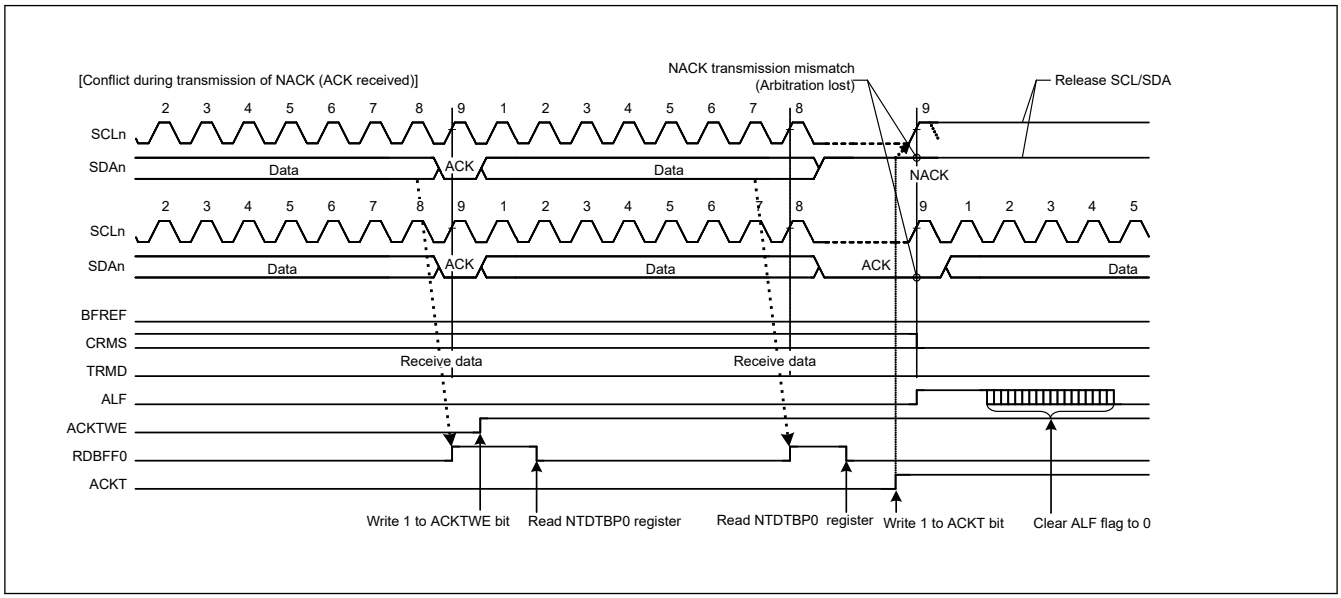


Figure 33.91 Example of arbitration-lost detection during transmission of NACK (NALE = 1)

The following section explains arbitration-lost detection using an example where two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives 2 bytes of data from the slave device, and master B receives 4 bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in both master A and master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. In this example,, master A sends NACK when it has received 2 final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received the necessary 4 bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if

a conflict like this occurs, master A cannot detect ACK transmitted by master B and issues a stop condition. Therefore, the issuance of the stop condition conflicts with the SCL clock output of master B, which disturbs communication.

When this module receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost.

If arbitration is lost during transmission of NACK, this module is immediately released from the slave-matched state and enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing (such as 0xFF transmission processing) necessary if the UDID (Unique Device Identifier) of assign address does not match in the Get UDID (general) processing after the Assign Address command.

The I3C detects arbitration-lost during transmission of NACK when the following condition is met while the BSTE.ALE bit = 1 and the BFCTL.NALE bit = 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

- When the internal SDA output level does not match the I3C_SDA line (ACK is received) during transmission of NACK (ACKCTL.ACKT bit = 1)

(3) Slave Arbitration-Lost Detection (SALE Bit)

The I3C has a function to cause arbitration to be lost if the data for transmission (the internal SDA output level) and the level on the I3C_SDA line do not match (the high output as the internal SDA output, that is, the I3C_SDA pin is in the high impedance state) and the low level is detected on the I3C_SDA line in slave transmit mode. This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.

If arbitration is lost during transmission of DATA, this module is immediately released from the slave-matched state and enters slave receive mode. This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminate subsequent redundant processing (processing for the transmission of 0xFF).

The I3C detects slave arbitration-lost when the following condition is met while the BSTE.ALE bit = 1 and the BFCTL.SALE bit = 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the I3C_SDA line in slave transmit mode (bits CRMS and TRMD in the PRSST register = 01b).

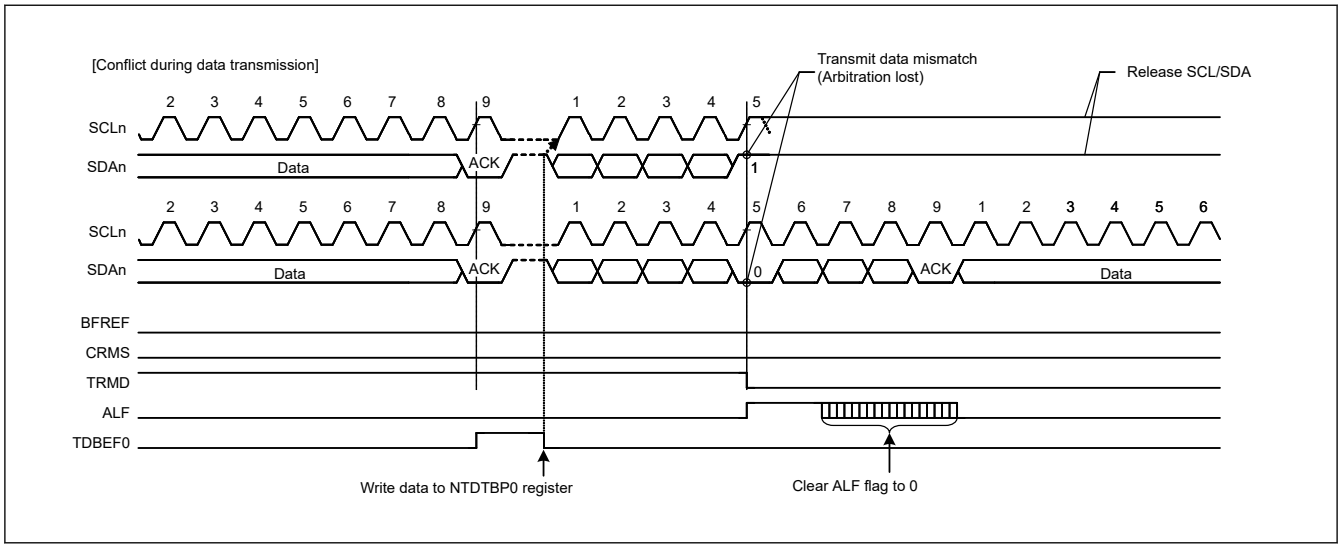


Figure 33.92 Example of slave arbitration-lost detection (SALE = 1)

33.3.2.3.6 Clock Stretching [I²C mode]

(1) Function to Prevent Wrong Transmission of Transmit Data

When data have not been written to the Normal Transfer Data Buffer Port Register 0 (NTDTBP0) with I3C in transmission mode (PRSST.TRMD = 1), the I3C_SCL line is automatically held at the low level over the intervals shown below. This low-hold period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

Master transmit mode

- Low-level interval after a START condition or Repeated START condition is issued
- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

Slave transmit mode

- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

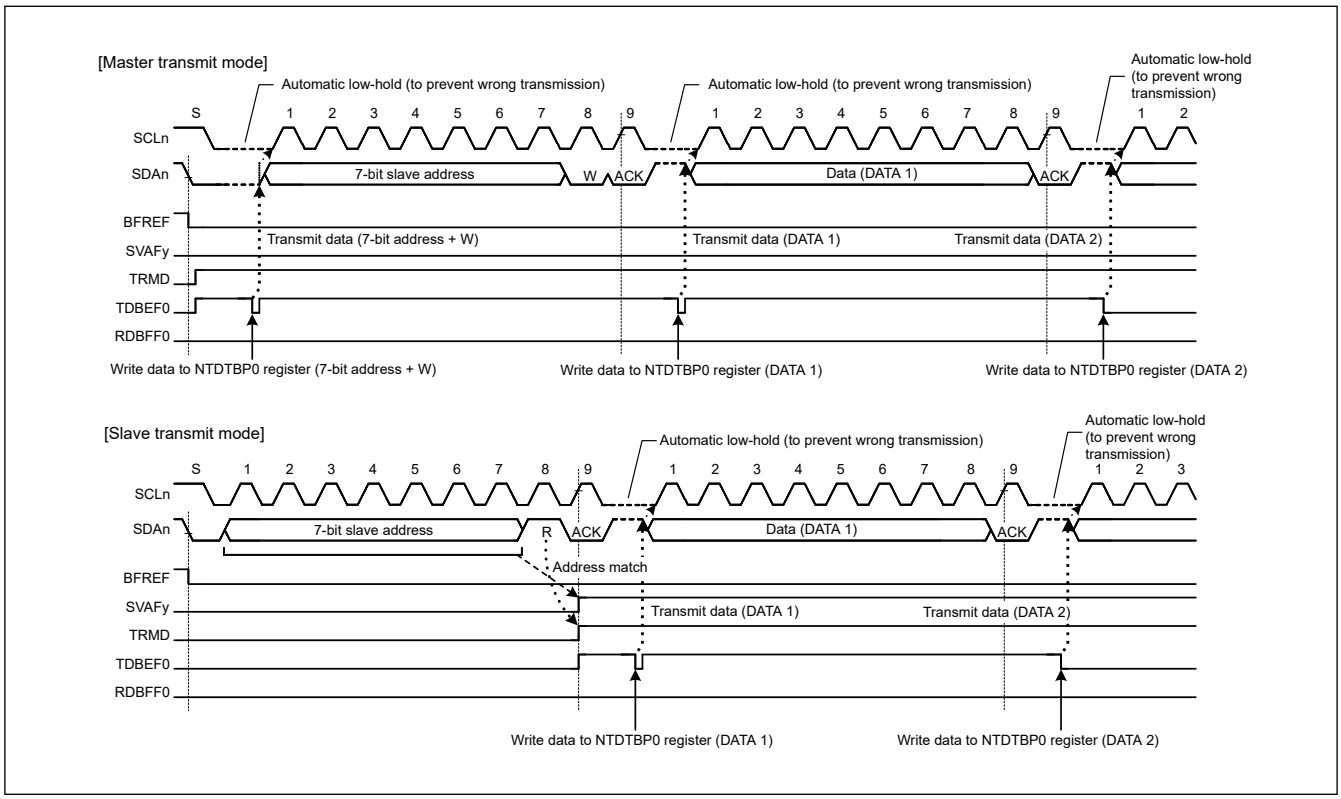


Figure 33.93 Automatic low-hold operation in transmit mode

(2) NACK Reception Transfer Abort Function

I3C has a function to abort transfer operation when NACK is received in transmit mode (PRSST.TRMD = 1). This function is enabled when the BSTE.NACKDE bit is set to 1 (transfer abort enabled). If the next transmit data has already been written (NTST.TDBEF0 = 0) when NACK is received, next data transmission at the falling edge of the ninth SCL clock cycle is automatically aborted. This prevents the I3C_SDA line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is aborted by this function (BST.NACKDF = 1), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to set the NACKDF flag to 0. In master transmit mode, restore operation using either of the methods below:

- After issuing a Repeated START condition, set the NACKDF flag to 0
- After issuing a STOP condition, set the NACKDF flag to 0 and then issue a START condition

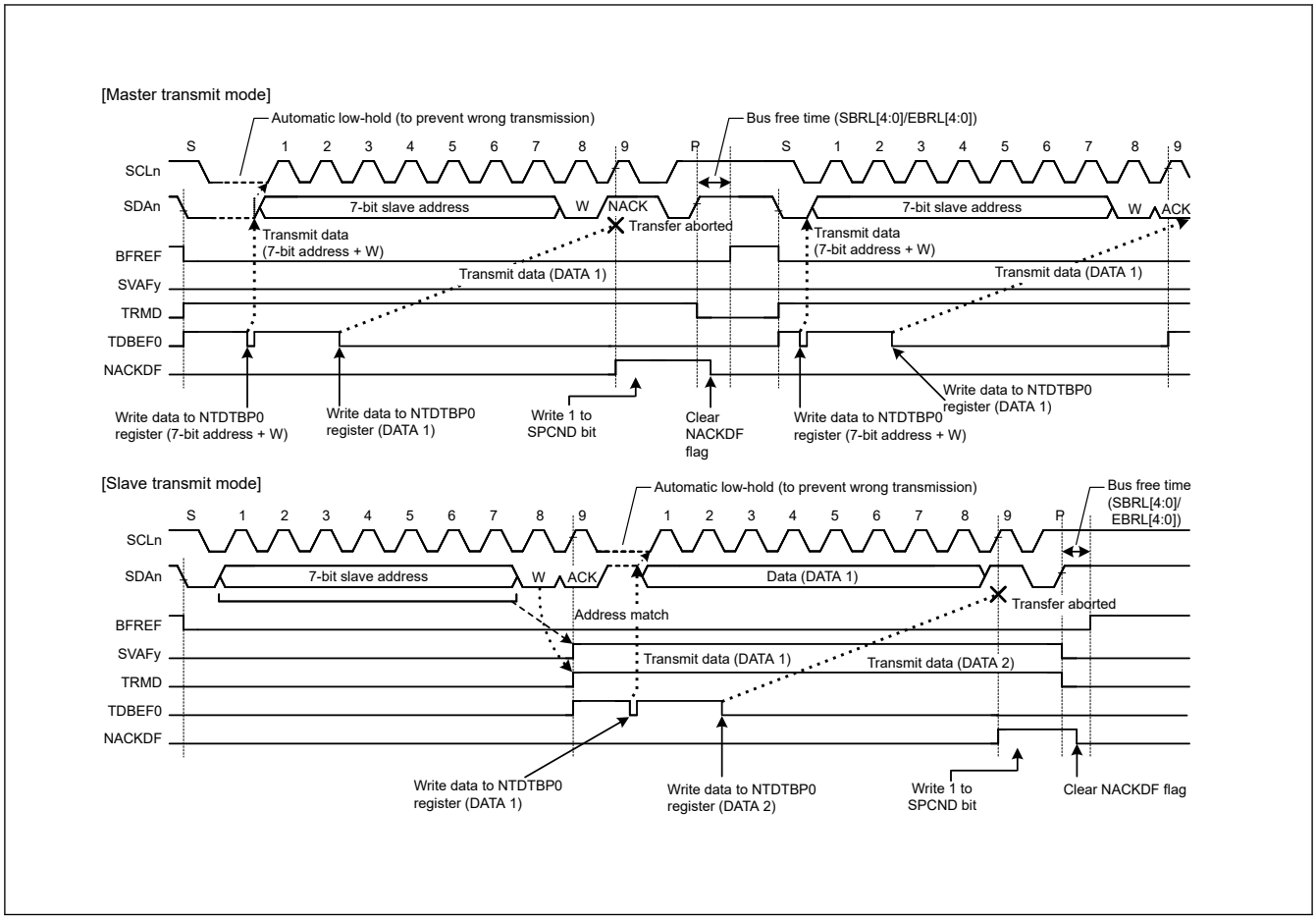


Figure 33.94 Abort of data transfer when NACK is received (NACKE = 1)

(3) Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (NTDTBP0) read is delayed for a period of one transfer frame or more with receive data full (NTST.RDBFF0 = 1) in receive mode (PRSST.TRMD = 0), I3C holds the I3C_SCL line low automatically immediately before the next data is received to prevent failure to receive data.

This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, I3C's own slave address or another slave address is received after a STOP condition is issued.

Sections in which the I3C_SCL line is held low can be selected with a combination of the RWE and ACKTWE bits in SCSTRCTL.

(a) 1-Byte Receive Operation and Automatic Low-Hold Function Using the RWE Bit

When the SCSTRCTL.RWE bit is set to 1, I3C performs 1-byte receive operation using the RWE bit function.

Furthermore, when the SCSTRCTL.ACKTWE bit = 0, I3C automatically sends the ACKCTL.ACKT bit value for the acknowledge bit in the period from the falling edge of the eighth SCL clock cycle to the falling edge of the ninth SCL clock cycle, and automatically holds the I3C_SCL line low at the falling edge of the ninth SCL clock cycle using the RWE bit function. This low-hold is released by reading data from NTDTBP0, which enables bytewise receive operation.

The RWE bit function is enabled for receive frames after a match with I3C's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

(b) 1-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the ACKTWE Bit

When the SCSTRCTL.ACKTWE bit is set to 1, I3C performs 1-byte receive operation using the ACKTWE bit function.

When the ACKTWE bit is set to 1, the NTST.RDBFF0 flag (Normal Rx Data buffer full) is set to 1 at the rising edge of the eighth SCL clock cycle, and the I3C_SCL line is automatically held low at the falling edge of the eighth SCL clock

cycle. This lowhold is released by writing a value to the ACKCTL.ACKT bit, but cannot be released by reading data from NTDTBP0, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.

The ACKTWE bit function is enabled for receive frames after a match with I3C's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

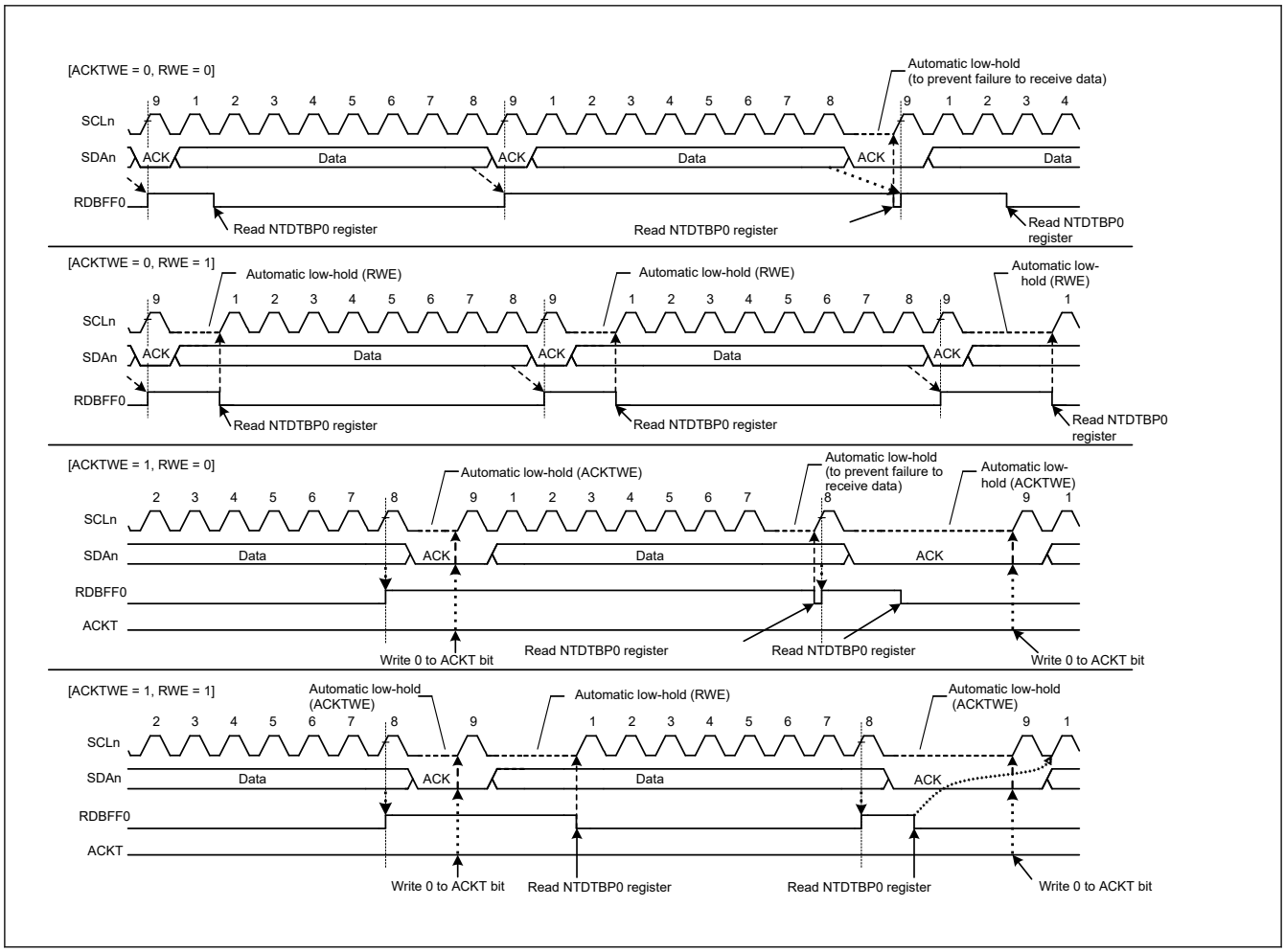


Figure 33.95 Automatic low-hold operation in receive mode (using ACKTWE and RWE bits)

33.3.2.3.7 Clock Stalling [I3C mode]

I3C has the function of stalling the SCL during the SCL Low period.

The SCL stall control is described in the table below.

Table 33.12 I3C clock stalling (1 of 2)

Clock stalling condition	Clock stalling control	Clock stalling period
I3C Transfer, ACK/NACK Phase	SCSTLCTL.ACKPE bit setting	During the count period of SCSTLCTL.STLCYC [15:0] value
	Tx Data Buffer Empty	Until data is written to the Tx Data Buffer
	Rx Data Buffer Full	Until data is read from the Rx Data Buffer
I3C Write Data Transfer, Parity Bit	SCSTLCTL.PARPE bit setting	During the count period of SCSTLCTL.STLCYC [15:0] value
	Tx Data Buffer Empty	Until data is written to the Tx Data Buffer
I3C Read Transfer, Transition Bit	Rx Data Buffer Full	Until data is read from the Rx Data Buffer

Table 33.12 I3C clock stalling (2 of 2)

Clock stalling condition	Clock stalling control	Clock stalling period
Assigned Address Phase	SCSTLCTL.AAPE bit setting	During the count period of SCSTLCTL.STLCYC [15:0] value

The following figure shows the stalling timing of each Condition.

(1) I3C Transfer, ACK/NACK Phase

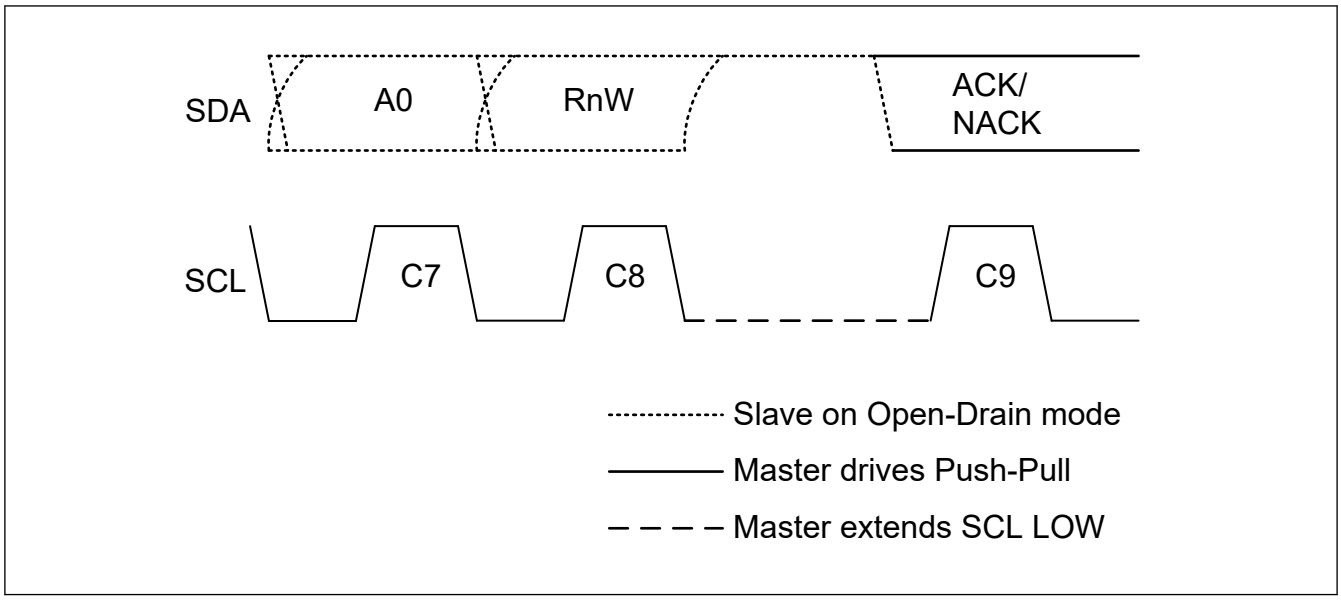


Figure 33.96 Master clock stalling in ACK phase

(2) I3C Write Data Transfer, Parity Bit

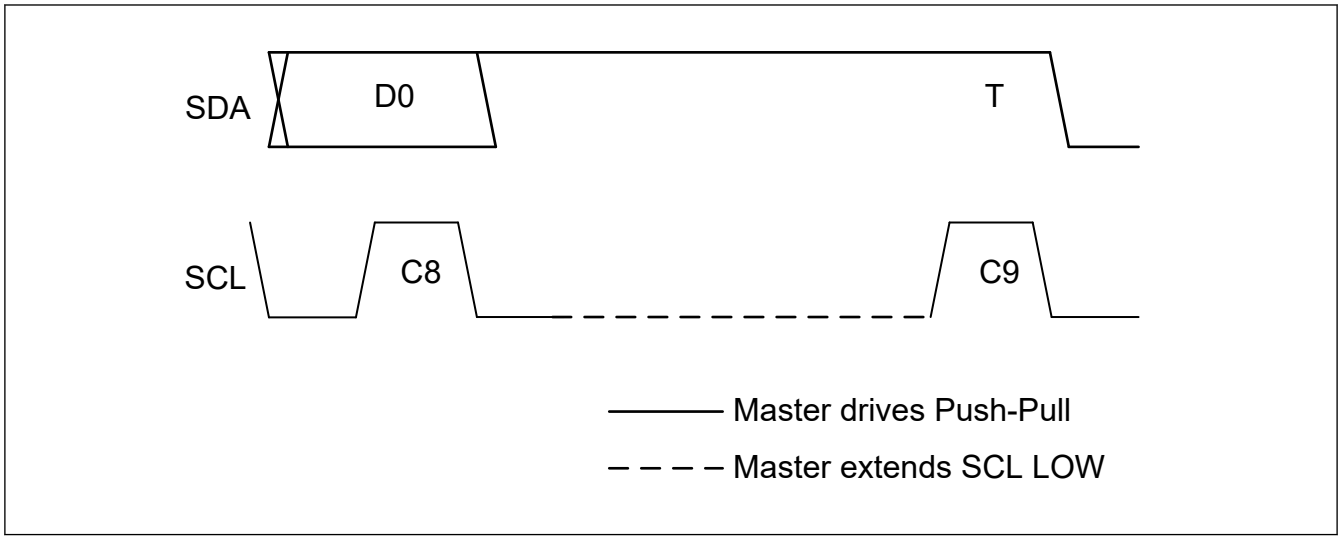


Figure 33.97 Master clock stalling in write parity bit

(3) I3C Read Transfer, Transition Bit

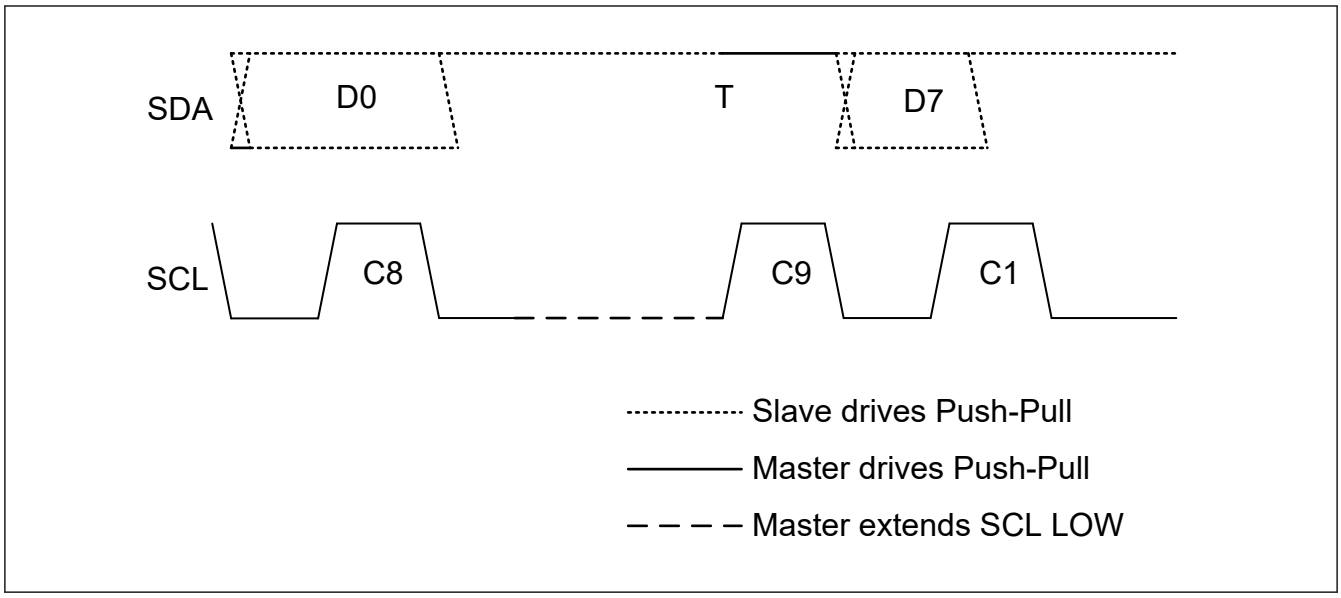


Figure 33.98 Master clock stalling in T-bit before next read data

(4) Dynamic Address Assignment, First Bit of Assigned Address

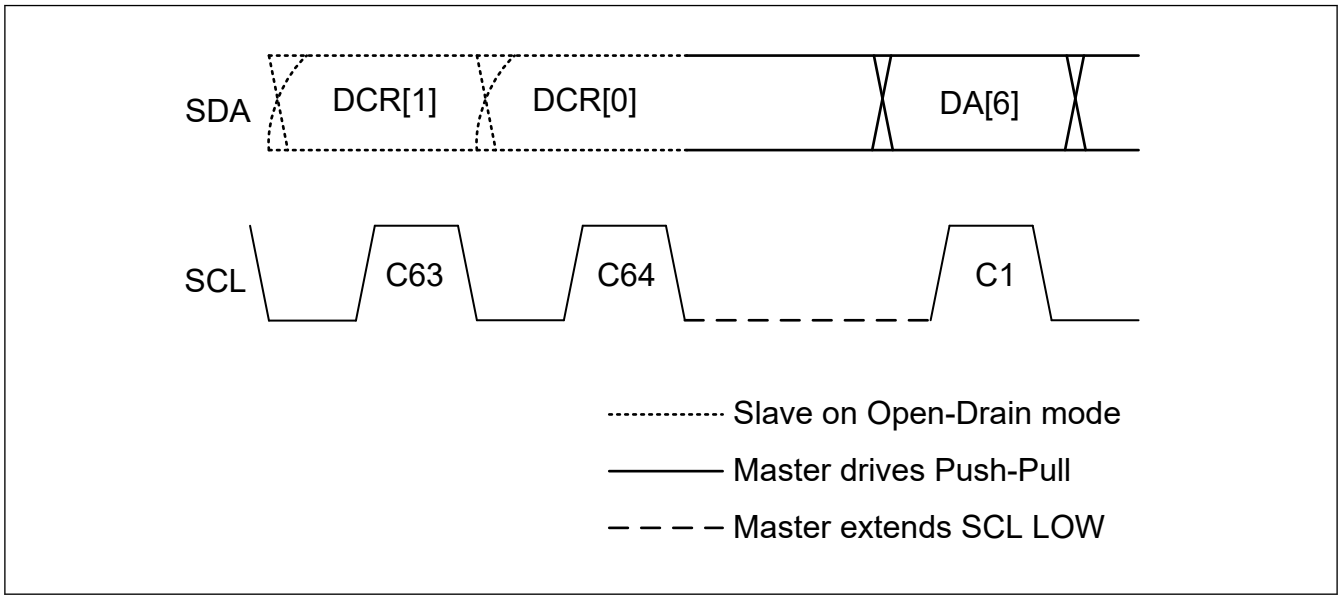


Figure 33.99 Master clock stalling in dynamic address first bit

33.3.2.3.8 In-Band Interrupt [I3C mode]

I3C detects In-Band Interrupt in the arbitrated Address Header following a START condition (but not following a Repeated START). If START Request (SDA Low Drive) is issued from Slave Device, I3C drives SCL low and completes START condition. After that, it supplies SCL and receives In-Band Interrupt Request.

The In-Band Interrupt to be detected is classified into the following three types.

- Slave Interrupt Request
- Mastership Request

The operation when detecting each In-Band Interrupt is described below.

(1) Slave Interrupt Request

1. Detect Slave Address with RnW bit High in Address Header.
2. Compare the detected Slave Address with the DVDYAD[7:0] in each DAT (DATBASm register).
3. When it does not match DAT.DVDYAD[7:0]:
   Responds NACK, then issues the STOP condition.
   When it matches the DAT.DVDYAD[7:0] bits and the DAT.DVSIRRJ bit = 1:
   It operates in the following order:
   (a) Responds NACK.
   (b) Issues Repeated START condition, then automatically issues Direct DISEC CCC to the detected Slave.
   (c) Issues the STOP condition.

   When it matches the DAT.DVDYAD[7:0] bits and the DAT.DVSIRRJ bit = 0:
   Responds ACK.
4. When DAT.DVIBIPL = 0:
   Issues the STOP condition.
   When DAT.DVIBIPL = 1:
   Drives the SCL to receive the IBI Data from the Slave following the ACK response and receives IBI Data.
   It stores the received IBI Data into the IBI Data Queue.
   Each time IBI Data of the size set by the NQTHCTL.IBIDSSZ[7:0] bits is received, the IBI Status Descriptor is stored in the IBI Queue.
5. After detection of Low of T-bit following IBI Data, issues STOP condition.
6. After issues of STOP condition
   NACK response:
   - If IBINCTL.NRSIRCTL = 0, the IBI Status Descriptor is not stored into the IBI Queue.
   - If IBINCTL.NRSIRCTL = 1, the IBI Status Descriptor is stored into the IBI Queue.

   ACK response:
   Stores the IBI Status Descriptor into the IBI Queue.

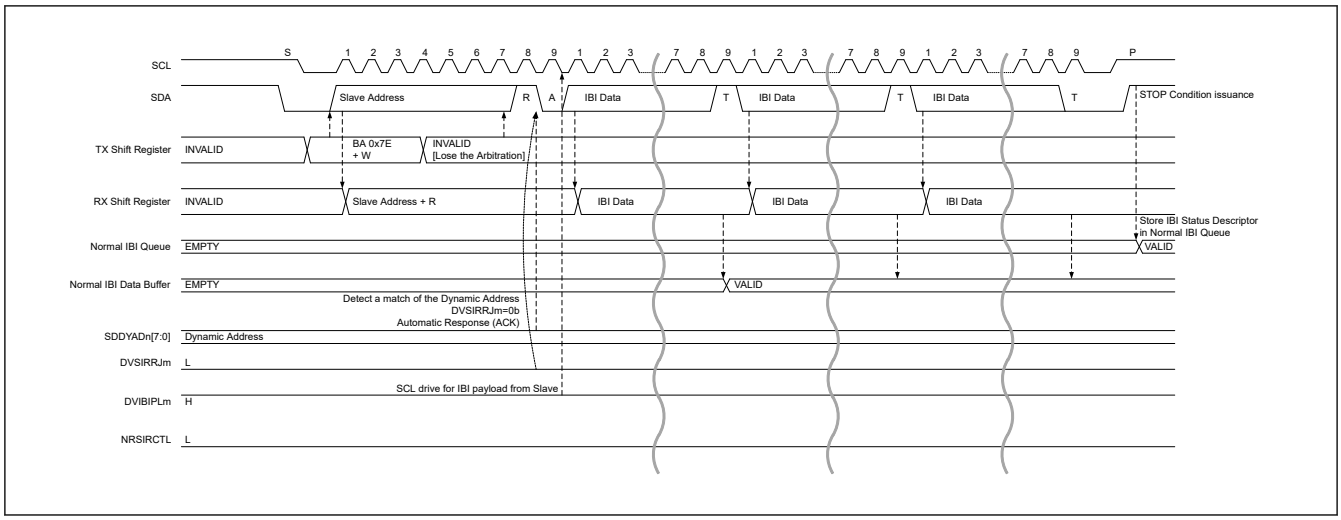


Figure 33.100 Slave interrupt request : ACK and DVIBIPL = 1

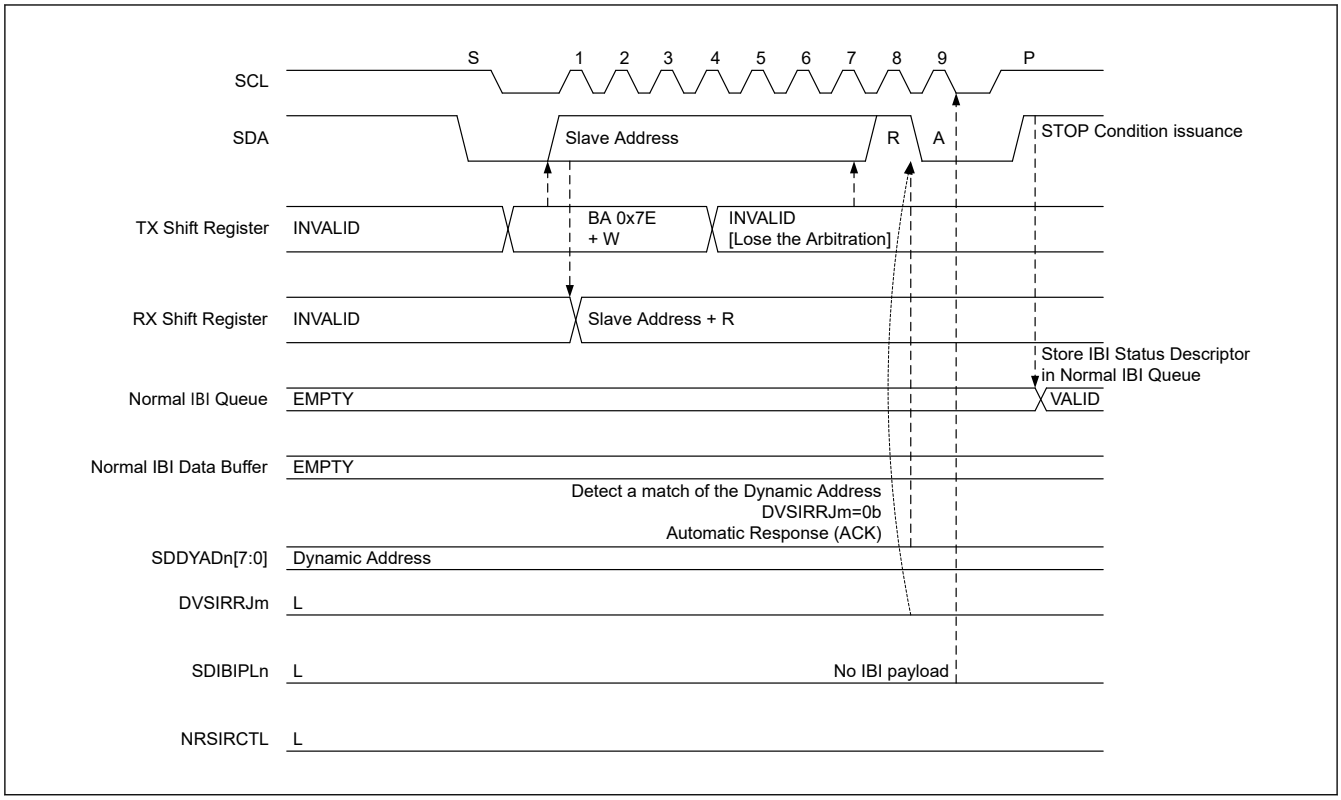


Figure 33.101 Slave interrupt request : ACK and DVIBIPL = 0

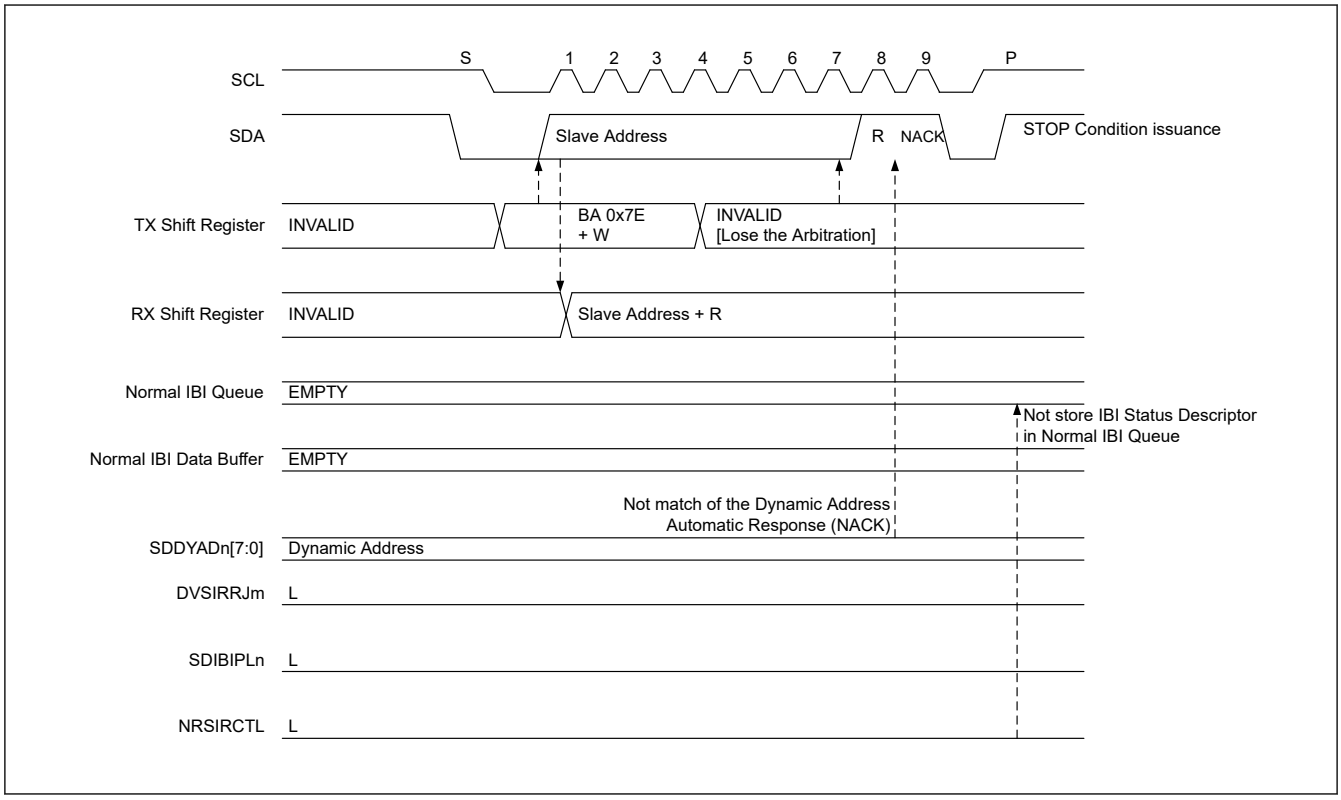


Figure 33.102 Slave interrupt request : NACK (not match the SDDYAD[7:0] of DAT) and NRSIRCTL = 0

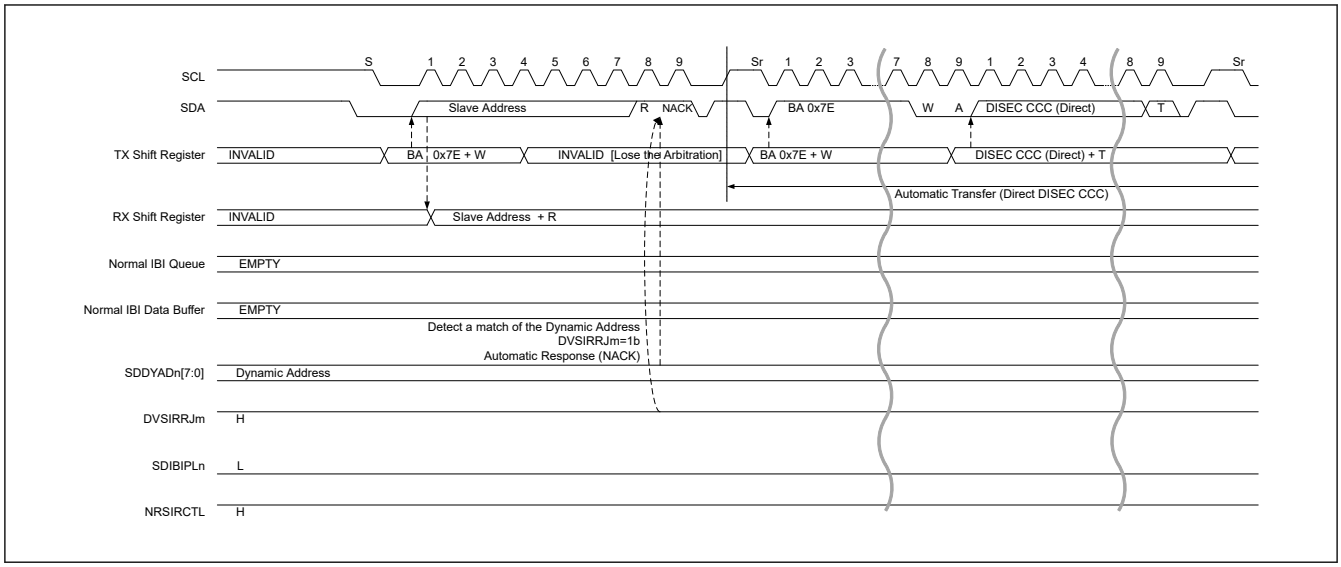


Figure 33.103 Slave interrupt request : NACK (DVSIRRJ = 1) and NRSIRCTL = 1 (1/2)

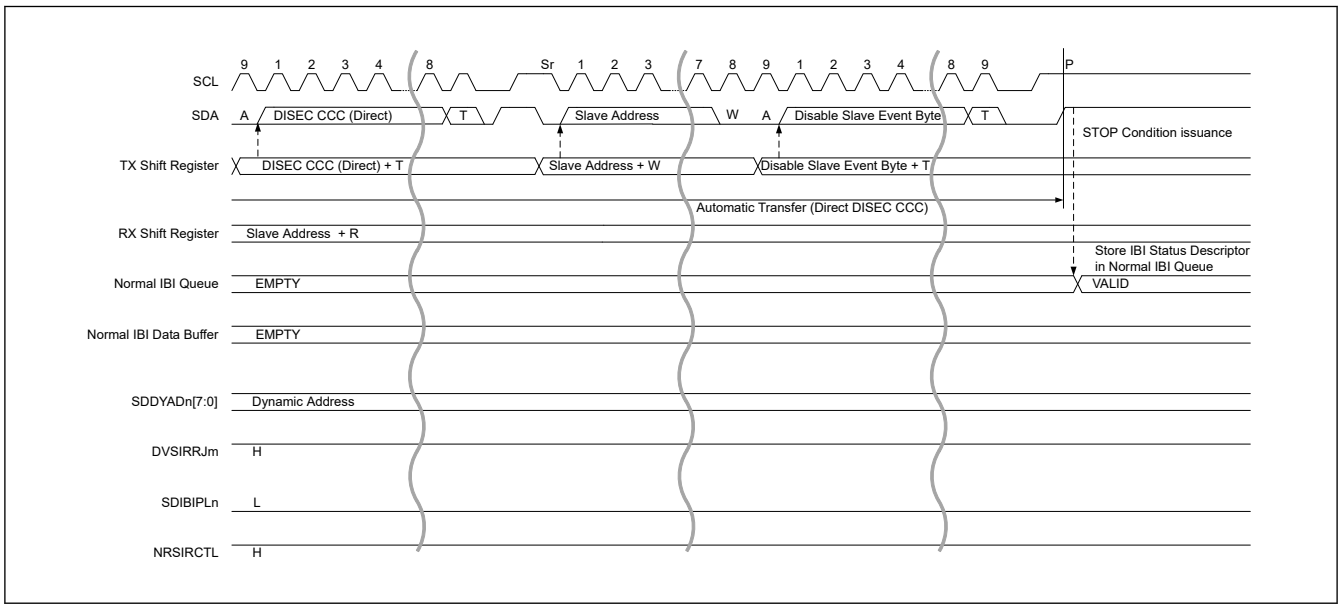


Figure 33.104 Slave interrupt request : NACK (DVSIRRJ = 1) and NRSIRCTL = 1 (2/2)

(2) Mastership Request

1. Detect Slave Address with RnW bit Low in Address Header.
2. Compare the detected Slave Address with the DVDYAD[7:0] in each DAT (DATBAS register).
3. When it does not match DAT.DVDYAD[7:0]:
   Responds NACK, then issues the STOP condition.
   When it matches the DAT.DVDYAD[7:0] bits and Device Role[1:0] in RBCR (MSDCTm) is other than I3C Master (01b):
   Responds NACK, then issues the STOP condition.
   When it matches the DAT.DVDYAD[7:0] bits and Device Role[1:0] in RBCR (MSDCTm) is I3C Master (01b):
   - When DAT.DVMRRJ = 1
     It operates in the following order.
     (a) Responds NACK.
     (b) Issued Repeated START condition and automatically issues Direct DISEC CCC to the detected Slave.
     (c) Issues the STOP condition.

- When DAT.DVMRRJ = 0
  Responds ACK, then issues STOP condition.

4. After issues of STOP condition,
   NACK response:
   - If IBINCTL.NRMRCTL = 0, the IBI Status Descriptor is not stored into the IBI Queue.
   - If IBINCTL.NRMRCTL = 1, the IBI Status Descriptor is stored into the IBI Queue.

   ACK response:
   Stores the IBI Status Descriptor into the IBI Queue.

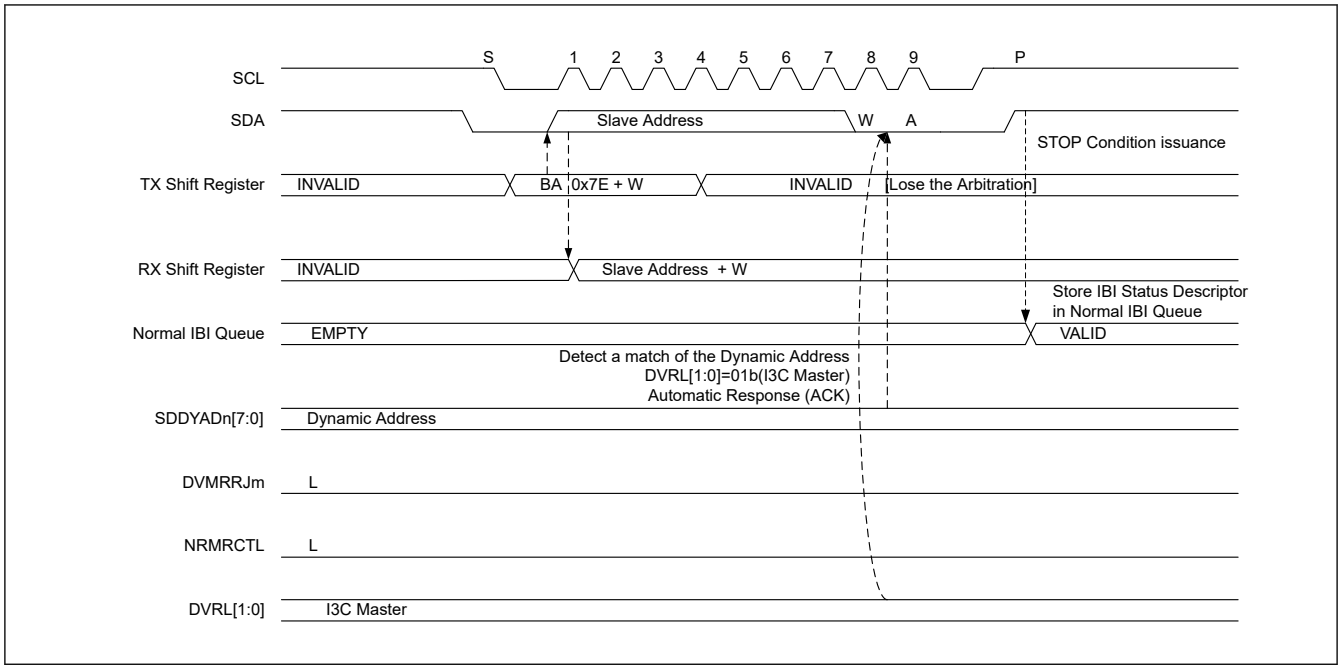


Figure 33.105 Mastership request : ACK

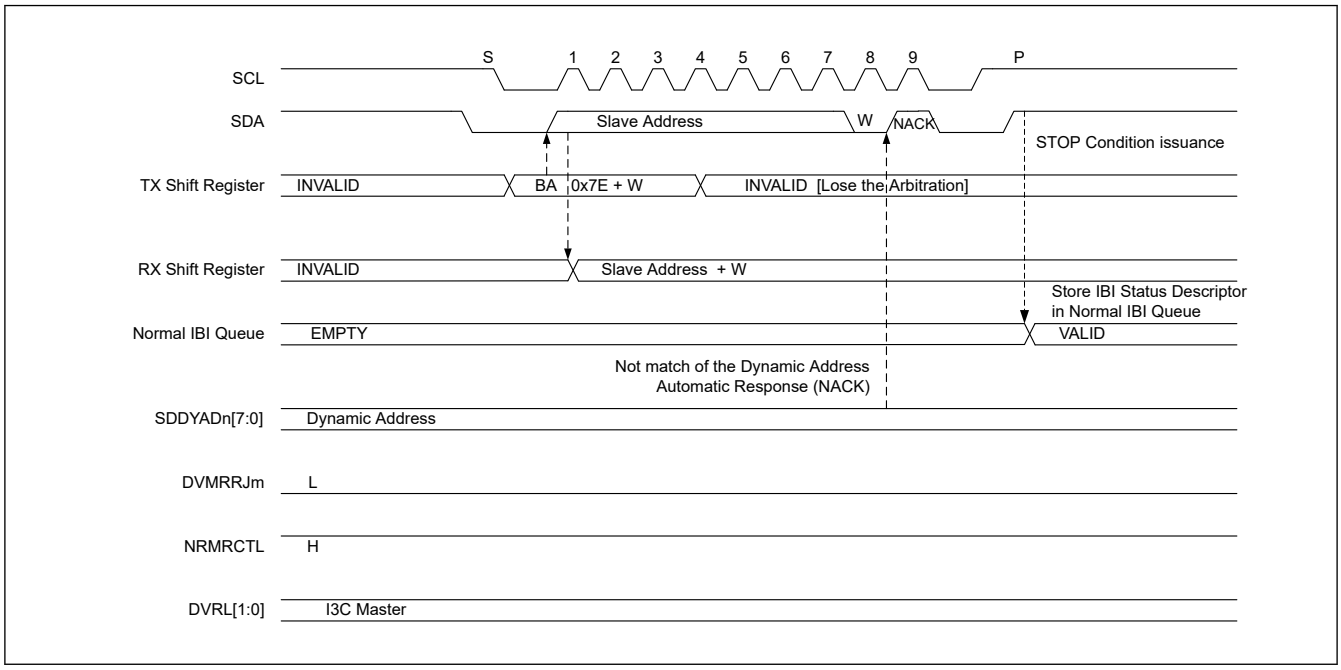


Figure 33.106 Mastership request : NACK (not match the DVDYAD[7:0] of DAT) and NRMRCTL = 1

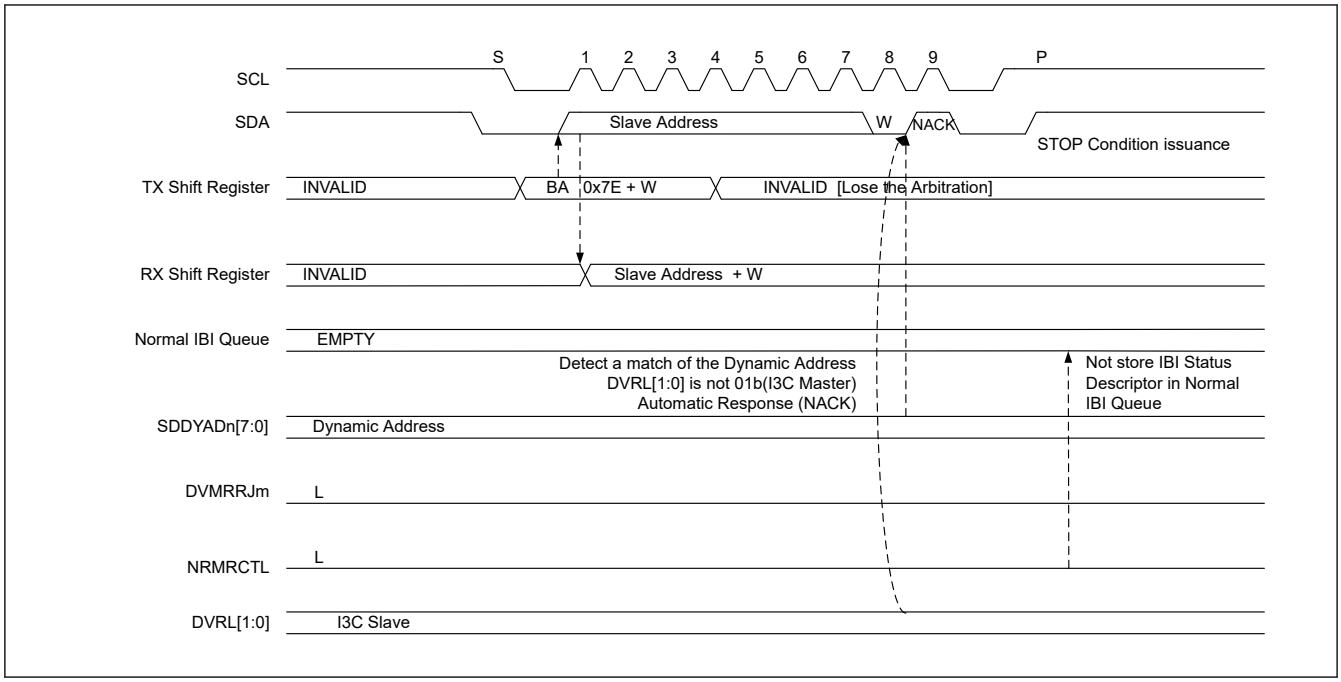


Figure 33.107 Mastership request : NACK (Device Role[1:0] is not 01 (I3C master)) and NRMRCTL = 0

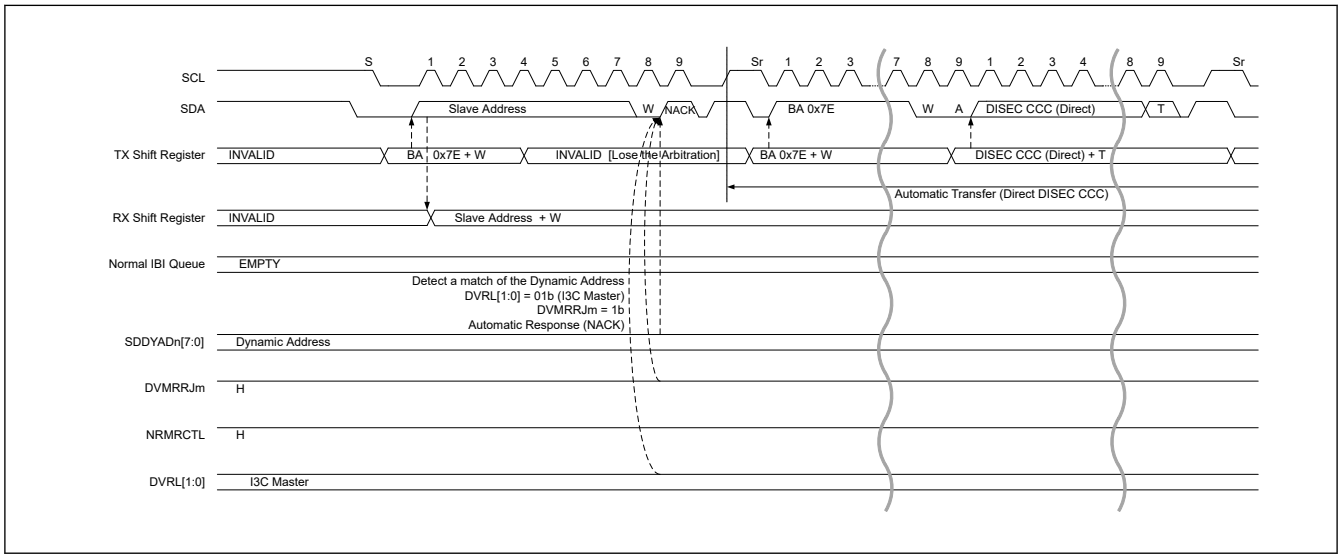


Figure 33.108 Mastership request : NACK (DVMRRJ = 1) and NRMRCTL = 1 (1/2)

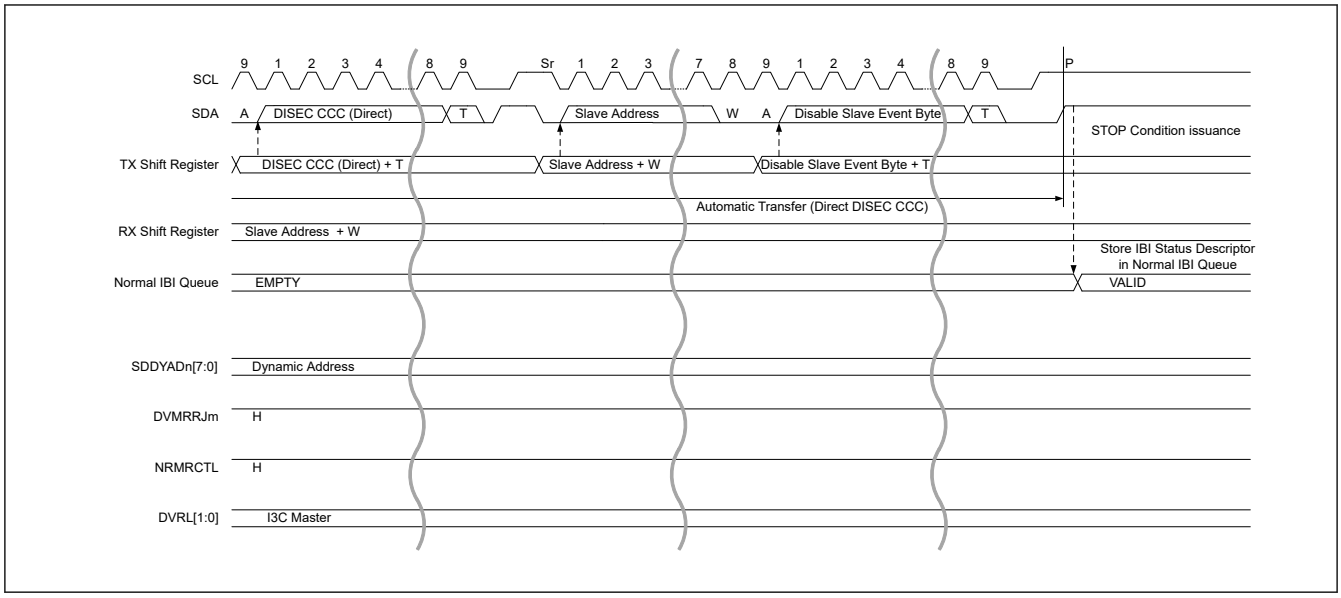


Figure 33.109 Mastership request : NACK (DVMRRJ = 1) and NRMRCTL = 1 (2/2)

33.3.2.3.9 Timing Control

Timing Control is a function that enables Master to efficiently read data from Slave by controlling and grasping the timing at which the Slave Device samples the sensor value.

For details, refer to 5.1.8 Timing Control of MIPI I3C Specification v1.0.

In I3C, timing Control supports the following three modes.

- Sync mode
- Async mode 0 (Asynchronous Basic mode)
- Async mode 1 (Asynchronous Advanced mode)

The resources for realizing Timing Control in each mode are described in the following sections.

(1) Sync Mode

1. I3C Master
   - When STCTL.STOE is set to 1, when the master sends an ST message (SETXTIME CCC with ST subcommand), there is a function to issue the synchronous timing event under the START condition of the ST message.
     While measuring the T_ph period with an external timer, the start of T_ph and the Delay Time [DT] of the ST message can be measured by capturing the count value with the synchronous timing event.
     The measured value of Delay Time is sent as a DT message (SETXTIME CCC with DT subcommand) following the ST message.
2. I3C Slave
   - When STCTL.STOE is set to 1, there is a function to issue the synchronous timing event for each START condition. STCTL.STOE is cleared when an ST message is received (SETXTIME CCC using the ST subcommand).
     Check the reception of the ST message with the Receive Status Descriptor, and use an external timer to correct the T_ph period based on the count value captured in the synchronous timing event and the Delay Time obtained from the DT message.
     While measuring the T_ph period with an external timer, the start of T_ph and the Delay Time [DT] of the ST message can be measured by capturing the count value with the synchronous timing event.
     The sampling timing is recalculated by the corrected T_ph.

(2) Async Mode 0 (Asynchronous Basic Mode)

For timing control in Async Mode 0, set the ATCTL register if necessary.

1. I3C Master

I3C has counters of MREFMREF (32-bit) and MC2 (16-bit) for Async mode 0.

- MREF Counter
  When ATCCNTE.ATCE is enabled, it starts counting.
  It captures as MREF on the SCL rise edge next to ACK for the IBI transmitted from the I3C Slave.
- MC2 Counter
  After enabling ATCCNTE.ATCE, it counts up from the SCL rise edge next to ACK for the IBI transmitted from the I3C Slave to the SCL rise edge next to the Tbit after Mandatory Byte, and capture it as MC2.

The MREF and MC2 capture values are stored next to the IBI Status Descriptor when IBI is received from the I3C Slave with the DATBASm.DVIBITS bit set to 1.

The MREF counter implemented in I3C is 32-bit counter.

However, if the 32-bit counter is insufficient due to system requirements, I3C has MREF counter overflow and MREF capture event for expansion. These events are enabled by setting ATCTL.MREFOE to 1.

MREF counter overflow is output when the internal MREF counter overflows.

The MREF counter can be extended by using it as a count event for an external timer. MREF capture event is output at the same timing as the capture timing of the internal MREF counter. By using it as the capture timing of the external timer, it can be used as an MREF counter concatenated with the value stored in the IBI Data buffer.

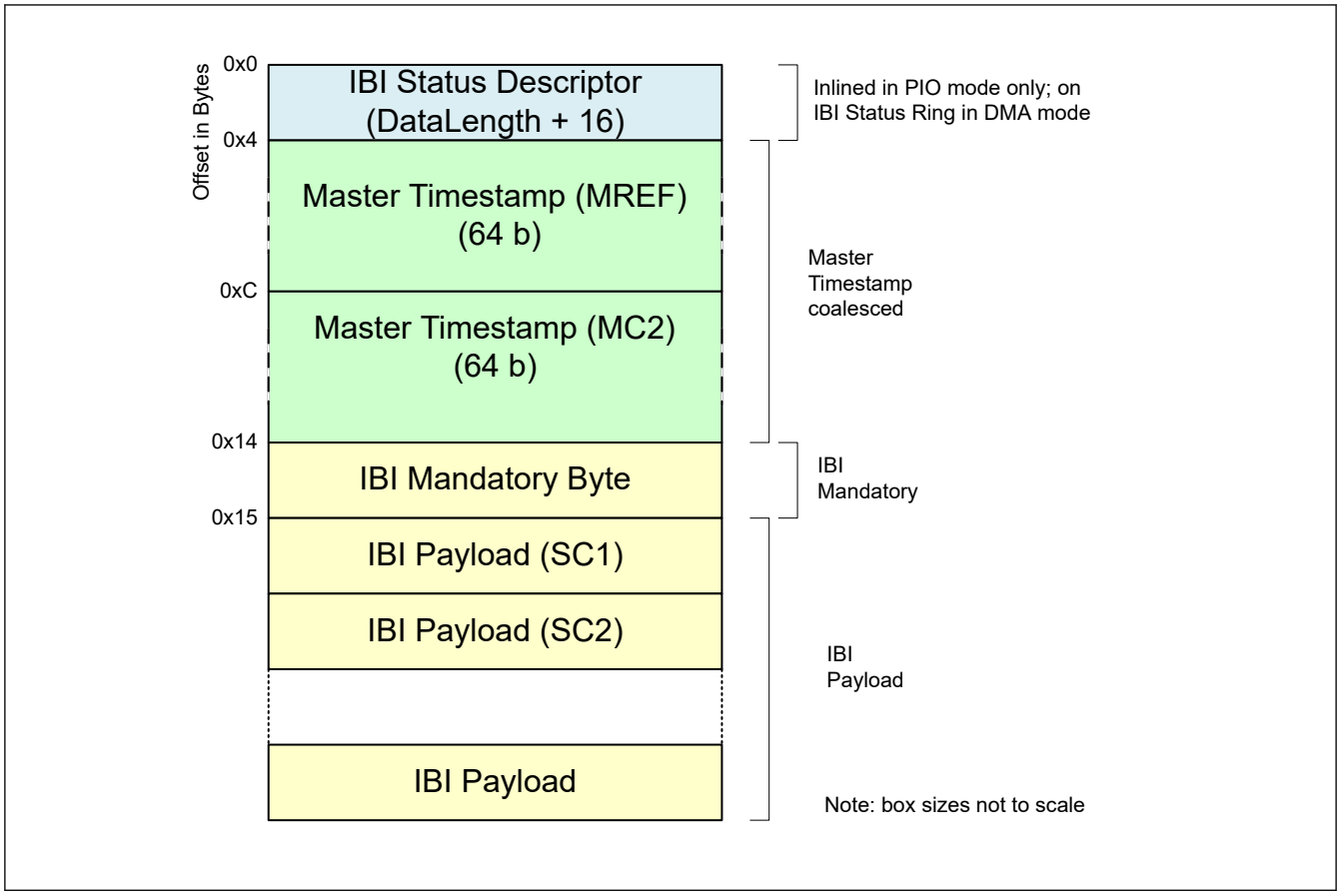


Figure 33.110 Master timestamp counters for IBI event

Note: Please evaluate the Sensor Event time of I3C Slave according to the calculation formula of the MIPI I3C specification v1.0 document.

2. I3C Slave

I3C has counters of SC1 (16-bit) and SC2 (8-bit) for Async mode 0.

- SC1 Counter

After enabling ATCCNTE.ATCE, it counts up from SC1 count trigger*1 to SCL rise edge next to ACK for the IBI, and capture it as SC1.

Note 1. SW or external trigger can be selected by selection bits.

- SC2 Counter
  After enabling ATCCNTE.ATCE, it counts up from SCL rise edge next to ACK for the IBI transmitted from I3C Slave to SCL rise edge next to Tbit after Mandatory Byte, and capture it as SC2.

When the CETSS.ASYNE [0] bit = 1 and the ITS bit in Command Descriptor for issuing IBI is 1, the SC1 and SC2 capture values are transmitted following the IBI Mandatory Byte as shown in the following figure.

If the SC1 and SC2 counters overflow, 0xFFFF and 0xFF are captured and CETSS.ICOVF is set 1.

The DATA_LENGTH[15:0] bits value of the Command Descriptor sets a value obtained by adding the number of data of SC1 and SC2 to the number of transmission data.

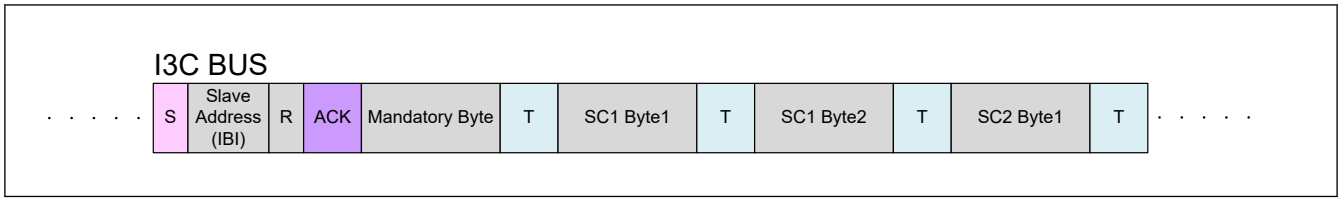


Figure 33.111 Example of asynchronous mode 0 timestamp data transfer

(3) Async Mode 1 (Asynchronous Advanced Mode)

For timing control in Async Mode 1, set the ATCTL register if necessary.

1. I3C Master
   I3C has counters of MREF (32-bit), MSyncCNT (32-bit) and MC2 (16-bit) for Async mode 1.
   - MREF Counter
     When ATCCNTE.ATCE is enabled, it starts counting.
     It captures as MREF at the SCL rise edge next to ACK for the IBI transmitted from the I3C Slave.
   - MSyncCNT Counter
     When ATCCNTE.ATCE is enabled, it starts counting.
     It captures as MSyncCNT for each aME (SDA falling edge of START condition), and store it in the capture register.
   - MC2 Counter
     After enabling ATCCNTE.ATCE, it counts up from SCL rise edge next to ACK for the IBI transmitted from I3C Slave to SCL rise edge next to Tbit after Mandatory Byte, and capture it as MC2.

   The MREF and MC2 capture values are stored next to the IBI Status Descriptor when the IBI is received from the I3C Slave with the DATBASm.DVIBITS bit set to 1 (same as Async mode 0).
   When ATCTL.AMEOE is enabled, an aME Event is issued for each aME. Use that event as a trigger to read the MSyncCNT capture value from the MRCCPT register and hold it in an external memory.
2. I3C Slave
   I3C has counters of SC1 (16-bit), SC2 (8-bit) and aME_TICK (8-bit) for Async mode 1.
   - SC1 Counter
     After enabling ATCCNTE.ATCE, it counts up from SC1 count trigger *1 to the first aME, and capture it as SC1.

     Note 1. SW or external trigger can be selected by selection bits.
   - SC2 Counter
     After enabling ATCCNTE.ATCE, it counts up from SCL rise edge next to ACK for the IBI transmitted from I3C Slave to SCL rise edge next to Tbit after Mandatory Byte, and capture it as SC2.
   - aME_TICK Counter
     After enabling ATCCNTE.ATCE, it counts every aME, and capture it as aME_TICK at the SCL rise edge next to ACK for the IBI.
     The aME_TICK counter is cleared on the first aME after the SC1 count trigger.

When the CETSS.ASYNE[1] bit = 1 and the ITS bit in Command Descriptor for issuing IBI is 1, the SC1, SC2 and aME_TICK capture values are transmitted following the IBI Mandatory Byte as shown in the following figure.

If the SC1 and SC2 counters overflow, 0xFFFF and 0xFF are captured and CETSS.ICOVF is set 1.

The DATA_LENGTH[15:0] bits value of the Command Descriptor sets a value obtained by adding the number of data of SC1, SC2 and aME_TICK to the number of transmission data.

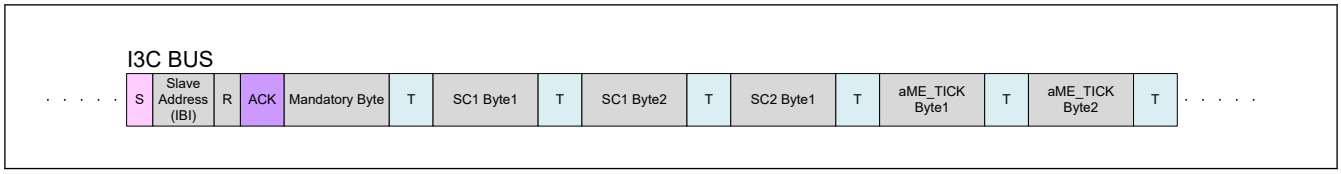


Figure 33.112 Example of asynchronous mode 1 timestamp data transfer

33.3.2.3.10 HDR-DDR CRC5 Algorithm

The CRC5 value is computed on a complete message, including the Command Word and all Data Words.

For a Command Word, the CRC5 is computed based on the value of the 16-bit payload, including:

- The Read vs. Write bit
- The Command Value
- The Slave Address, and
- The lowest-order bit (Write Reserved and Read Parity adjustment).

For Data Words, the CRC5 is computed based on all Data 16-bit values transmitted for that Command.

The CRC5 value is initialized to 0x1F. The CRC5 polynomial is below.

$CRC5 = X^5 + X^2 + X^0$

33.3.2.3.11 Port Control

(1) Extra SCL Clock Cycle Output Function

In master mode, I3C module has a facility for the output of extra SCL clock cycles to release the I3C_SDA line of the slave device from being held at the low level due to the master being out of synchronization with the slave device.

This function is mainly used in master mode to release the I3C_SDA line of the slave device from the state of being fixed to the low level by including extra cycles of SCL output from I3C with single cycles of the SCL clock as the unit in the case of a bus error where I3C cannot issue a Repeated START condition or a STOP condition because the slave device is holding the I3C_SDA line at the low level. Do not use this facility in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.

When the OUTCTL.EXCYC bit is set to 1, an additional clock pulse at the frequency set by the REFCKCTL.IREFCKS[2:0] bits and the STDBR.SBRHO[7:0] and STDBR.SBRLO[7:0] registers is output from the I3C_SCL pin. After output of this clock pulse, the EXCYC bit automatically becomes 0. After confirming that the EXCYC bit is 0, wait for the setup time of the Repeated START condition or STOP condition, and then confirm the detection of the Repeated START condition or STOP condition. If the Repeated START condition or STOP condition is not detected, consecutive additional clock pulses can be output by writing 1 to the EXCYC bit again.

When I3C module is in master mode and the slave device is holding the I3C_SDA line at the low level because synchronization with the slave device has been lost due to the effects of noise, etc., the output of a Repeated START condition or a STOP condition is not possible. The facility for output of an extra cycle of the SCL clock can be used to output extra cycles of SCL one by one to make the slave device release the I3C_SDA line from being held at the low level, thus recovering the bus from an unusable state. Release of the I3C_SDA line by the slave device can be monitored by reading the SDILV bit in PRSTDBG. After the I3C_SDA line has been released by the slave device, the preset of a Repeated START condition or a STOP condition is issued.

Use this function with the BFCTL.MALE bit set to 0 (master arbitration-lost detection is disabled).

[Output conditions for using the EXCYC bit in OUTCTL]

- When the bus is free (BFREF flag in BCST = 1) or in master mode (CRMS bit = 1 in PRSST and BFREF flag = 0 in BCST)
- When the communication device does not hold the I3C_SCL line low

Figure 33.113 shows the operation timing of the extra SCL clock cycle output function (EXCYC bit).

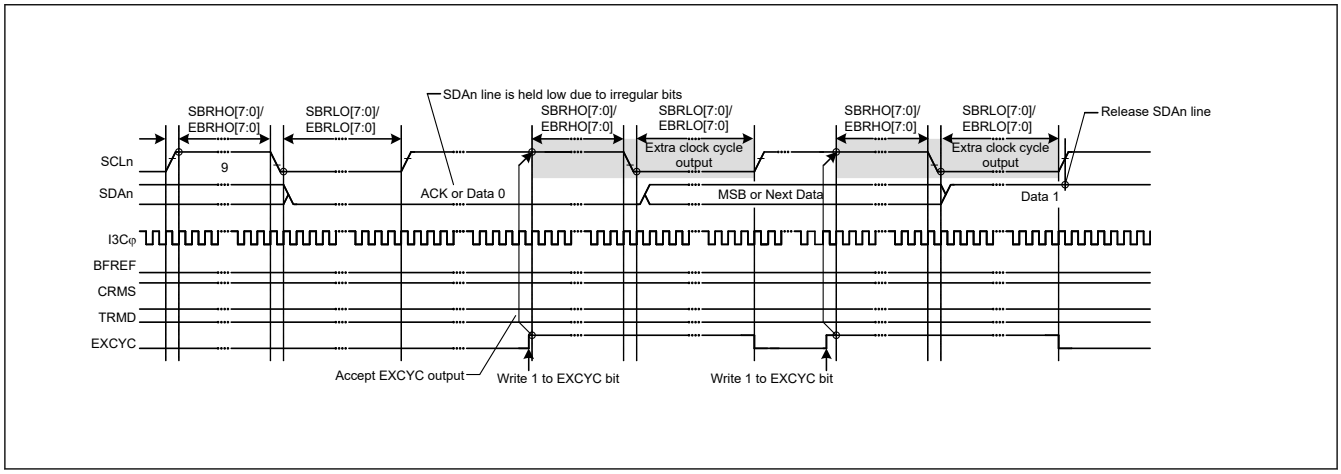


Figure 33.113 Extra SCL clock cycle output function (EXCYC bit)

33.3.2.3.12 SMBus Operation [I²C mode]

I3C is available for data communication conforming to the SMBus (Version 2.0). To perform SMBus communication, set the BFCTL.SMBS bit to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus specification, set the REFCKCTL.IREFCKS[2:0] bits, the STDBR.SBRHO[7:0] bits, and the STDBR.SBRLO[7:0] bits. In addition, determine the values of the OUTCTL.SDODCS bit and the OUTCTL.SDOD[2:0] bits to meet the data hold time specification of 300 ns or more. If I3C is used only as an I²C slave device, the transfer rate setting is not necessary, whereas the STDBR.SBRLO[7:0] bits needs to be set to a value longer than the data setup time (250 ns).

For the SMBus device default address (1100 001), use one of the slave device address table basic registers 0 to 2 (SDATBASn .SDSTAD[6:0] bits (y = 0 to 2), and set the corresponding SDATBASn .SDADLS bit (7-bit/10-bit address format select) (y = 0 to 2) to 0 (7-bit address format).

When transmitting the UDID (Unique Device Identifier), set the BFCTL.SALE bit to 1 to enable the slave arbitration lost detection function.

(1) SMBus Timeout Measurement

(a) Measuring timeout of slave device

The following period (timeout interval: $T_{LOW:SEXT}$) must be measured for slave devices in SMBus communication.

- From START condition to STOP condition

To measure timeout for slave devices, measure the period from START condition detection to STOP condition detection with the GPT timer using a START condition detection interrupt (I3C_EEI) and STOP condition detection interrupt (I3C_EEI) of I3C. The measured timeout period must be within the total clock low-level period [slave device] $T_{LOW:SEXT}$: 25 ms (max.) of the SMBus specification.

If the time measured with the GPT exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (min.) of the SMBus specification, the slave device must release the bus by writing 1 to the RSTCTL.INTLRST bit to issue an internal reset of I3C. When an internal reset is issued, I3C stops driving the bus for the I3C_SCL pin and I3C_SDA pin and make the I3C_SCL/I3C_SDA pin outputs high-impedance, which releases the bus.

(b) Measuring timeout of master device

The following periods (timeout interval: $T_{LOW:MEXT}$) must be measured for master devices in SMBus communication.

- From START condition to acknowledge bit
- Between acknowledge bits

- From acknowledge bit to STOP condition

To measure timeout for master devices, measure these periods with the GPT timer using a START condition detection interrupt (I3C_EEI), STOP condition detection interrupt (I3C_EEI), and transmit end interrupt (I3C_TEND) or Normal Rx Data buffer full interrupt (I3C_RX) of I3C. The measured timeout period must be within the total clock lowlevel extended period (master device) $T_{LOW:MEXT}$: 10 ms (max.) of the SMBus specification, and the total of all $T_{LOW:MEXT}$ from START condition to STOP condition must be within $T_{LOW:SEXT}$: 25 ms (max.).

For the ACK receive timing (rising edge of the ninth SCL clock cycle), monitor the BST.TENDF flag in master transmit mode (master transmitter) and the NTST.RDBFF0 flag in master receive mode (master receiver). For this reason, perform bytewise transmit operation in master transmit mode, and hold the SCSTRCTL.ACKTWE bit 0 until the byte just before reception of the final byte in master receive mode. While the ACKTWE bit = 0, the RDBFF0 flag is set to 1 at the rising edge of the ninth SCL clock cycle.

If the period measured with the GPT exceeds the total clock low-level extended period (master device) $T_{LOW:MEXT}$: 10 ms (max.) of the SMBus specification or the total of measured periods exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (min.) of the SMBus specification, the master device must stop the transaction by issuing a STOP condition. In master transmit mode, immediately stop the transmit operation (writing data to NTDTBP0).

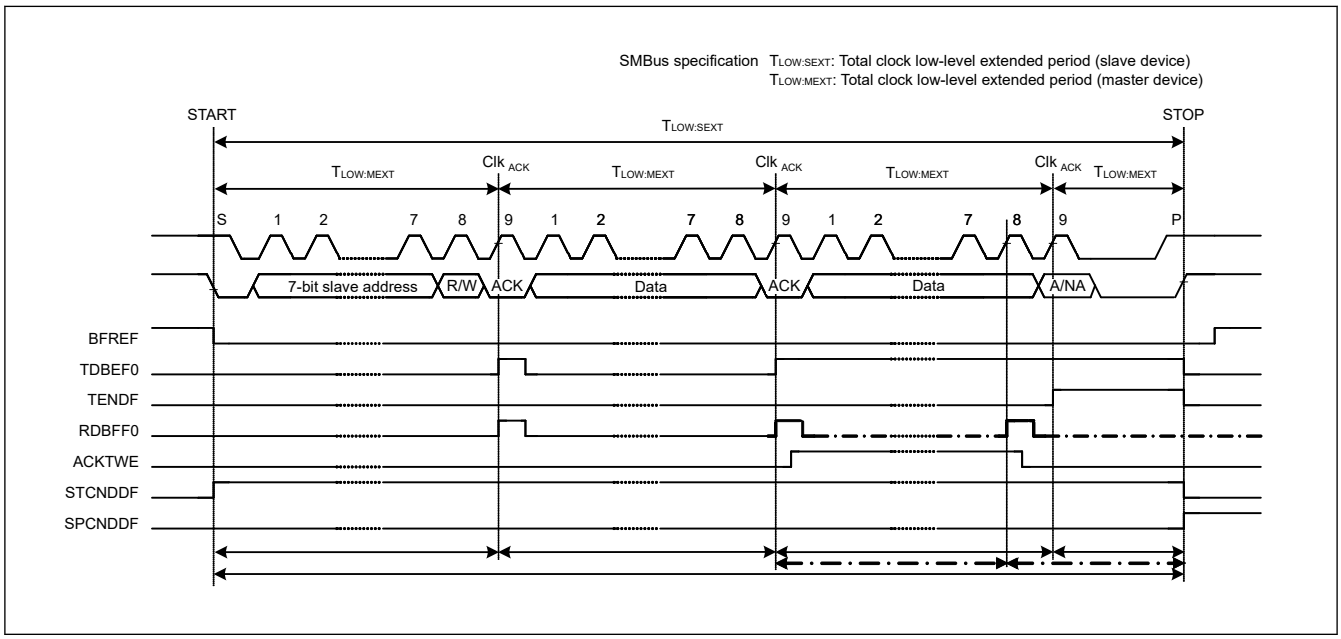


Figure 33.114 SMBus timeout measurement

(2) Packet Error Code (PEC)

This MCU incorporates a CRC calculator. The CRC calculator enables transmission of a packet error code (PEC) or checking the received data of the SMBus in data communication of I3C. For the CRC generating polynomials of the CRC calculator, see section 41, Cyclic Redundancy Check (CRC).

The PEC data in master transmit mode can be generated by writing all transmit data to the CRC data input register (CRCDIR) in the CRC calculator.

The PEC data in master receive mode can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC data output register (CRCDOR) with the received PEC data.

To send ACK or NACK according to the match or mismatch result when the final byte is received as a result of the PEC code check, set the SCSTRCTL.ACKTWE bit to 1 before the rising edge of the eighth SCL clock cycle during reception of the final byte, and hold the I3C_SCL line low at the falling edge of the eighth clock cycle.

(3) SMBus Host Notification Protocol (Notify ARP Master Command)

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of its own slave address or to request its own slave address from the SMBus host.

For a product of this MCU to operate as an SMBus host (or ARP master), the host address (0001 000) sent from the slave device must be detected as a slave address, so I3C has a function for detecting the host address. To detect the host address as a slave address, set the BFCTL.SMBS bit and the SVCTL.HOAE bit to 1. Operation after the host address has been detected is the same as normal slave operation.

33.3.2.3.13 Common Command Codes (CCC) [I3C mode]

For the common command code (CCC), refer to 5.1.9 Common Command Codes (CCC) in MIPI I3C Specification v1.0. I3C is based on Table 15 I3C Common Command Codes in 5.1.9.3 Common Command Definitions of MIPI I3C Specification v1.0.

The MIPI Reserved area and Vendor Extension area of Command Code are described below.

I3C Master mode :

When sending CCCs in the MIPI Reserved area and Vendor Extension area from the I3C Master, only Broadcast/Direct SET CCCs using the Immediate Transfer Command can be sent.

Sending Direct GET CCC is not supported.

I3C Slave mode :

Only Broadcast/Direct SET CCC can be received for CCC in MIPI Reserved area and Vendor Extension area.

Receiving Direct GET CCC is not supported.

33.3.2.4 Error Detection

33.3.2.4.1 SDR Error Detection and Recovery Methods for I3C Slave Devices [I3C mode]

The seven error types summarized in Table 33.13 are supported for all I3C slave devices. Each error type is further explained below the table.

Table 33.13 SDR slave error types

Error type	Description	Error detection method	Error recovery method
S0	Broadcast address/W (= 0x7E/W) or Dynamic address/RW	Detect any of the following: 0x3E / W 0x5E / W 0x6E / W 0x76 / W 0x7A / W 0x7C / W 0x7F / W 0x7E / R	Enable HDR EXIT Detector and ignore all other patterns
S1	CCC code	Parity check, using T-Bit	Enable HDR EXIT detector and neglect other patterns
S2	Write data	Parity check, using T-Bit	Enable STOP detector and neglect other patterns
S3	Assigned address during Dynamic address arbitration	Parity check, using PAR Bit	Generate NACK (after PAR), then wait for another Repeated START and 7E/R to re-transmit the Provisional ID
S4	0x7E/R after Sr during Dynamic address arbitration	Detect any value other than 0x7E/R after Sr during Dynamic Address Arbitration	Generate NACK (after 0x7E/R), then enable STOP Detector and ignore all other patterns
S5	Transaction after detecting CCC	Detect illegally formatted CCC	Generate NACK (after Slave Address), then enable STOP Detector and ignore all other patterns
S6 (optional)	Monitoring error	Slave detects (through monitoring) that transmitted Data differs from what it intended to transmit (Does not apply during Dynamic address arbitration)	Stop the transmission, then enable STOP Detector and ignore all other patterns

33.3.2.4.2 SDR Error Detection and Recovery Methods for I3C Master Devices [I3C mode]

The two error types summarized in Table 33.14 are supported for all I3C master devices. Each error type is further explained below the table.

Table 33.14 SDR master error types

Error type	Description	Error detection method	Error recovery method
M0	Transaction after sending CCC	Detect illegally formatted CCC	Stop the transmission, then send STOP and retry the transmission.
M1 (optional)	Monitoring error	Master detects (through monitoring) transmitted data different from what it intended to transmit (Does not apply during Dynamic address arbitration)	Stop the transmission, then send STOP and retry the transmission.
M2	No response to Broadcast address (0x7E)	Master detects NACK after Broadcast address (0x7E) transmission	Upon detection of NACK, master transmits HDR exit pattern followed by STOP

33.3.2.4.3 HDR-DDR Error Detection

Four error types are defined for HDR-DDR mode.

Table 33.15 HDR-DDR error types

Error type	Description	Error detection method	Error recovery method
Framing	The Preamble 2-bits before command and data	The Slave detects that the Preamble 2-bits before command and data is not valid values.*1	Master: Issue SCL clocks until 19 SCL clocks (38 bits) have been seen with SDA High. The Master shall then Park SCL Low, and emit an HDR Exit Pattern using SDA. Slave: Wait for HDR EXIT Pattern
Parity checking	Parity error	The Master and the Slave detects mismatched parity.*2	
CRC5 checking	CRC5 error	The Master and the Slave detects mismatched CRC.*3	
NACK receiving	NACK by the Slave on a Read command	The Master detects a NACK by the Slave on a Read command.*4	Issue SCL clocks until 19 SCL clocks (38 bits) have been seen with SDA high. The Master shall then Park SCL low, and emit either an HDR exit pattern or an HDR restart pattern using SDA.
Monitoring	Monitoring error	Both the Master and the Slave should always monitor the data that they each transmit, and they detect the monitored data differs from the data that the Master or Slave intended to send.*5	Both the Master and the Slave can stop the transmission and then Master and Slave should each do the following. Master: Issue SCL clocks until 19 SCL clocks (38 bits) have been seen with SDA High. The Master shall then Park SCL Low, and emit an HDR Exit Pattern using SDA. After the Master sends HDR EXIT, it can retry the transmission. Slave: Wait for HDR Exit Pattern

Note 1. This supports a positional error detection mechanism:

- A Command Word shall always follow the Enter HDR CCC and HDR Restart Pattern, and shall never appear in any other position. It is an error condition for a Command Word to appear in any other position, or to be missing where expected.
- A Data Word shall always follow either a Command Word or another Data Word, and shall never appear in any other position. It is an error condition for a Data Word to appear in any other position, or to be missing where expected.
- A single CRC Word shall always follow the last Data Word for a Command, such that it ends the Message. As a result, a CRC Word shall always be followed by either the HDR Restart Pattern or the HDR Exit Pattern. It is error condition for a CRC Word to appear in any other position, or to be missing where expected.
- The first nibble of a valid CRC shall contain the allowed value 0xC. Any other value in the first nibble should be considered a framing error.

Note 2. Parity encoding (for transmitters) parity checking (for receivers) shall be performed on all Command Words and Data Words. Mismatched parity is an error condition.
Note 3. CRC5 encoding (for transmitters) and CRC5 checking (for receivers) shall be performed on all payload bits for Command Words and Data Words. Mismatched CRC is an Error.
Note 4. NACK by the Slave on a Read command is not a normal behavior (ACK is normal). The Master may choose to treat a NACK on a Read command as a possible framing error. The Master can choose to treat the NACK of a Read command as a possible line error, and therefore use the same approach in order to ensure that the Slave is not driving the Bus.
Note 5. If the Master or Slave performs such monitoring, and if the monitored Data differs from the Data that the Master or Slave intended to send (except for Data transferred during a Dynamic Address Arbitration procedure), then this shall be considered an error.

33.3.2.4.4 HDR-TSP/TSL Error Detection

Two error types are defined for HDR-TSP/TSL Mode.

Table 33.16 HDR-TSP/TSL error types

Error type	Description	Error detection method	Error recovery method
Symbol 2 checking	Continuous error of symbol 2	More than one symbol 2 in a row. (In symbol 2 the I3C_SCL line does not change, and the I3C_SDA line does change.) *1	Master: Wait until the Slave stops transitioning the Bus for 2x the maximum edge-to-edge duration used by that Slave. Then the Master shall force out the HDR Exit Pattern. Slave: Wait for HDR Exit Pattern.
Parity checking	Parity errors	The Slave detects mismatched parity. *2	
Monitoring	Monitoring error	Both the Master and the Slave should always monitor the data that they each transmit, and they detect the monitored data differs from the data that the Master or Slave intended to send.*3	Both the Master and the Slave can stop the transmission and then Master and Slave should each do the following. Master: Wait until the Slave stops transitioning the Bus for 2x the maximum edge-to-edge duration used by that Slave. Then the Master shall force out the HDR Exit Pattern. After the Master sends HDR EXIT, it can retry the transmission. Slave: Wait for HDR EXIT Pattern.

Note 1. Exception: More than one Symbol 2 in a row is allowed for HDR Restart or HDR Exit, but only from a known starting state (SCL Low and SDA High), and only at a Data Word boundary (allowing for one Symbol to set up that state). Thus the Symbol pattern 2,2 is only allowed under the following situations:
- As part of the HDR Exit Pattern, which uses 3 or 4 such Symbol pairs and involves SCL Low
- As part of the HDR Restart pattern, which uses 2 such Symbol pairs and involves SCL Low
- As the result of one normal Symbol ending in 2, and the next Symbol starting with 2

Note 2. An error may be seen by an HDR Slave, or by an HDR Master when a Slave is driving the I3C Bus:
- A Slave that sees an error shall stop tracking Symbols and use HDR Exit and HDR Restart Pattern detection. This means to wait until safe. A Slave shall enable the HDR Exit and HDR Restart Pattern Detector upon entry to any HDR Mode, ensuring that it is always safe.
- A Master that sees an error shall wait until the Slave stops transitioning the Bus for 2x the maximum edge-toedge duration used by that Slave. Then the Master shall force out the HDR Exit Pattern.

Note 3. If the Master or Slave performs such monitoring, and if the monitored Data differs from the Data the Master or Slave intended to send (except for data transferred during the Dynamic Address Arbitration procedure), then this shall be considered an error.

Note: It could be appropriate to wait for the Symbol 2,2, but the Master would still need to see quiescence. If the Master is seeing errors, then the results of even a Symbol 2,2 are questionable.

33.3.2.4.5 Timeout Error Detection

I3C includes a timeout function for detecting when the I3C_SCL line has been stuck longer than the predetermined time. I3C can detect an abnormal bus state by monitoring that the I3C_SCL line is stuck low or high for a predetermined time.

The timeout function monitors the I3C_SCL line state and counts the low-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the I3C_SCL line changes (rising or falling), but continues to count unless the I3C_SCL line changes. If the internal counter overflows due to no I3C_SCL line change, I3C can detect the timeout and report the bus hung state.

This timeout function is enabled when BSTE.TODE = 1. It detects a hung state that the I3C_SCL line is stuck low or high during the following conditions: (When TMOCTL.TOMDS[1:0] = 00b)

- The bus is busy (BCST.BFREF = 0) in master mode (PRSST.CRMS = 1).
- I3C's own slave address is detected (SVST register is not 0x0000) and the bus is busy (BCST.BFREF = 0) in slave mode (PRSST.CRMS = 0).
- The bus is free (BCST.BFREF = 1) while generation of a START condition is requested (CNDCTL.STCND = 1).

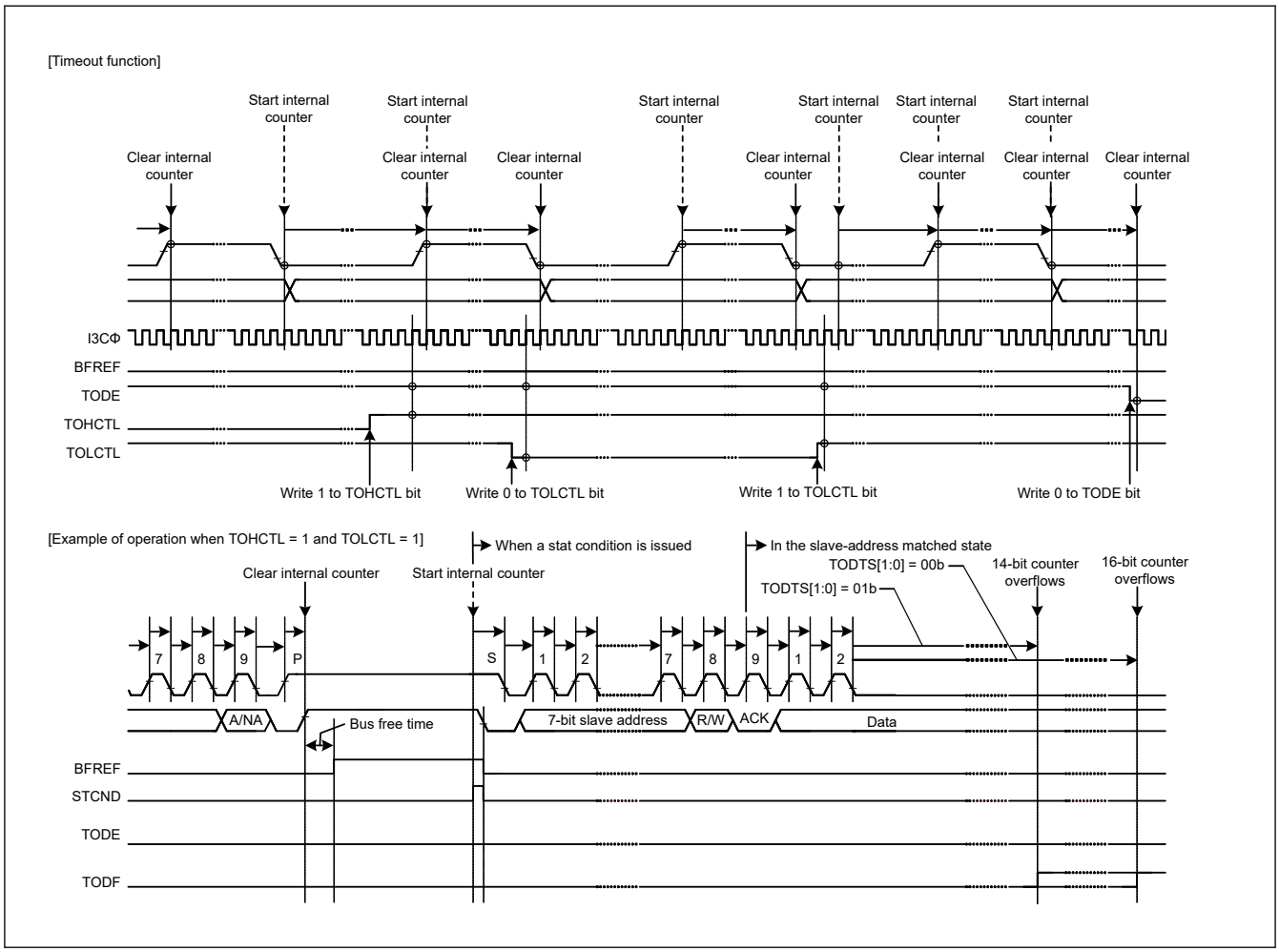


Figure 33.115 Timeout error detection (TODE, TODTS[1:0], TOHCTL, and TOLCTL bits)

33.3.2.4.6 Resume Operation [I3C mode]

I3C enters the Halt state as a result of any type of error occurring in a transfer.

The error type is indicated by the field ERR_STATUS in Response Descriptor or Receive Status Descriptor. After I3C has entered the Halt state, the user must write the value 1 to the RSM bit to resume operation. I3C shall auto-clear the RSM bit once it has initiated the next Command transfer or detected the START condition.

33.3.2.4.7 Abort Operation [I3C mode]

When the BCTL.ABT bit is set to 1, I3C relinquish control of the bus before completing the currently issued transfer. In response to an abort request, I3C issues the STOP condition on the bus after the complete data byte is transferred or received. After I3C has aborted, the user shall clear the BCTL.ABT bit to allow operation on the bus.

Note: For Read transaction, when BCTL.ABT is set to 1, that receive data is stored in Rx data buffer. However, for HDR-TSP / TSL, receive data received after that is not stored.

Abbreviations

Pa: Parity

Pr: Preamble

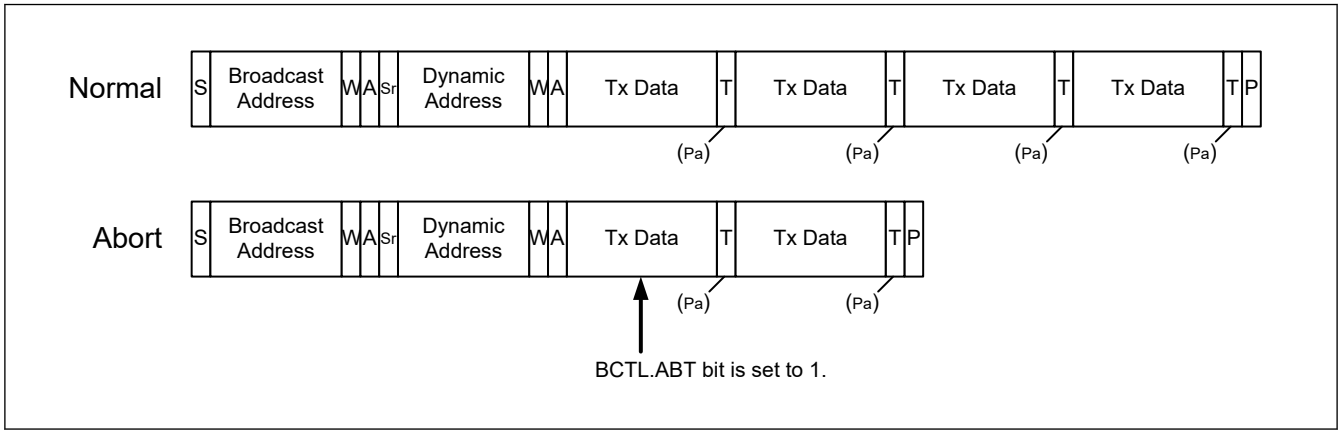


Figure 33.116 Abort operation of SDR write transfer

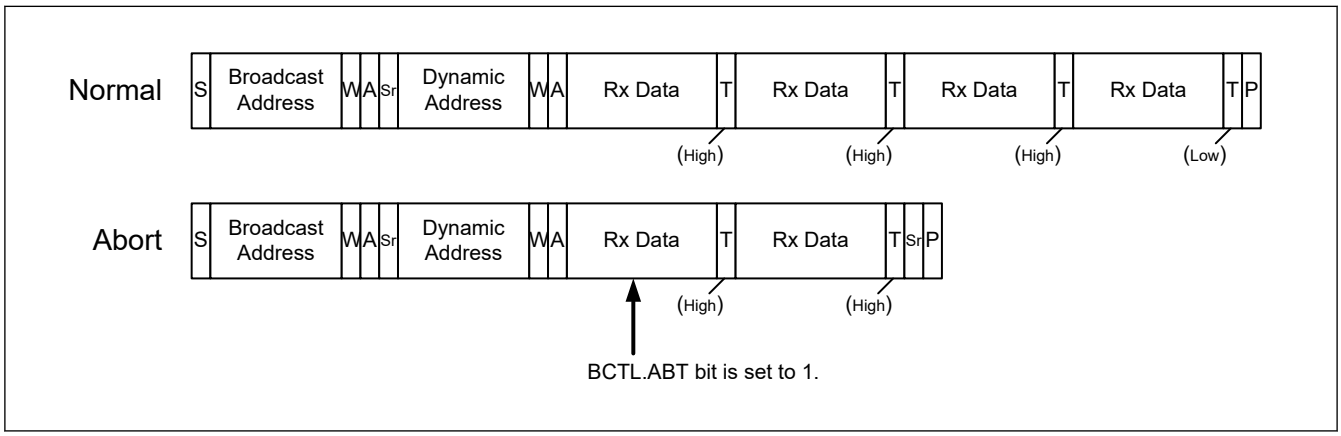


Figure 33.117 Abort operation of SDR read transfer

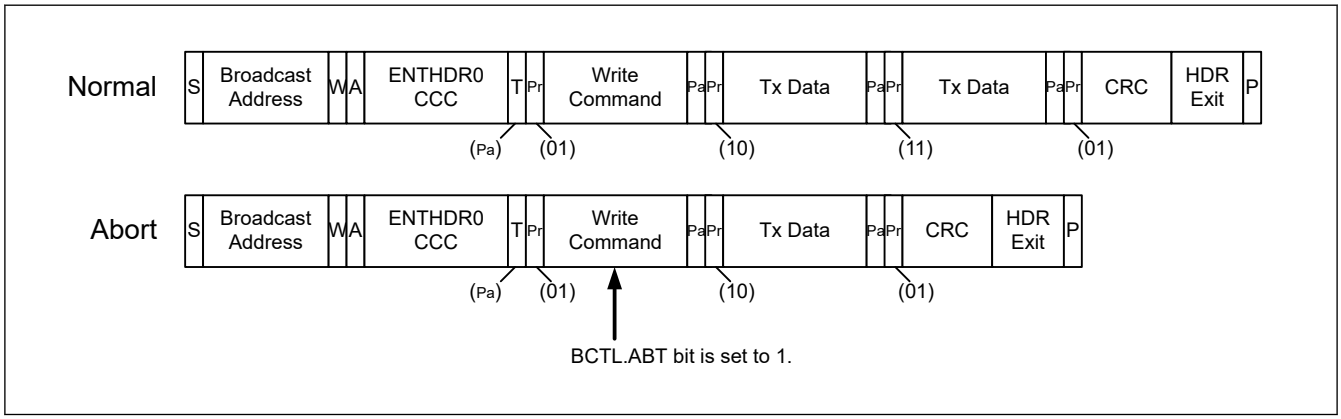


Figure 33.118 Abort operation of HDR-DDR write transfer

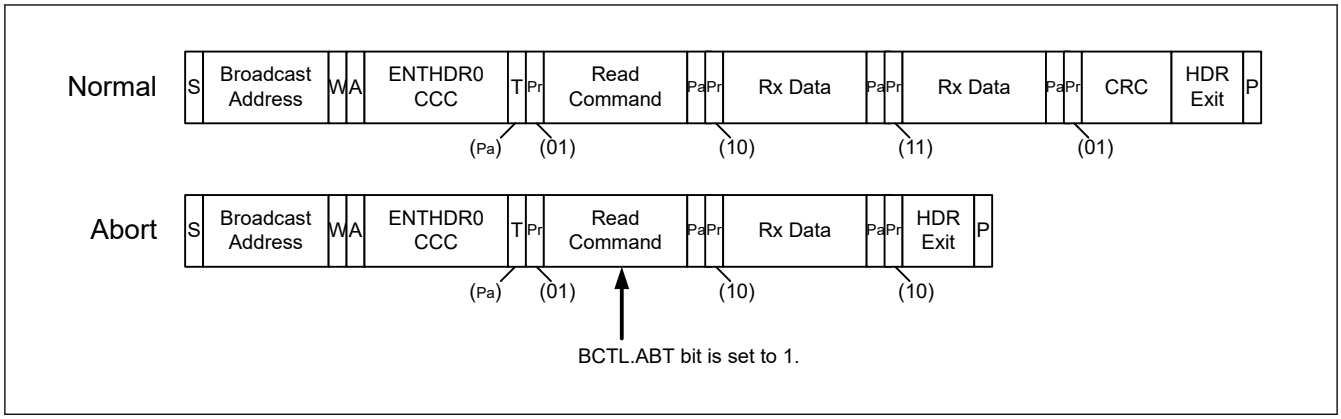


Figure 33.119 Abort operation of HDR-DDR read transfer

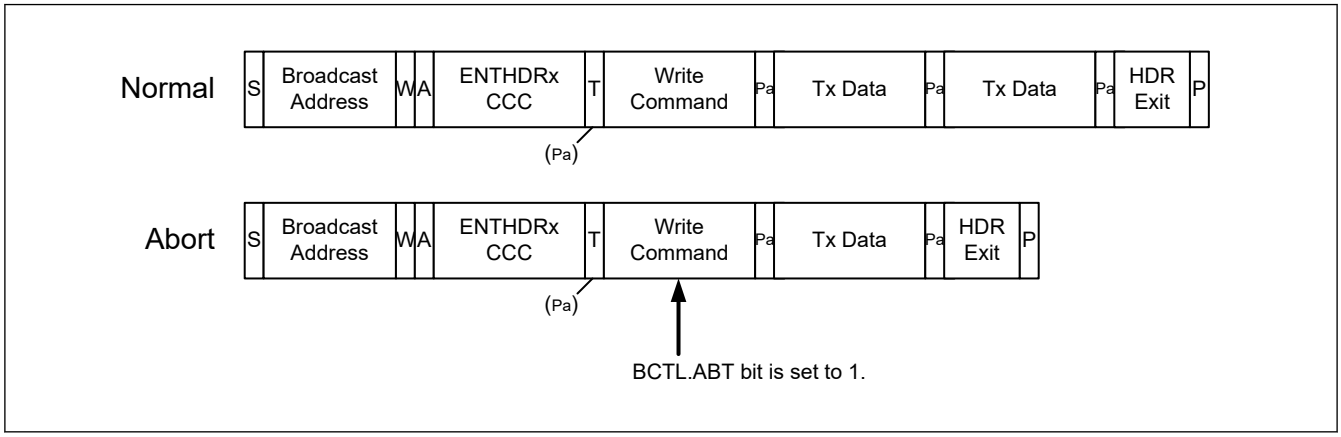


Figure 33.120 Abort operation of HDR-TSP/TSL write transfer

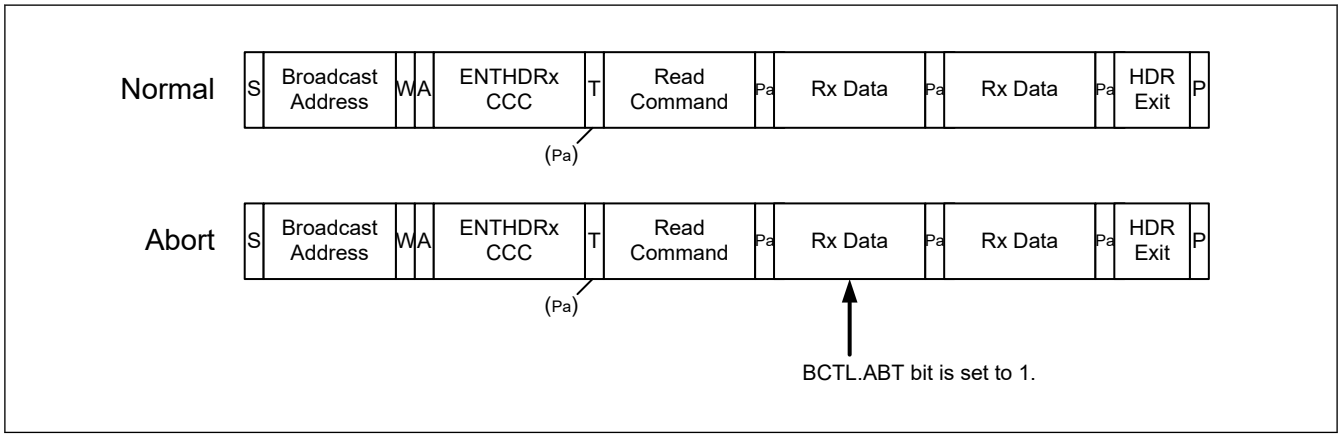


Figure 33.121 Abort operation of HDR-TSP/TSL read transfer

33.3.2.4.8 Error Recovery Operation

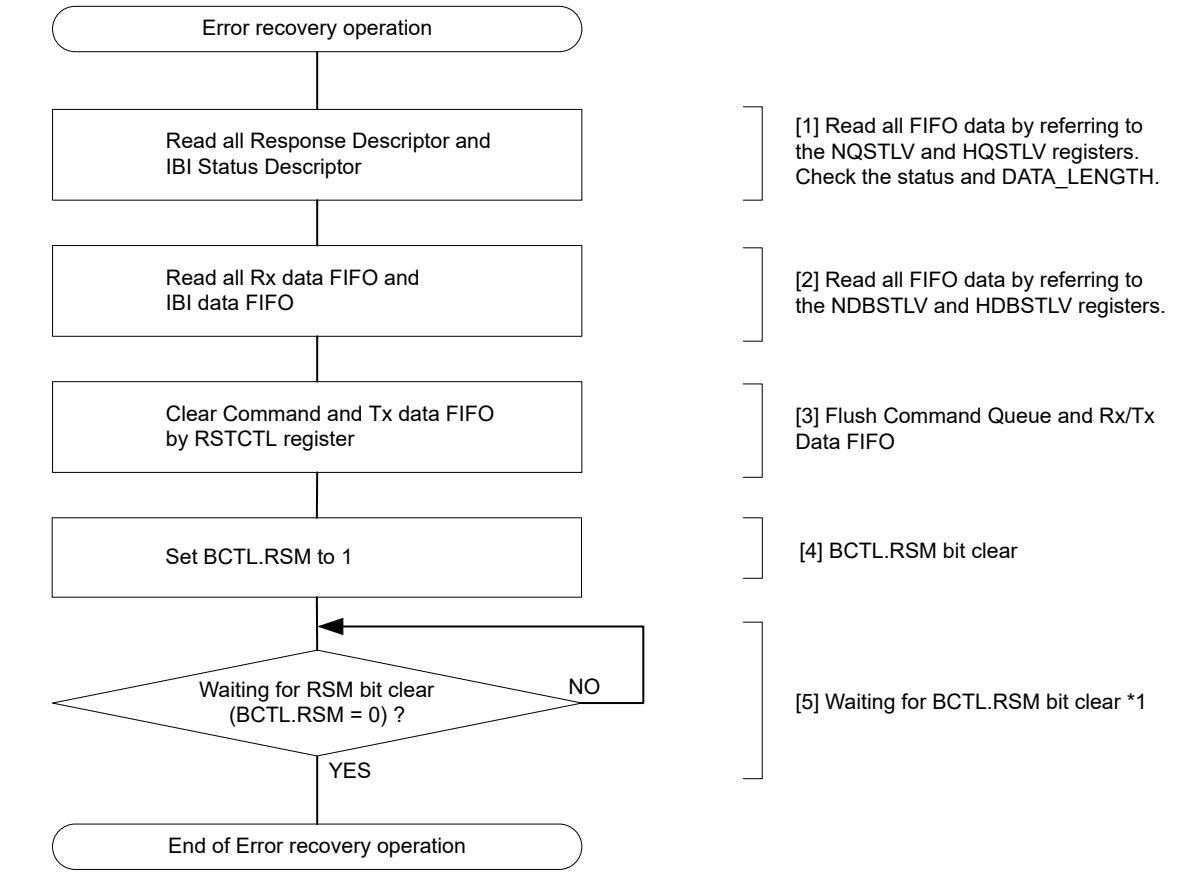
(1) Error Recovery Operation

When an error occurs, the INST.INEF, NTST.TEF, NTST.TABTF, HTST.TEF and HTST.TABTF flags are set to 1 according to the cause of the error, or the interrupts associated with each flag are asserted (when detection and interrupts are enabled.)

There is a possibility of communication error or internal module error.

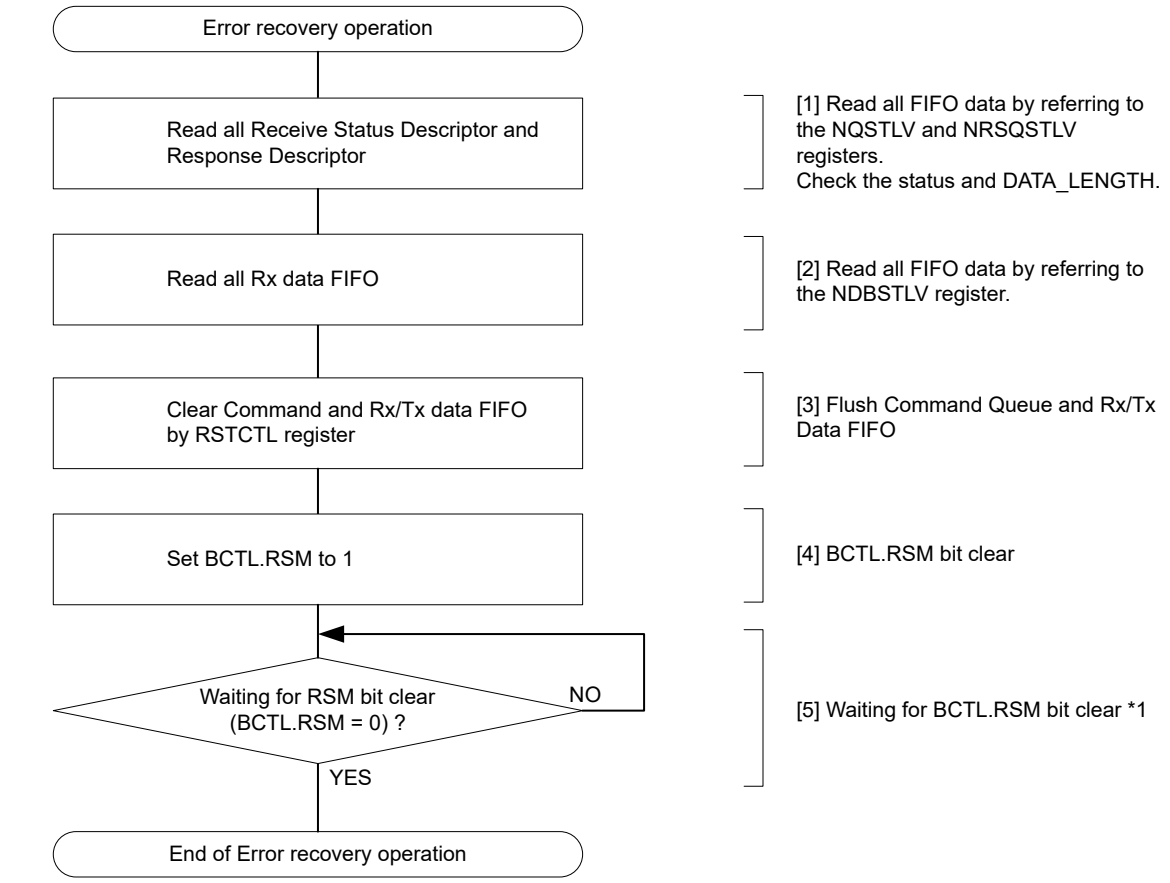
Note: If an error occurs, I3C will be suspended (BCTL.RSM becomes 1). After I3C is suspended, the application must write the value 1 to the BCTL.RSM bit to resume I3C operaion and recover from the suspended state.

Figure 33.122 and Figure 33.123 show the error recovery flow.



Note 1. The Command Descriptor can be written to the Command Queue via writing to the Command Queue Port (NCMDQP or HCMDQP) before the RSM bit is cleared.

Figure 33.122 Example of error recovery operation flowchart for I3C master



Note 1. The Command Descriptor for issuing IBI can be written to the Command Queue via writing to the Command Queue Port (NCMDQP) before the RSM bit is cleared.

Figure 33.123 Example of error recovery operation flowchart for I3C slave

When I3C Slave recovers from an error according to the error recovery flow, after setting BCTL.RSM to 1, BCTL.RSM becomes 0 after detecting a state in which Bus Available period communication is not performed on I3C Bus.

If communication occurs on the I3C bus within the Bus Available period, BCTL.RSM is not set to 0 and error recovery is not completed, NACK response will be made to the communication.

(2) Master Error Detection and Escalation Handling

If the Master does not receive an ACK of a transmitted private message to a slave and steps 1 and 2 described in Chapter 5.1.10.2.4 of MIPI I3C Spec v1.0 fail, the processing flow of step 3 is shown in Figure 33.124 and Figure 33.125.

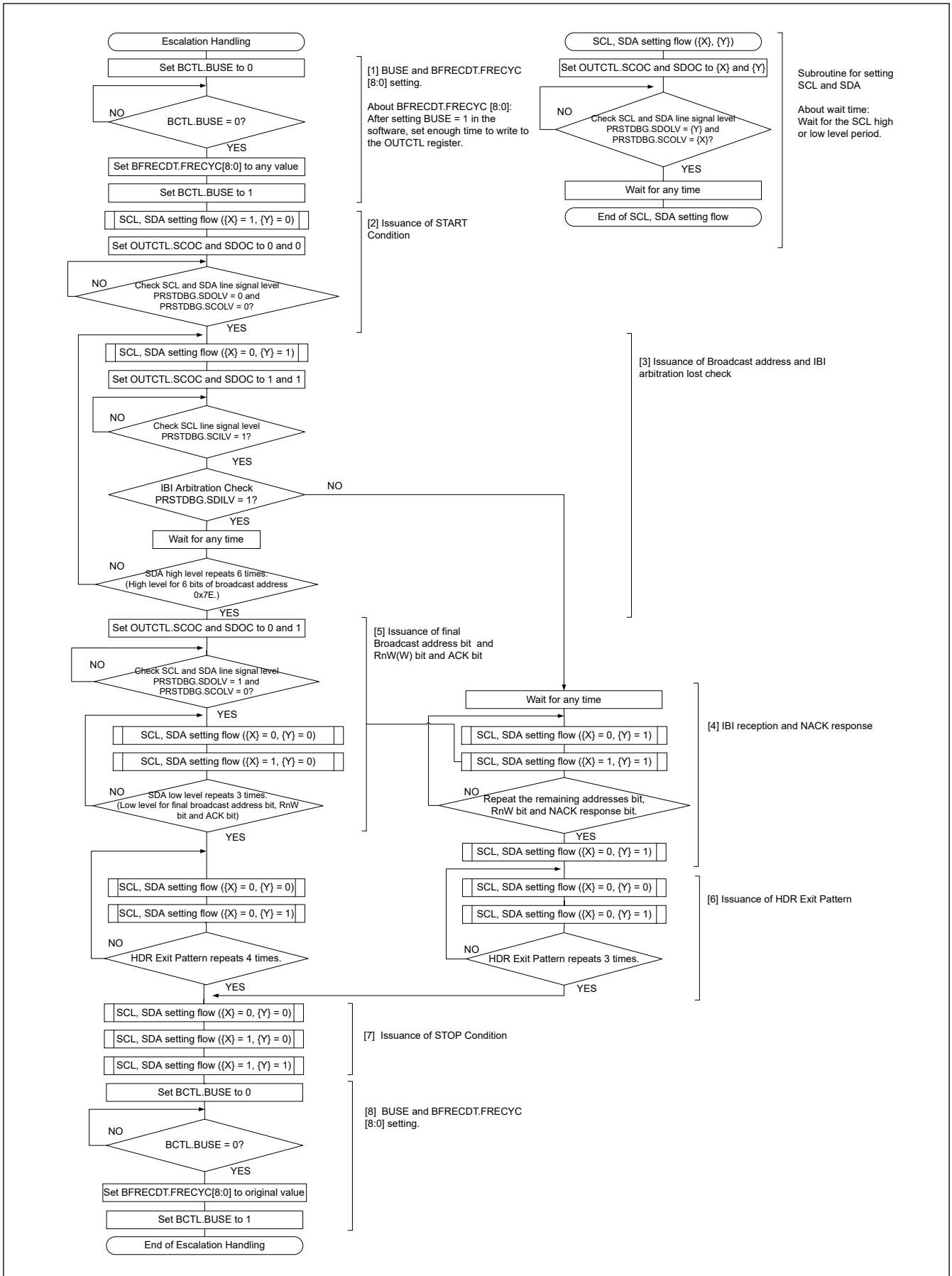


Figure 33.124 Escalation handling flowchart for I3C master

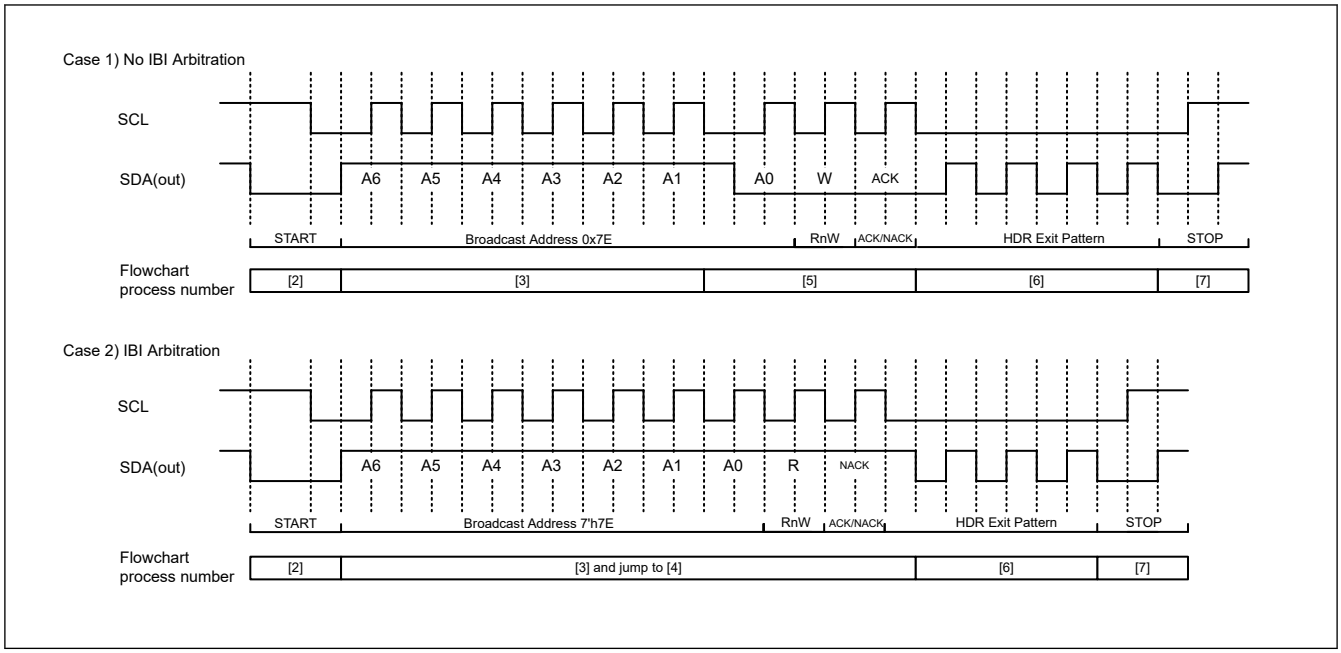


Figure 33.125 I3C escalation handling timing diagram for I3C Master

33.3.2.5 Low Power Function

33.3.2.5.1 Wake Up function [I^2C mode]

I3C is equipped with the Wake-up function that causes the microcomputer to transition from low power consumption mode with system clock (ICLK) is stopped (software standby mode, etc.) to the normal operation. The Wake-up function is used to generate a Wake-up interrupt signal when the received data matches the address set to Wake-up interrupt factor also receives data in a state where the operating clock (PCLK/TCLK) is stopped (PCLK/TCLK asynchronous operation). This wake-up interrupt signal causes the microcomputer to transition to the normal operation. After Wake-up interrupt occurs, switch I3C to PCLK/TCLK synchronous operation, it will be able to continue the communication operation.

The Wake-up function has four wake-up operation modes (normal WU mode 1, normal WU mode 2, command recovery mode, and EEP response mode). The table below describes the behavior in these four wake-up operation modes.

Table 33.17 Wake-up operation mode

	ACK response timing	ACK Type responded before recovery to PCLK/TCLK synchronous operation	SCL state before recovery to PCLK/TCLK synchronous operation
Normal WU mode 1	Before recovery to PCLK/TCLK synchronous operation*1	ACK	Fixed to L
Normal WU mode 2	After recovery to PCLK/TCLK synchronous operation*2	Before recovery: no response (NACK level retained) After recovery: ACK response	Fixed to L
Command recovery mode	Before recovery to PCLK/TCLK synchronous operation*1	ACK	Open
EEP response mode	Before recovery to PCLK/TCLK synchronous operation*1	NACK	Open

Note 1. Switching timing from PCLK/TCLK asynchronous operation to PCLK/TCLK synchronous operation is the fall of the 9th clock of SCL.
Note 2. Switching timing from PCLK/TCLK asynchronous operation to PCLK/TCLK synchronous operation is the fall of the 8th clock of SCL.

The following is able to select as Wake-Up interrupt factor.

- Host address detection (valid when SVCTL.HOAE = 1)
- General call address detection (valid when SVCTL.GCAE = 1)
- Slave address 0*1 detection (valid when SVCTL.SVAE[0] = 1)
- Slave address 1*1 detection (valid when SVCTL.SVAE[1] = 1)

- Slave address 2[*1] detection (valid when SVCTL.SVAE[2] = 1)

Note 1. 7-bit address only can be set. Set SDADLS bit to 0 in SDATBASn .

(1) Normal Wake-Up mode 1

This section describes the behavior, the timing, and a use case of normal WU mode 1.

A wake-up interrupt triggered by the match of the slave address makes the transition to the normal operation in the manner described below. Also, the detailed timing is provided in Figure 33.128.

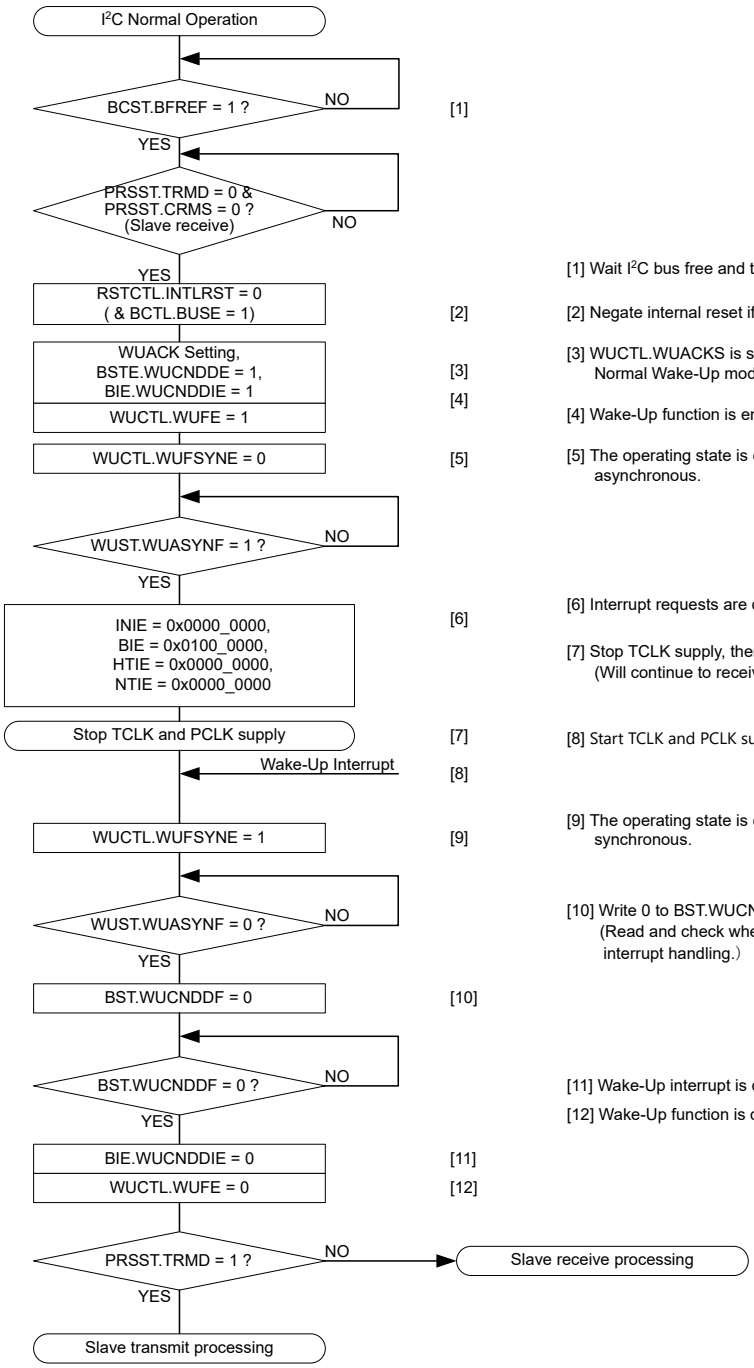
Before wake-up recovery:	ACK is sent in response to the data received with its own slave address.
During wake-up recovery:	ACK response is made at the 9th clock cycle of SCL, and the SCL is held low afterwards.[*1]
After wake-up recovery:	Normal operation continues.

Note 1. Between ninth clock cycle and first clock cycle during Wake-Up recovery, SCSTRCTL.RWE = 1 does not work.

If the slave address does not match, the I3C_SCL line is not held low after the fall of the 9th clock cycle of SCL, and the slave operation continues.

See Figure 33.126 below for a use case.

A wake-up interrupt is not generated at the transition to the normal operation if the transition is triggered by a cause (other recovery causes (IRQ)) other than a wake-up interrupt signal generated by a slave address match. BST.WUCNDDF is not set in this case. Carry out the following processing according to Figure 33.127.



[1] Wait I2C bus free and this module stay in the standby state.

[2] Negate internal reset if assert the internal reset.

[3] WUCTL.WUACKS is set up according to each mode. (Normal Wake-Up mode 1/ Normal Wake-Up mode 2) Wake-Up Interrupt is enabled.

[4] Wake-Up function is enabled.

[5] The operating state is changed from PCLK/TCLK synchronous to PCLK/TCLK asynchronous.

[6] Interrupt requests are disabled except WUI.

[7] Stop TCLK supply, then stop PCLK supply. (Will continue to receive)

[8] Start TCLK and PCLK supply by Wake-Up interrupt (address match).

[9] The operating state is changed from PCLK/TCLK asynchronous to PCLK/TCLK synchronous.

[10] Write 0 to BST.WUCNDDF. (Read and check whether clearing has been completed befor return from interrupt handling.)

[11] Wake-Up interrupt is disabled.

[12] Wake-Up function is disabled.

Figure 33.126 Use case of normal WU mode 1 (wake-up recovery by a wake-up interrupt triggered by the match of the slave address)

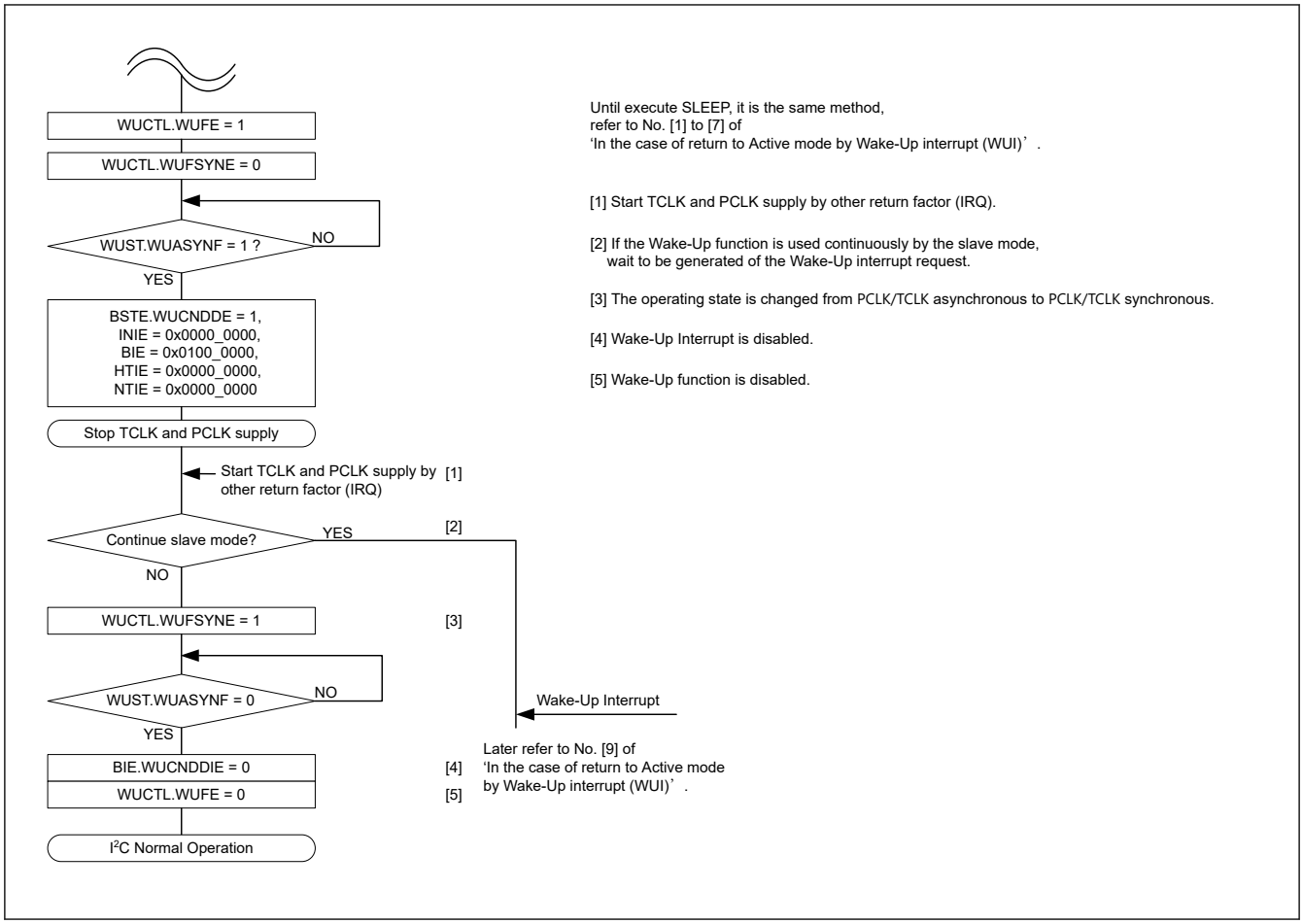


Figure 33.127 Use case of normal WU modes 1 and 2 (wake-up recovery by other recovery causes (IRQ))

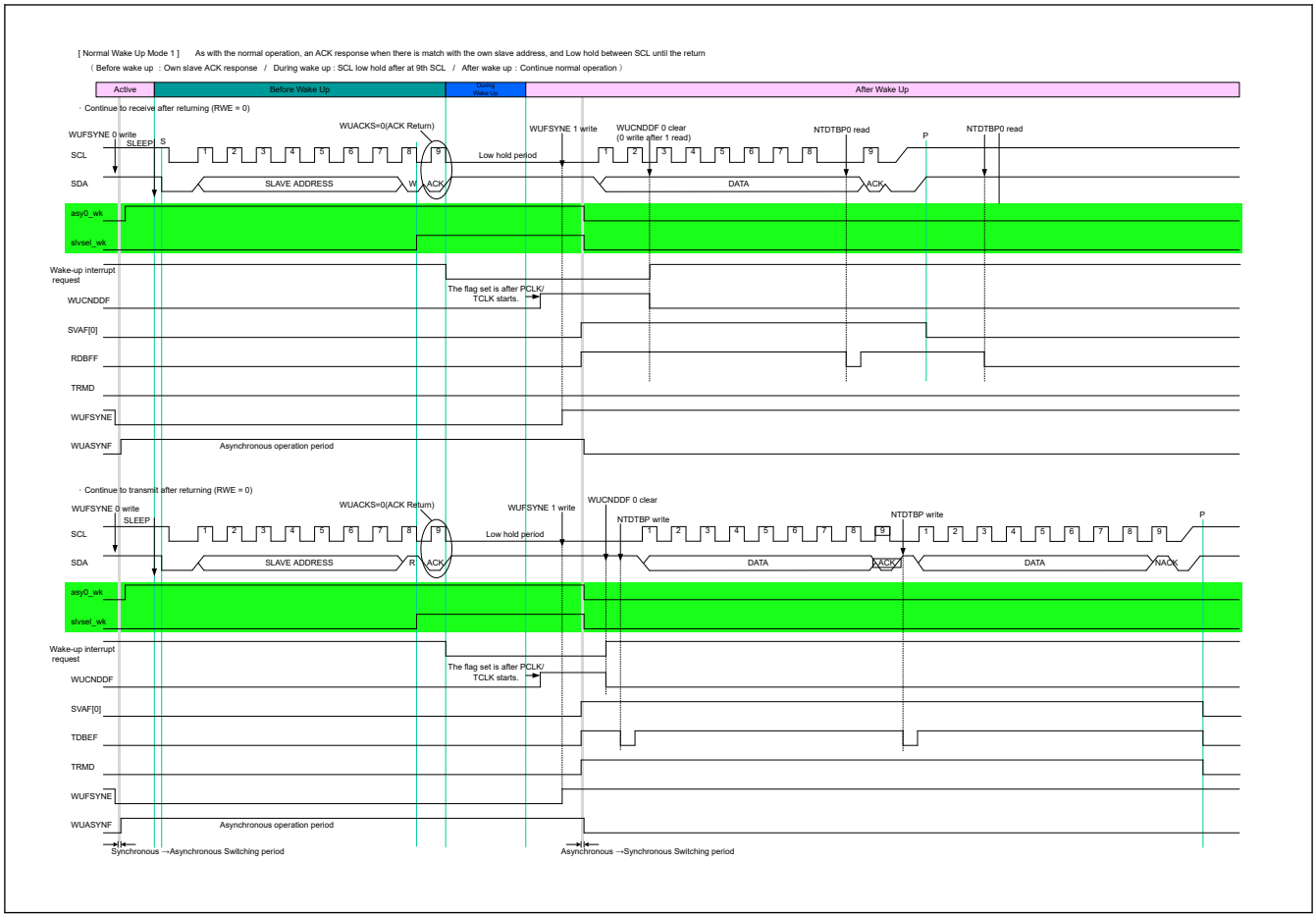


Figure 33.128 Timing of normal wake up mode 1

(2) Normal Wake Up Mode 2

This section describes the behavior, the timing, and a use case of normal WU mode 2.

A wake-up interrupt triggered by the match of the slave address makes the transition to the normal operation in the manner described below.

Also, the detailed timing is provided in Figure 33.130.

Before wake-up recovery:	No response to the data received with its own slave address (until 8th SCL cycle end)
During wake-up recovery:	Holding the I3C_SCL line low during the 8th and 9th clock cycles
After wake-up recovery:	Returning ACK at the 9th clock cycle of SCL, and continuing the normal operation

If the slave address does not match, the I3C_SCL line is not held low after the fall of the 8th SCL v clock cycle. The slave operation continues.

See Figure 33.129 below for a use case.

A wake-up interrupt is not generated at the transition to the normal operation if the transition is triggered by a cause (other recovery causes (IRQ)) other than a wake-up interrupt signal generated by a slave address match. BST.WUCNDDF is not set in this case. Carry out the following processing according to Figure 33.127.

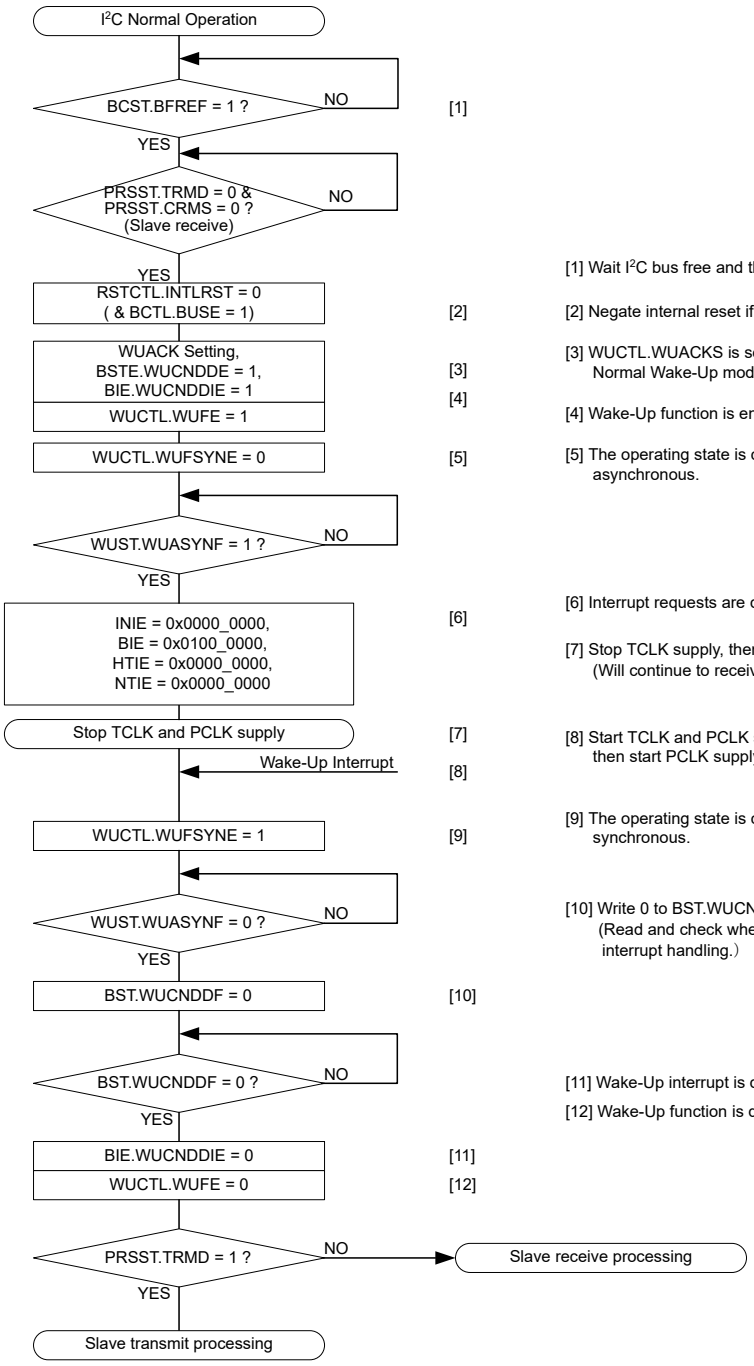


Figure 33.129 Use case of normal WU mode 2 (wake-up recovery by a wake-up interrupt triggered by the match of the slave address)

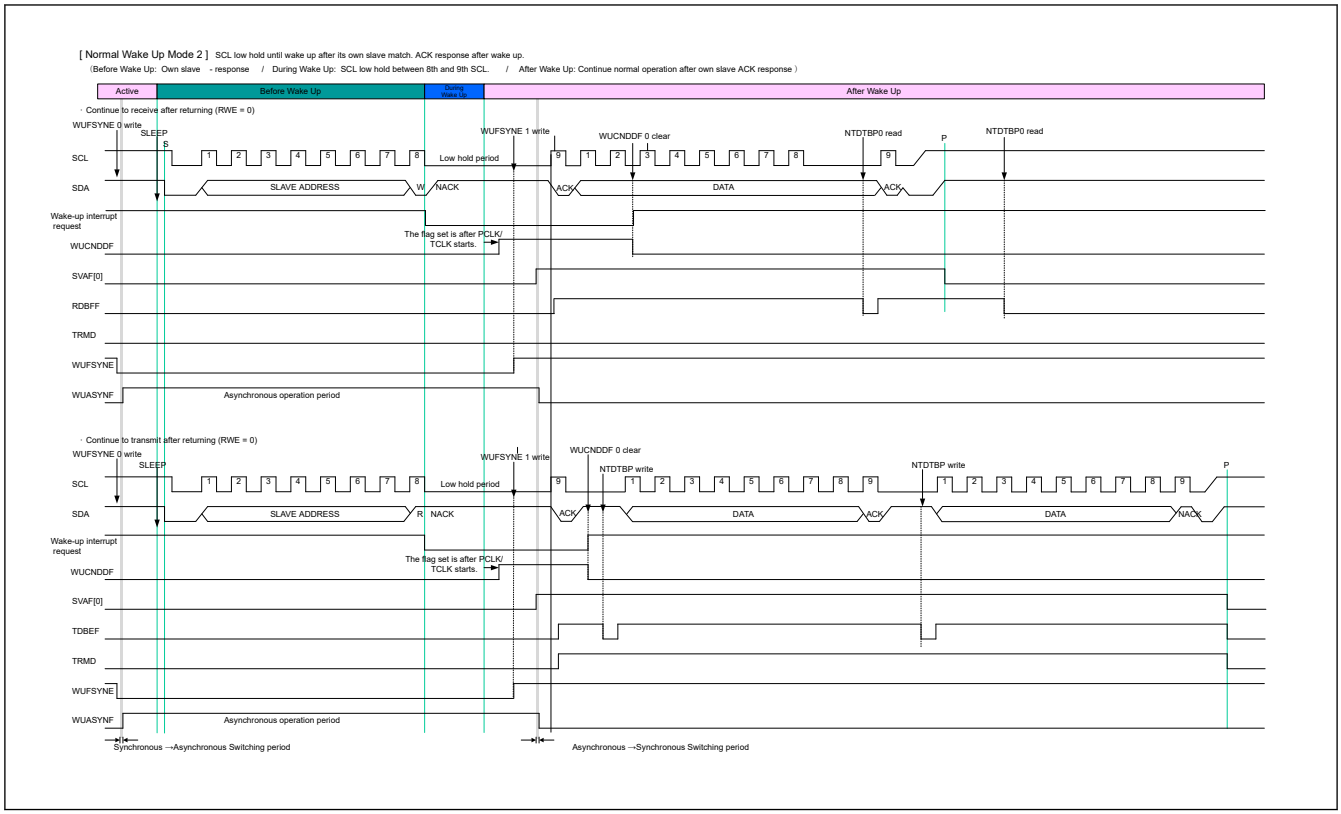


Figure 33.130 Timing of normal wake up mode 2

(3) Command recovery mode/ EEP response mode (Special Wake Up mode)

In the command recovery mode and EEP response mode, the I3C_SCL line is not held low during the wake-up recovery period (after the rise of the 9th clock cycle of SCL), so other I^2C/I3C devices can use the I^2C bus during this period. This section describes the behavior, the timing, and use cases of the command recovery mode and the EEP response mode.

A wake-up interrupt triggered by the match of the slave address makes the transition to the normal operation in the manner described below. Also, the detailed timing is provided in Figure 33.133.

Before wake-up recovery:	In response to the data received with its own slave address, ACK (command recovery mode) or NACK (EEP response mode) is returned.
During wake-up recovery:	The I3C_SCL line is not held low.
After wake-up recovery:	Normal operation continues after I3C initial setting.

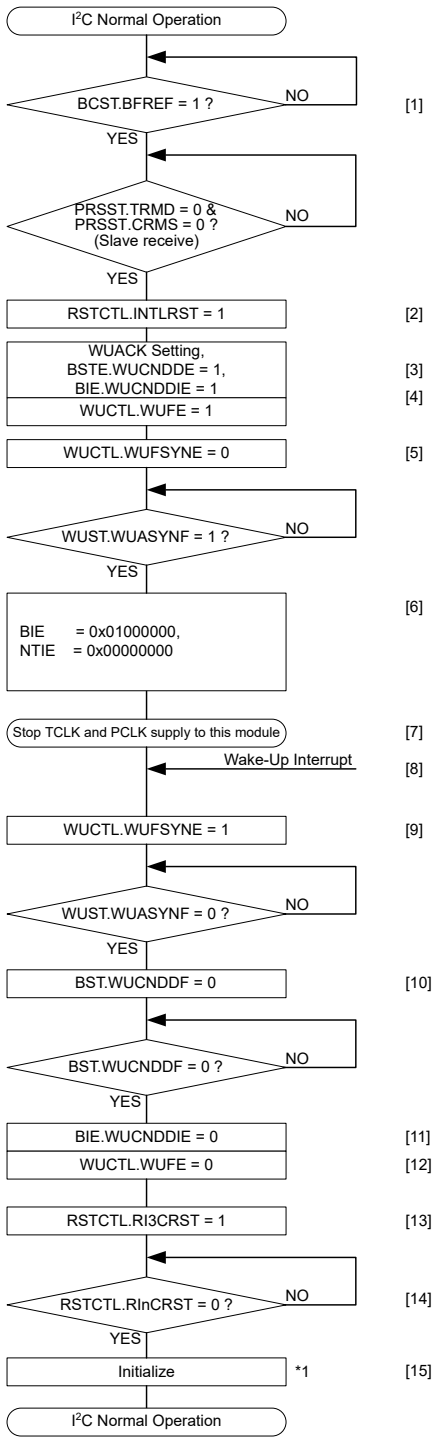
Note: Because the I3C_SCL line is not held low during wake-up recovery, the transmission/reception of the data that follows the slave address is not possible.

Note: The command recovery mode and the EEP response mode are internal reset (RSTCTL.INTLRST = 1) states. Therefore, the match of the slave address does not set the SVST flags (HOAF, GCAF, and SVAF[2:0]).

If the slave address does not match, the slave operation continues.

See Figure 33.132 below for a use case.

A wake-up interrupt is not generated at the transition to the normal operation if the transition is triggered by a cause (other recovery causes (IRQ)) other than a wake-up interrupt signal generated by a slave address match. BST.WUCNDDF is not set in this case. Carry out the following processing according to Figure 33.132.



[1] Wait I²C bus free and this module stay in the standby state.

[2] Assert the internal reset.

[3] WUCTL.WUACKS is set up according to each mode. (Command return mode/ EEP response mode) Wake-Up Interrupt is enabled.

[4] Wake-Up function is enabled.

[5] This module of operating state is changed from PCLK/TCLK synchronous to PCLK/TCLK asynchronous.

[6] Interrupt requests are disabled except WUI.

[7] Stop TCLK supply to this module, then stop PCLK supply to this module. (This module will continue to receive)

[8] Start TCLK and PCLK supply to this module by Wake-Up interrupt (address match), then start PCLK supply to this module.

[9] This module of operating state is changed from PCLK/TCLK asynchronous to PCLK/TCLK synchronous.

[10] Write 0 to BST.WUCNDDF. (Read and check whether clearing has been completed before return from interrupt handling.)

[11] Wake-Up interrupt is disabled.

[12] Wake-Up function is disabled.

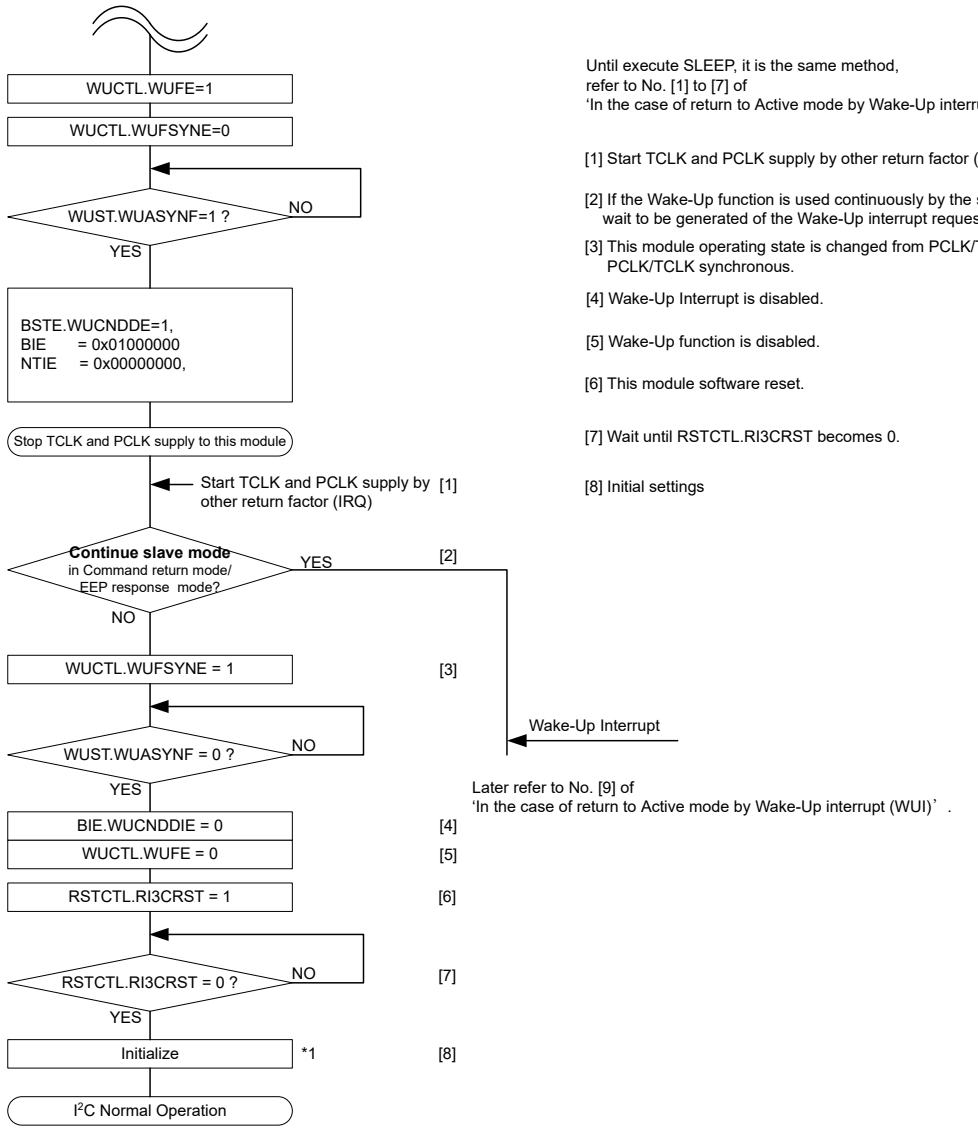
[13] Software reset.

[14] Wait until RSTCTL.RI3CRST becomes 0.

[15] Initial settings

Note 1. See section 33.3.3.1. Initial Setting Flow for details of initialization.

Figure 33.131 Use case of command recover mode and EEP response mode (wake-up recovery by a wake-up interrupt triggered by the match of the slave address)



Note 1. See section 33.3.3.1. Initial Setting Flow for details of initialization.

Figure 33.132 Use case of command recover mode and EEP response mode (wake-up recovery by other recovery causes (IRQ))

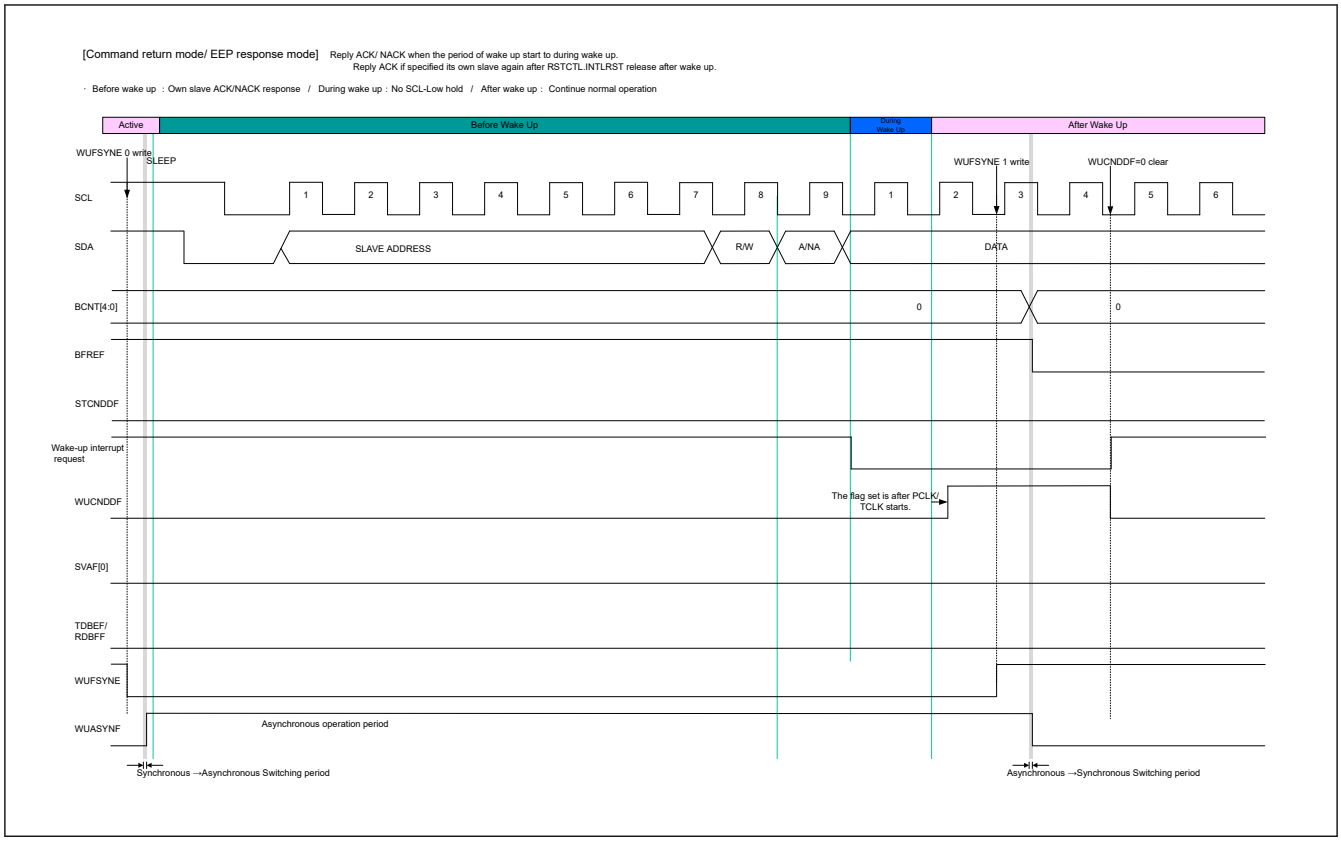


Figure 33.133 Timing of command recovery mode/EEP response mode

(4) Precautions on the use of the Wake-Up function

Precautions on the use of the Wake-up function is shown below.

- Do not change the registers in I3C except the WUCTL.WUFSYNE bit while the WUST.WUASYNF flag = 1 (while PCLK/TCLK asynchronous operation).
- Set WUCTL.WUFE = BSTE.WUCNDDE = BIE.WUCNDDIE = 1 and PRSST.CRMS = PRSST.TRMD = 0 (slave reception mode) before switching PCLK/TCLK asynchronous mode.
- Cannot select the device ID and the 10-bit slave address for wake-up interrupt factor. Set the DVIDE bit in SVCTL and SDADLS bit in SDATBASn (n = 0 to 2) to 0.
- Sets all bits in BIE (TENDIE, NACKDIE, SPCNDDIE, STCNDDIE, ALIE, TODIE) and TDBEIE0 and RDBFIE0 bits in NTIE to 0 (interrupt disabled) before switching the asynchronous operation.
- Do not use the timeout function while the Wake-up function is enabled (WUCTL.WUFE = 1).
- Wake-up interrupt is generated while PCLK/TCLK asynchronous operation (when WUST.WUASYNF = 1). For detecting slave address matching, detection of slave address match in PCLK/TCLK synchronous mode (WUST.WUASYNF = 0) does not generate Wake-up interrupt, and BST.WUCNDDF flag is not set.
- If WUCTL.WUFSYNE bit to 0 write timing and START condition of detecting a conflict, I3C might start the next reception in PCLK/TCLK synchronous operation mode. In this case, WUST.WUASYNF flag becomes 1 (switch to PCLK/TCLK asynchronous mode) when data communication is finished and detected STOP condition and starts the Wake-up event detection.
- If you want to switch from PCLK/TCLK asynchronous operation to PCLK/TCLK synchronous operation without address match detection, it will switch in the STOP condition detection. When the WUCTL.WUFSYNE bit was set to 1 in a bus free state, it is continued PCLK/TCLK asynchronous operation (reception operation: waiting communication frame). WUST.WUASYNF flag becomes to 0 when I3C detect the STOP condition of the next communication frame, and I3C switches to PCLK/TCLK synchronous operation.
- After writing 0 to WUFSYNE bit in WUCTL, do not change I3C operation mode setting register (BFCTL, SCSTRCTL, ACKCTL, INCTL, SVCTL, SDATBASn (n = 0 to 2)) until switched to the PCLK/TCLK asynchronous operation from PCLK/TCLK synchronous operation (while WUST.WUASYNF flag = 1). If register value changes such as by

the interrupt processing in this period, I3C might malfunction without succeeding to the setting to the asynchronous operation.

- During PCLK/TCLK asynchronous operation (WUST.WUASYNF = 1), do not refer to each flag of SVST, BST, NTST, HTST register and BCST.BFREF flag.
- Do not set ACKCTL.ACKT = 1 in order to make an ACK response in the synchronization unit when Wake-up is performed by slave address match in Normal wake-up mode 2.

33.3.2.5.2 Wake Up function

(1) I3C Master Wake-Up

Wake-up interrupt causes of I3C master are shown below.

- SDA low detection (IBI request from I3C slave)

The operation when transitioning to active mode (normal operation) by Wake-Up interrupt of SDA Low detection is shown below.

Before wake-up recovery:	SDA Low Drive is detected and the I3C_WU interrupt is asserted.
During wake-up recovery:	Keep I3C_SCL Line High.
After wake-up recovery:	Drive SCL Low and complete START condition. SCL is supplied on the I3C bus and IBI from I3C slave is received.

If transition to active mode (normal operation) due to other factors, disable the Wake-up function as necessary.

After confirming PRSTDBG.SDILV = 1, set WUCTL.WUFE = 0.

Do not use the timeout function while the Wake-up function is enabled (WUCTL.WUFE = 1).

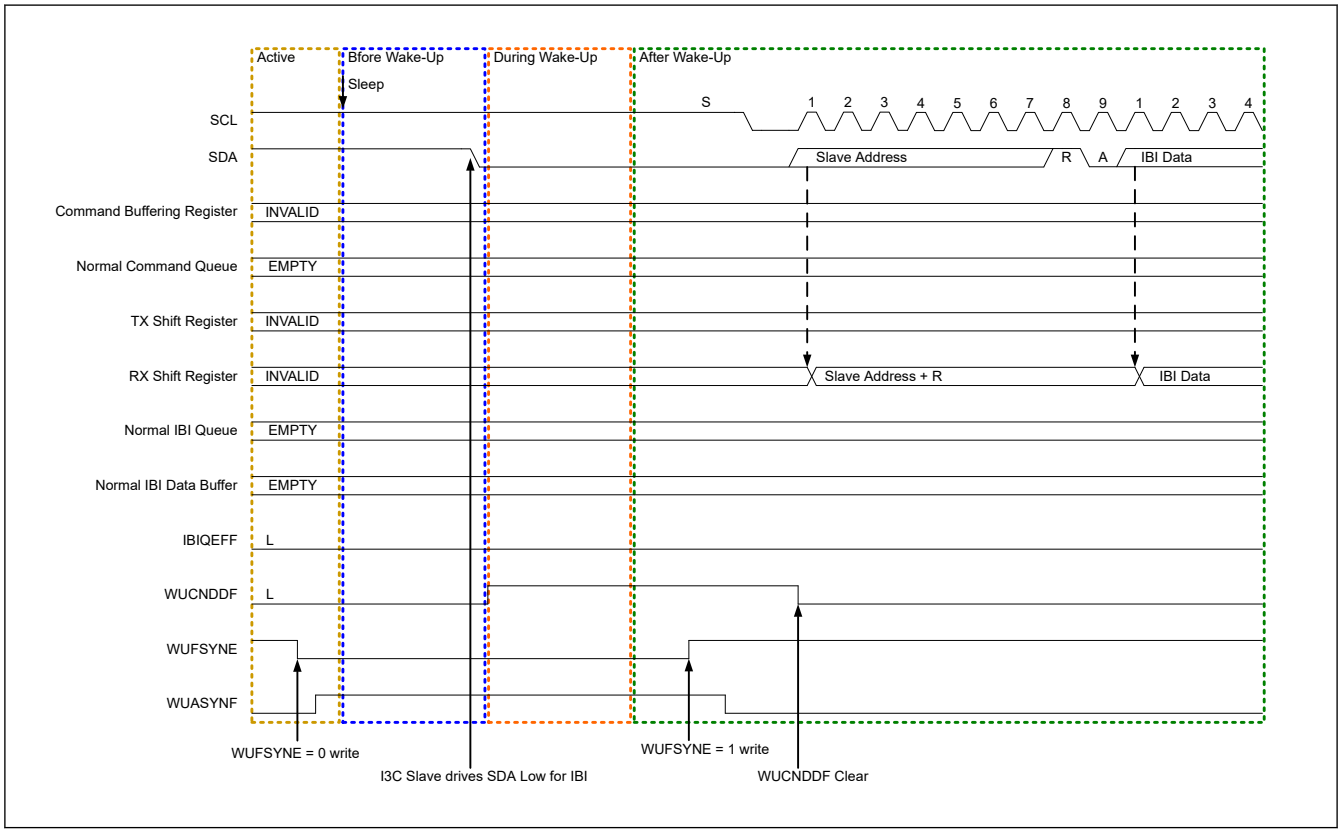


Figure 33.134 I3C master wake-up operation

(2) I3C Slave Wake-Up

Wake-up interrupt causes of I3C slave are shown below.

- Broadcast address (0x7E) and detects its own Slave address match

The operation when the Broadcast address (0x7E) and transition to active mode (normal operation) by Wake-up interrupt by detecting its own Slave address match is shown.

Before wake-up recovery:	1. If I3C detects BA (0x7E/W) following a START (or Repeated START) Condition, then I3C shall generates ACK (after 0x7E/W). 2. If I3C detects its own Dynamic address after a Repeated START condition following Step1, then I3C shall generates NACK (after its own Dynamic address) and then issues a I3C_WU interrupt.
During wake-up recovery:	I3C always generates NACK.
After wake-up recovery:	Normal operation continues.

If transition to active mode (normal operation) due to other factors, disable the Wake-up function as necessary. Do not use the timeout function while the Wake-up function is enabled (WUCTL.WUFE = 1).

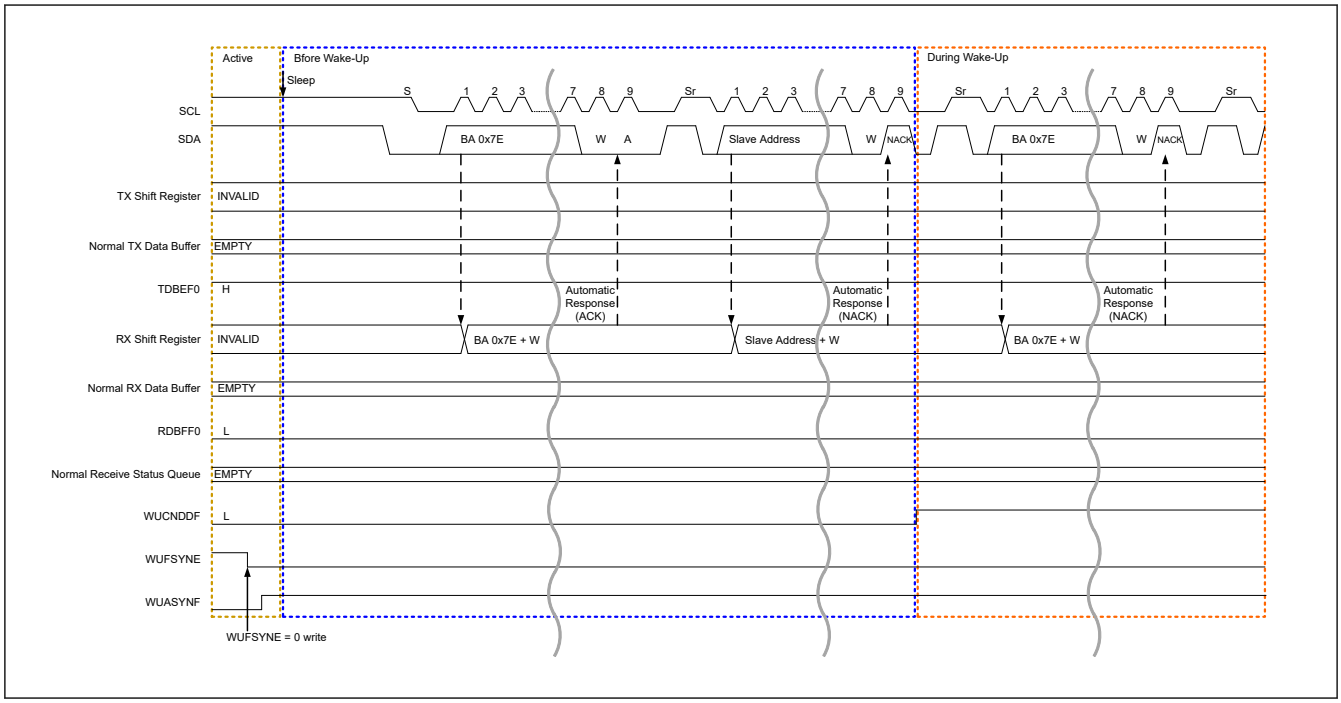


Figure 33.135 I3C slave wake-up operation (1/2)

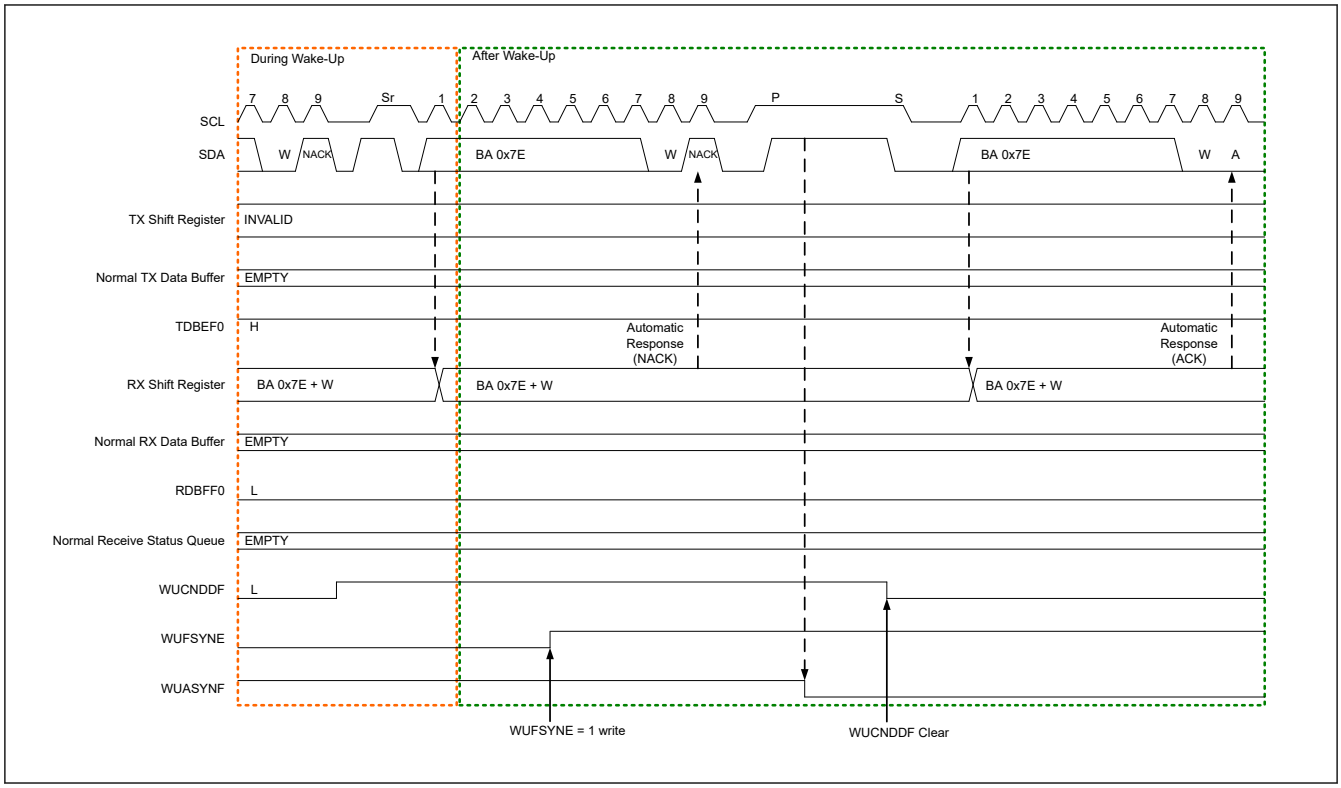


Figure 33.136 I3C slave wake-up operation (2/2)

33.3.2.6 Other

33.3.2.6.1 SCL Synchronization Circuit [I²C mode]

This function is enabled while the PRTS.PRTMD bit is set to 1.

In generation of the SCL clock, I3C starts counting out the value for width at high level specified in STDBR.SBRHO[7:0] when it detects a rising edge on the I3C_SCL line and drives the I3C_SCL line low once counting of the width at high level is complete.

When I3C detects the falling edge of the I3C_SCL line, it starts counting out the width at low level period specified in STDBR.SBRLO[7:0], and then stops driving the I3C_SCL line (releases the line) once counting of the width at low level is complete. The SCL clock is thus generated.

If multiple master devices are connected to the I²C bus, a collision of SCL signals may arise due to contention with another master device. In such cases, the master devices have to synchronize their SCL signals. Since this synchronization of SCL signals must be bit by bit, I3C is equipped with a facility (the SCL synchronization circuit) to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the I3C_SCL line while in master mode.

When I3C has detected a rising edge on the I3C_SCL line and thus started counting out the width at high level specified in STDBR.SBRHO[7:0], and the level on the I3C_SCL line falls because an SCL signal is being generated by another master device, I3C stops counting when it detects the falling edge, drives the level on the I3C_SCL line low, and starts counting out the width at low level specified in STDBR.SBRLO[7:0]. When I3C finishes counting out the width at low level, it stops driving the I3C_SCL line to the low level (releases the line). At this time, if the width at low level of the SCL clock signal from the other master device is longer than the width at low level set in this module, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the I3C_SCL line has been released. When I3C finishes outputting the low-level period of the SCL clock, the I3C_SCL line is released and the SCL clock rises. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the SCSYNE bit in BFCTL is set to 1.

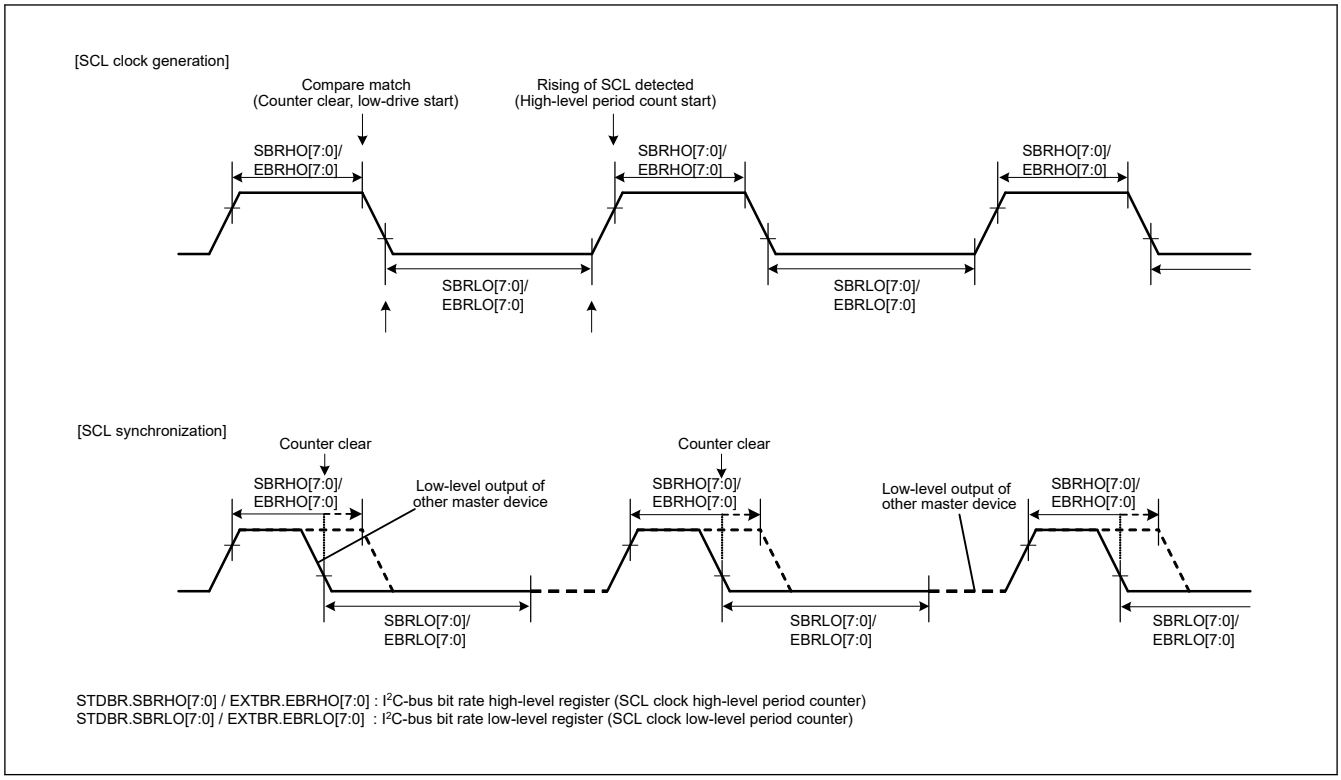


Figure 33.137 Generation and synchronization of the SCL signal

33.3.2.6.2 Facility for Delaying SDA Output [I²C mode]

I3C module incorporates a facility for delaying output on the I3C_SDA line. The delay can be applied to all output (issuing of the START, Repeated START, and STOP conditions, data, and the ACK and NACK signals) on the I3C_SDA line.

With the SDA output delay facility, SDA output is delayed from detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval over which the SCL clock is at the low level. Doing this leads to usage with the aim of preventing erroneous operation of communications devices, with the aim of satisfying the 300 ns (minimum) data-hold time requirement of the SMBus specification.

The output delay facility is enabled by setting the SDOD[2:0] bits in OUTCTL to any value other than 000b, and disabled by setting the same bits to 000b.

While the SDA output delay facility is enabled (while the SDOD[2:0] bits in OUTCTL are set to any value other than 000b), the SDODCS bit in OUTCTL selects the clock source for counting by the SDA output delay counter as the internal base clock (I3Cφ) for I3C module or as a clock signal derived by dividing the frequency of the internal base clock by two (I3Cφ/2). The counter counts the number of cycles set in the SDOD[2:0] bits in OUTCTL. After counting of the set number of cycles of delay is completed, I3C module places the required output (START, Repeated START, or STOP condition, data, or an ACK or NACK signal) on the I3C_SDA line.

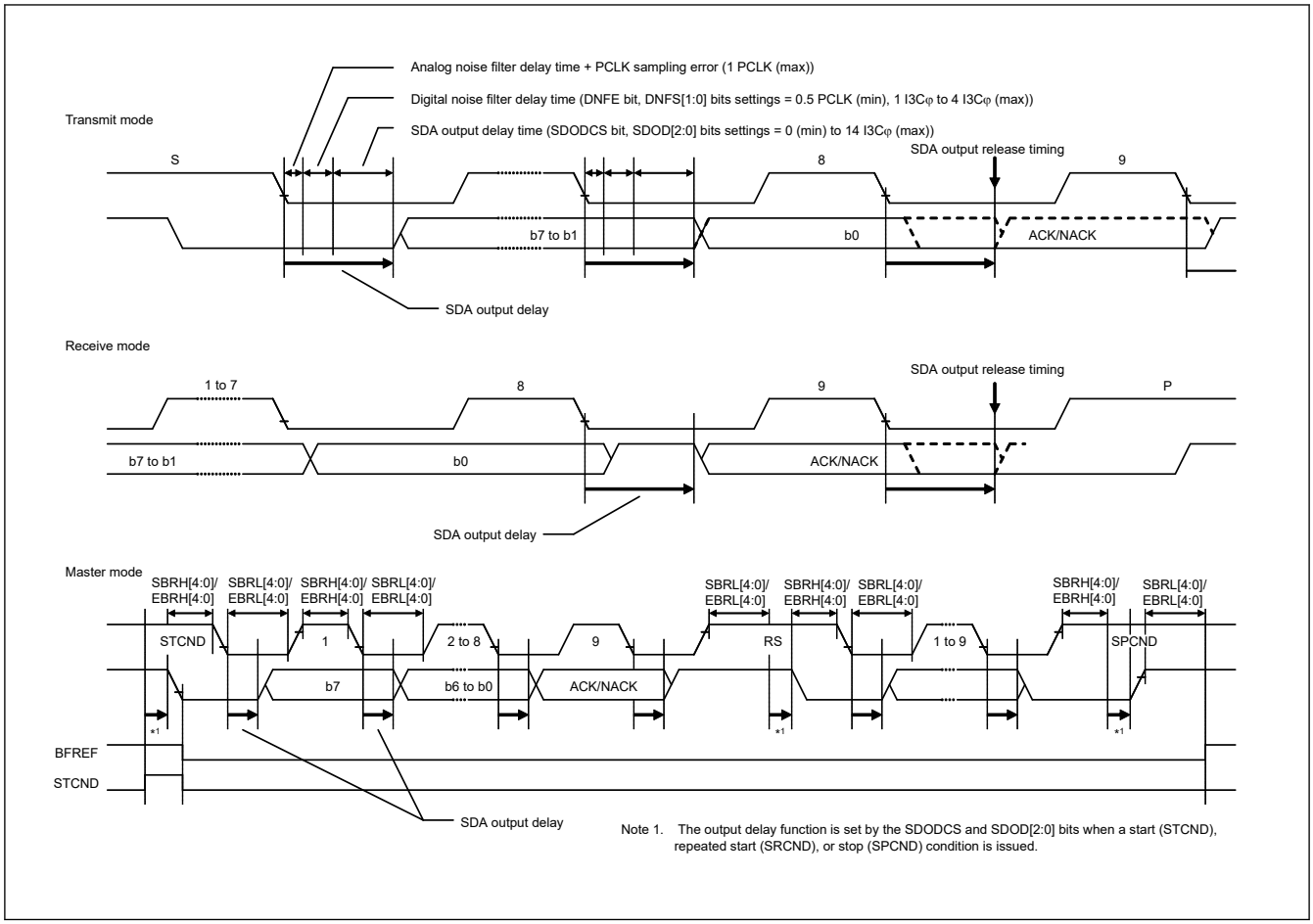


Figure 33.138 SDA output delay facility

33.3.2.6.3 Digital Noise-Filter Circuits [I^2C mode]

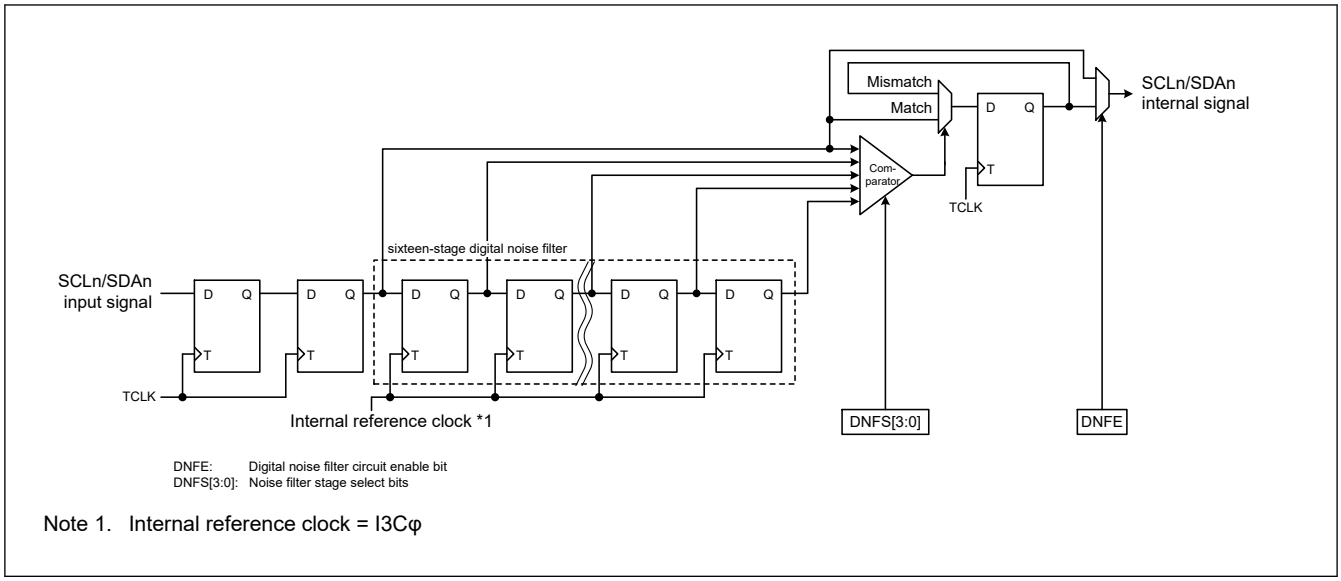
The states of the I3C_SCL and I3C_SDA pins are conveyed to the internal circuitry through digital noise-filter circuits. Figure 33.139 is a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of I3C consists of 16 flip-flop circuit stages connected in series and a match detection circuit. When HS mode is selected, only the first four flip-flop circuits stages are enabled.

The number of effective stages in the digital noise filter is selected by the INCTL.DNFS[3:0] bits. The selected number of effective stages determines the noise-filtering capability as a period from one to sixteen I3Cφ cycles.

The input signal to the I3C_SCL pin (or I3C_SDA pin) is sampled on rising edges of the I3Cφ signal. When the input signal level matches the output level of the number of effective flip-flop circuit stages as selected by the INCTL.DNFS[3:0] bits, the signal level is conveyed to the subsequent stage. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (TCLK) and the transfer rate is small (For example, data transfer at 400 kbps with TCLK = 4 MHz), the characteristics of the digital noise filter may lead to the elimination of needed signals as noise.



Note 1. Internal reference clock = I3Cφ

Figure 33.139 Block diagram of digital noise filter circuit

33.3.3 Operation

33.3.3.1 Initial Setting Flow

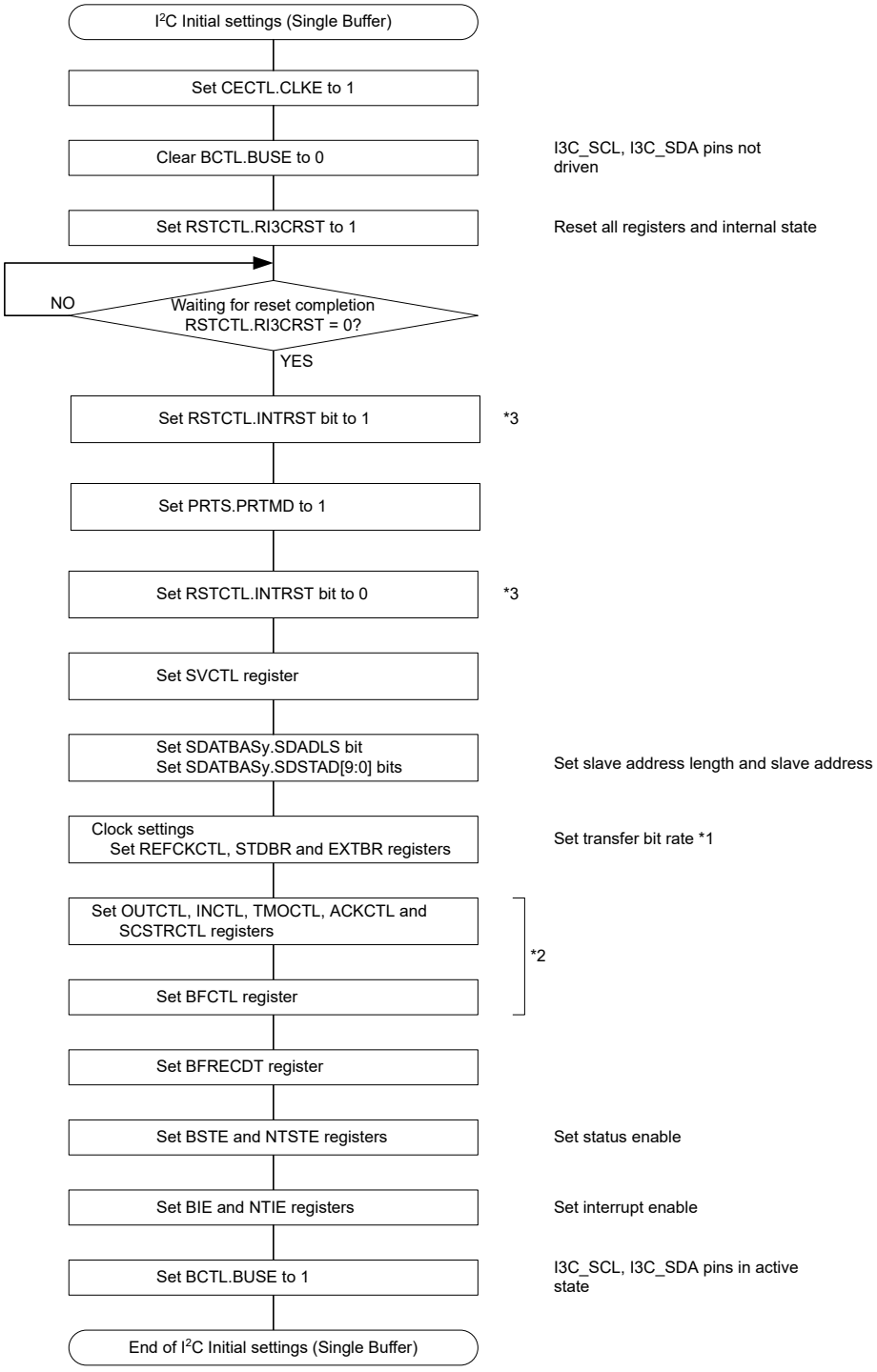
33.3.3.1.1 I2C Initial Setting Flow (Single Buffer Transfer)

Before starting data transmission and reception in I2C protocol mode, initialize I3C module.

First, set the CECTL.CLKE to 1, and set the BCTL.BUSE bit to 0 (I3C_SCL, I3C_SDA pins not driven).

Next, Set the RSTCTL.RI3CRST bit to 1. This initialize the registers and internal state of I3C module. Then, it waits for RI3CRST to become "0". For flags and registers to be initialized by this operation, see section 33.6. Reset Descriptions.

After that, set each registers required for operation. For detail procedure, see Figure 33.140.



Note: y = 0 to 2
Note 1. When the I3C is used only in slave mode, set the STDBR register to a value longer than the data setup time.
Note 2. Set these registers as necessary.
Note 3. This step can be omitted if certain conditions are met. For details, see section 33.7.4. Notes on Rewriting the PRTS.PRTMD Bit.

Figure 33.140 Example of I²C initialization flowchart (single buffer transfer)

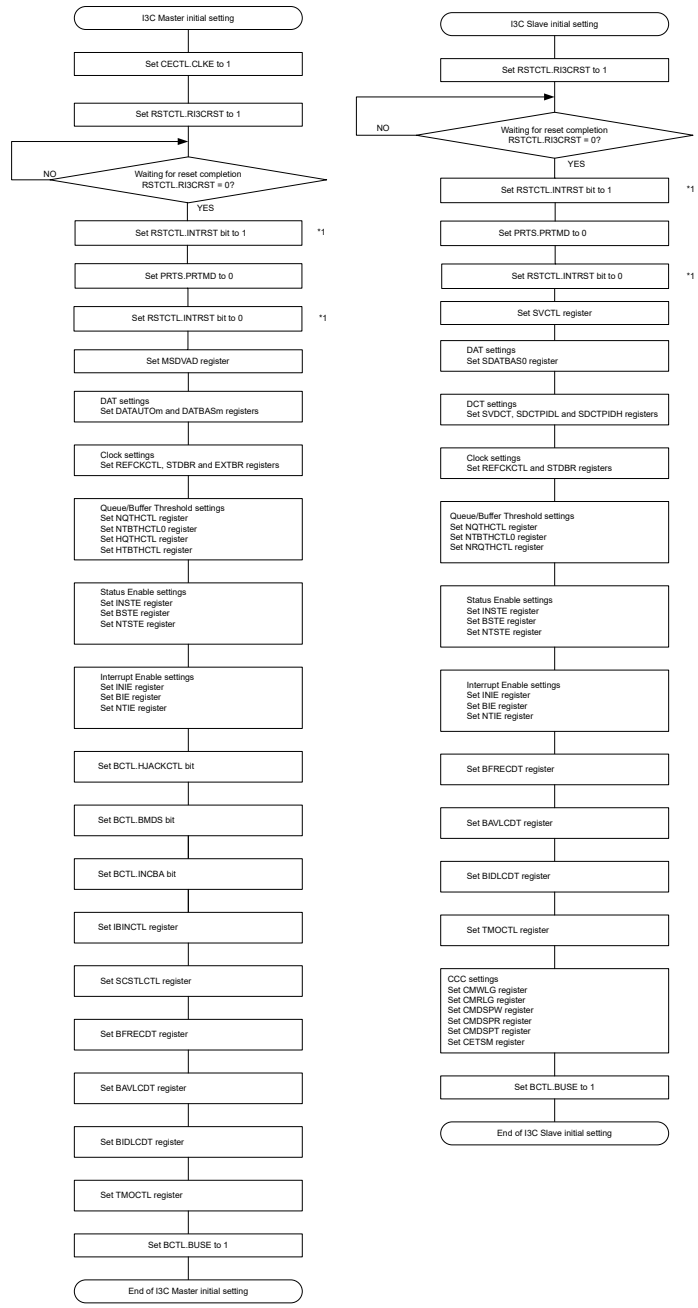
33.3.3.1.2 I3C Initial Setting Flow

Before starting data transmission and reception in I3C protocol mode, initialize I3C module.

First, set the CECTL.CLKE to 1.

Next, Set the RSTCTL.RI3CRST bit to 1. This initialize the registers and internal state of I3C module. Then, it waits for RI3CRST to become 0. For flags and registers to be initialized by this operation, see section 33.6. Reset Descriptions.

After that, set each registers required for operation. For detail procedure on each of the master mode and slave mode, see Figure 33.141.



Note 1. This step can be omitted if certain conditions are met. For details, see section 33.7.4. Notes on Rewriting the PRTS.PRTMD Bit.

Figure 33.141 Example of each initialization flowchart for I3C master mode and slave mode

33.3.3.2 I3C Communication Flow

Figure 33.142 illustrates how I3C communication is initiated:

- All I3C communication occurs within a frame. The frame begins with a START, followed by one or more transfers, and a STOP.

- For the HDR modes:
  - First the dedicated Broadcast I3C address (0x7E) is issued to all Slaves on the I3C bus.
  - Then one of the EnterHDR CCCs is issued, indicating that the Master is entering an HDR mode. Each HDR mode has its own EnterHDR CCC.
  - This is followed by one or more HDR transfers.
  - HDR mode is ended by using the HDR exit pattern protocol.

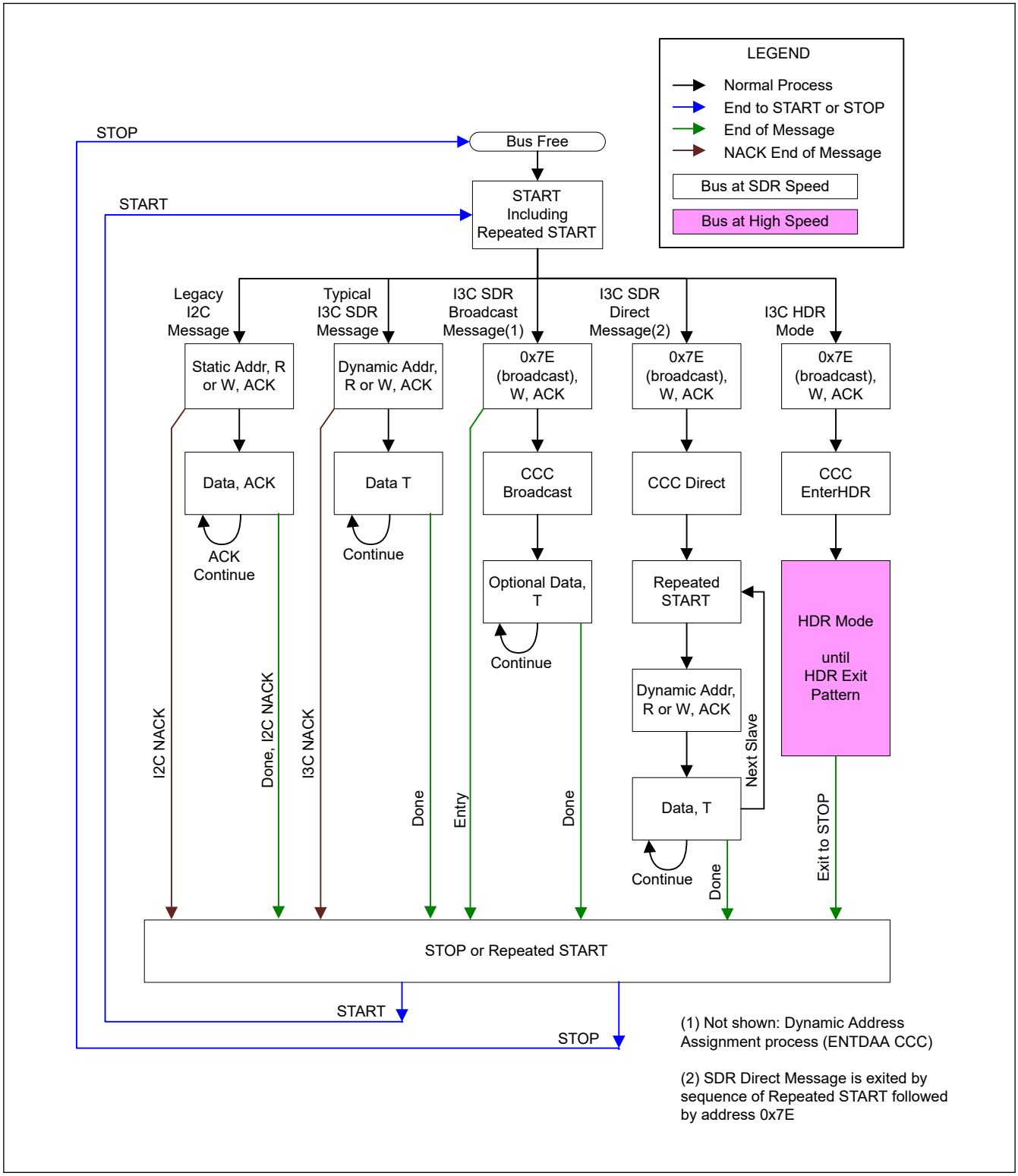


Figure 33.142 I3C communication flow

I3C is based on a frame encapsulation approach. A frame includes a data payload. The transfer protocol for the data payload is either SDR or HDR. Frames are bordered by I^2C-like bus management.

The I3C frame always includes at least the START, the Header, the Data, and the STOP. The Header following a START allows for Bus Arbitration. The Master uses the Header to address Slave device (s). Slave devices (s) may use the Header Arbitration for multiple purposes: for In-Band Interrupt, and for Secondary Master functionality.

Common Command Codes (CCCs) are used to enter the High Data Rate (HDR) modes. It is important to understand that I3C bus activity for the HDR Message does not follow the Legacy I^2C format.

I3C allows only one Master to have control of the I3C bus at a time. Mechanisms for handoff of the Master role from one device to another device are provided.

33.3.3.3 Master Mode Communication Flow

33.3.3.3.1 I²C Master Transmission Flow (Single Buffer Transfer)

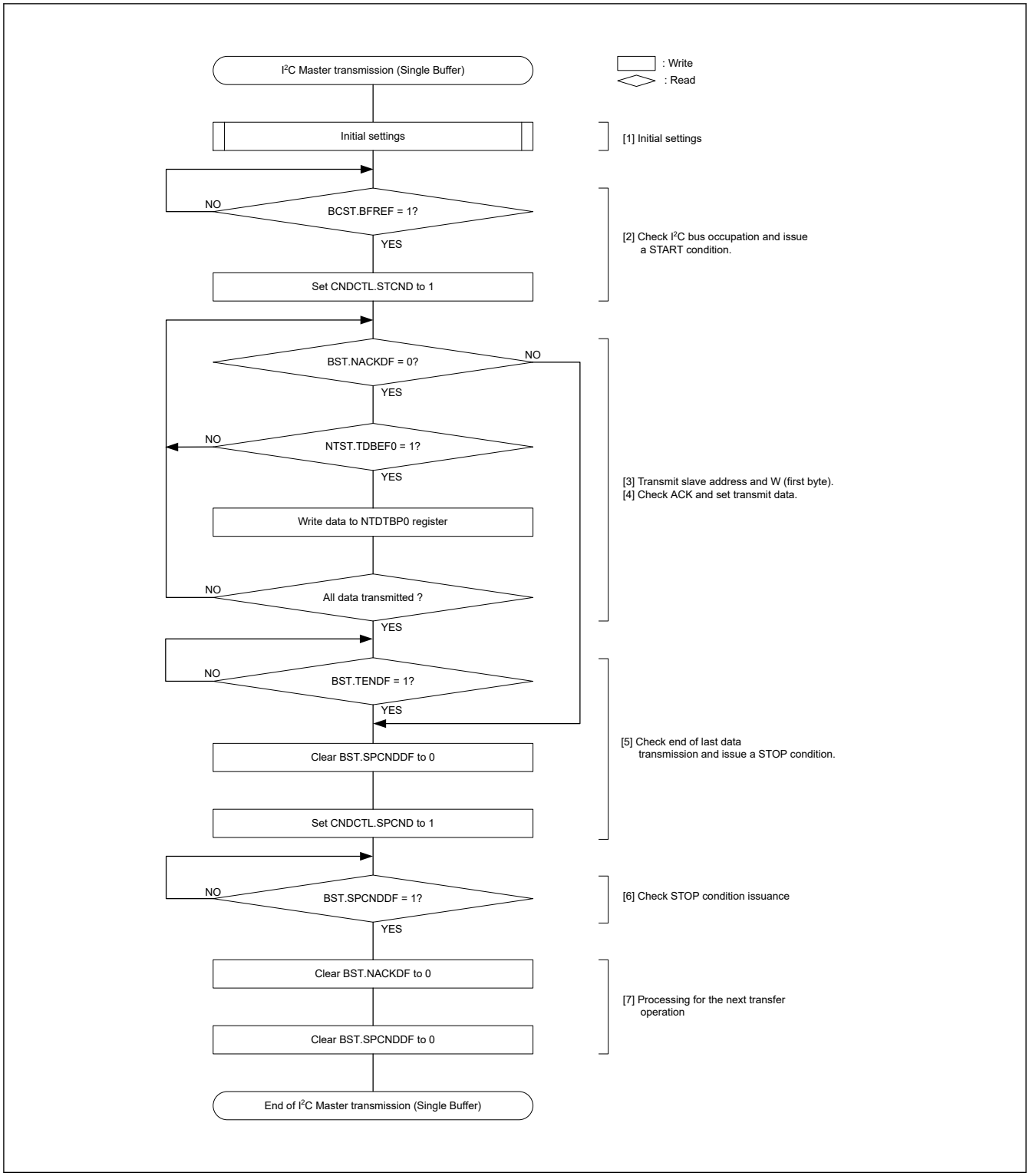


Figure 33.143 Example of I²C master transmission flowchart (single buffer transfer)

33.3.3.3.2 I²C Master Reception Flow (Single Buffer Transfer)

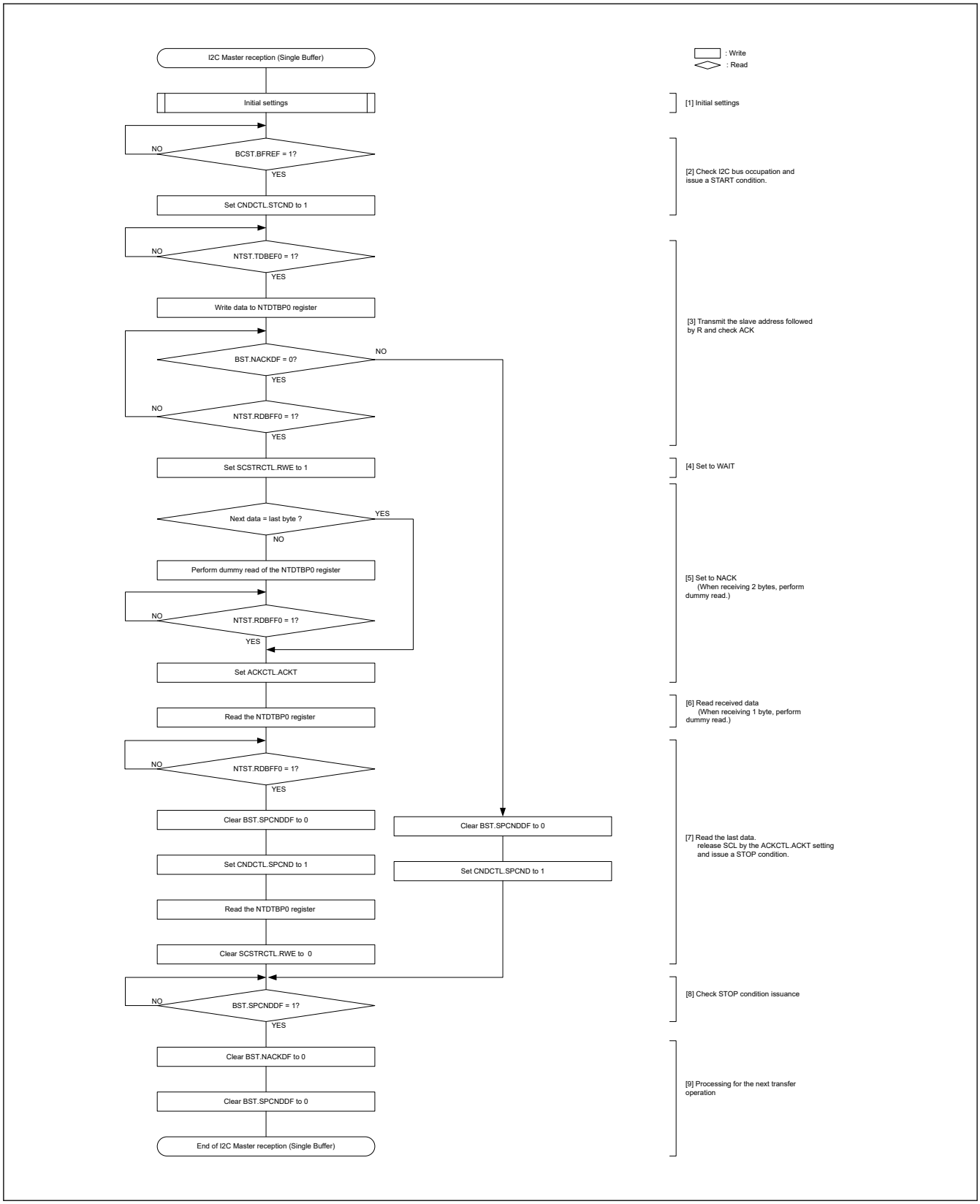


Figure 33.144 Example of I²C master reception flowchart (7-bit address format, 1 or 2 bytes)

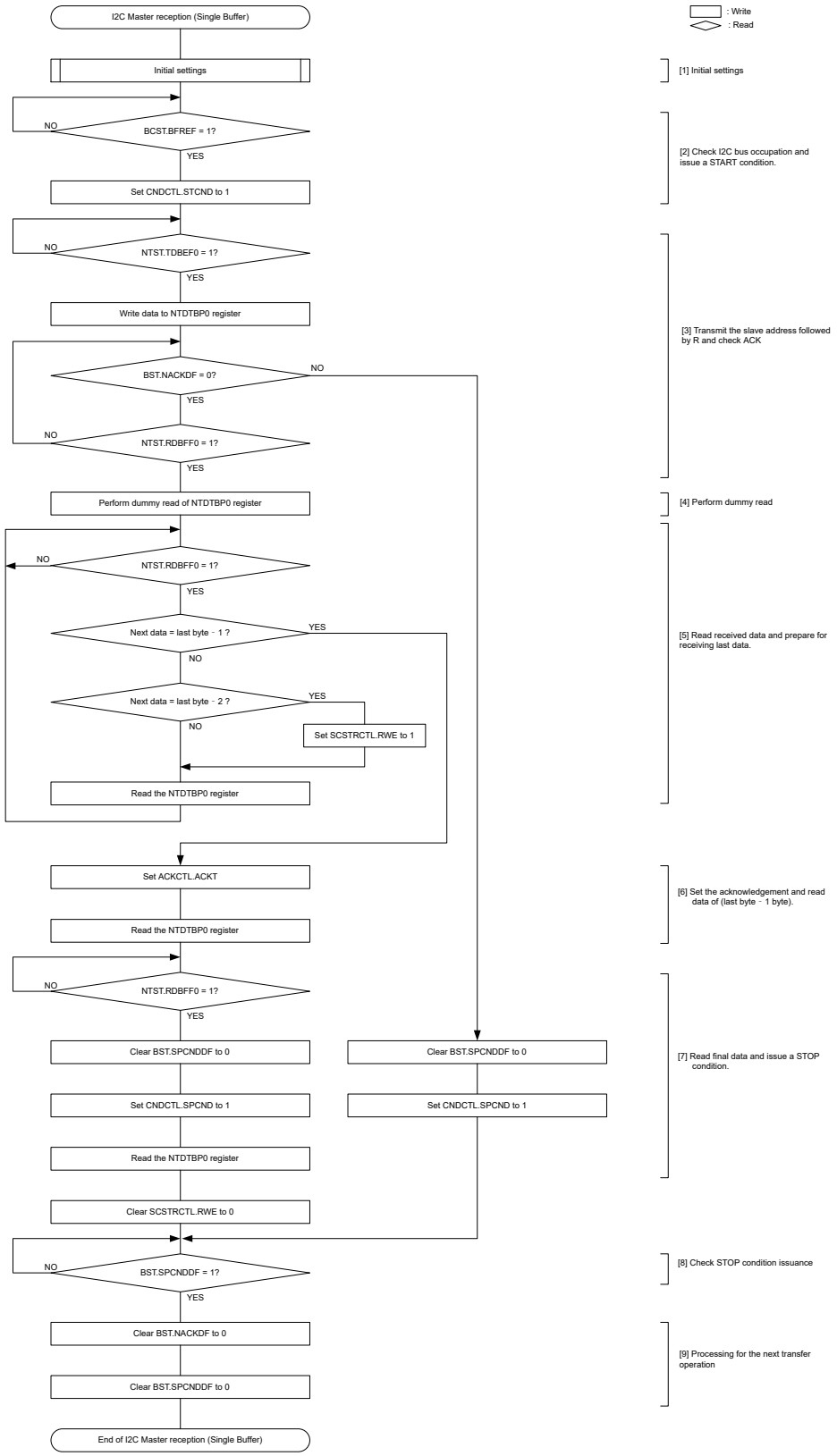
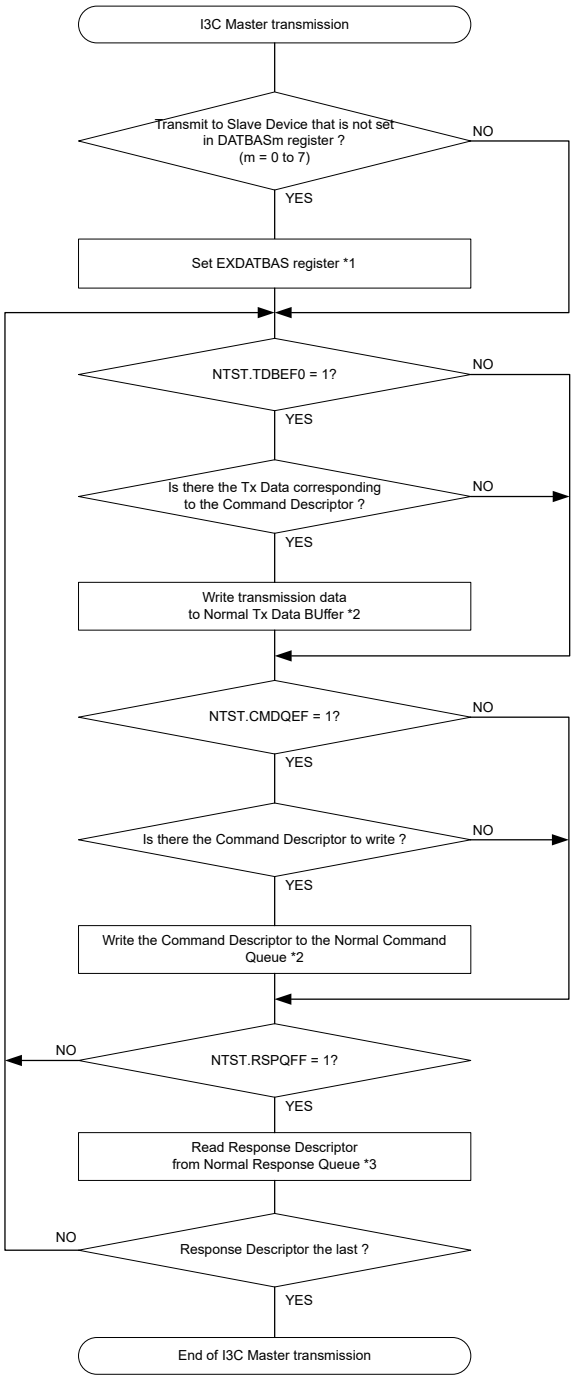


Figure 33.145 Example of I²C master reception flowchart (7-bit address format, 3 bytes or more)

33.3.3.3.3 I3C Master Transmission Flow (Normal FIFO Buffer Transfer)

Master transmission flow in I3C normal FIFO buffer transfer is common to Legacy I^2C and SDR (Private Transfer, Broadcast CCC, Direct CCC), HDR-DDR, and HDR-TSP/TSL.

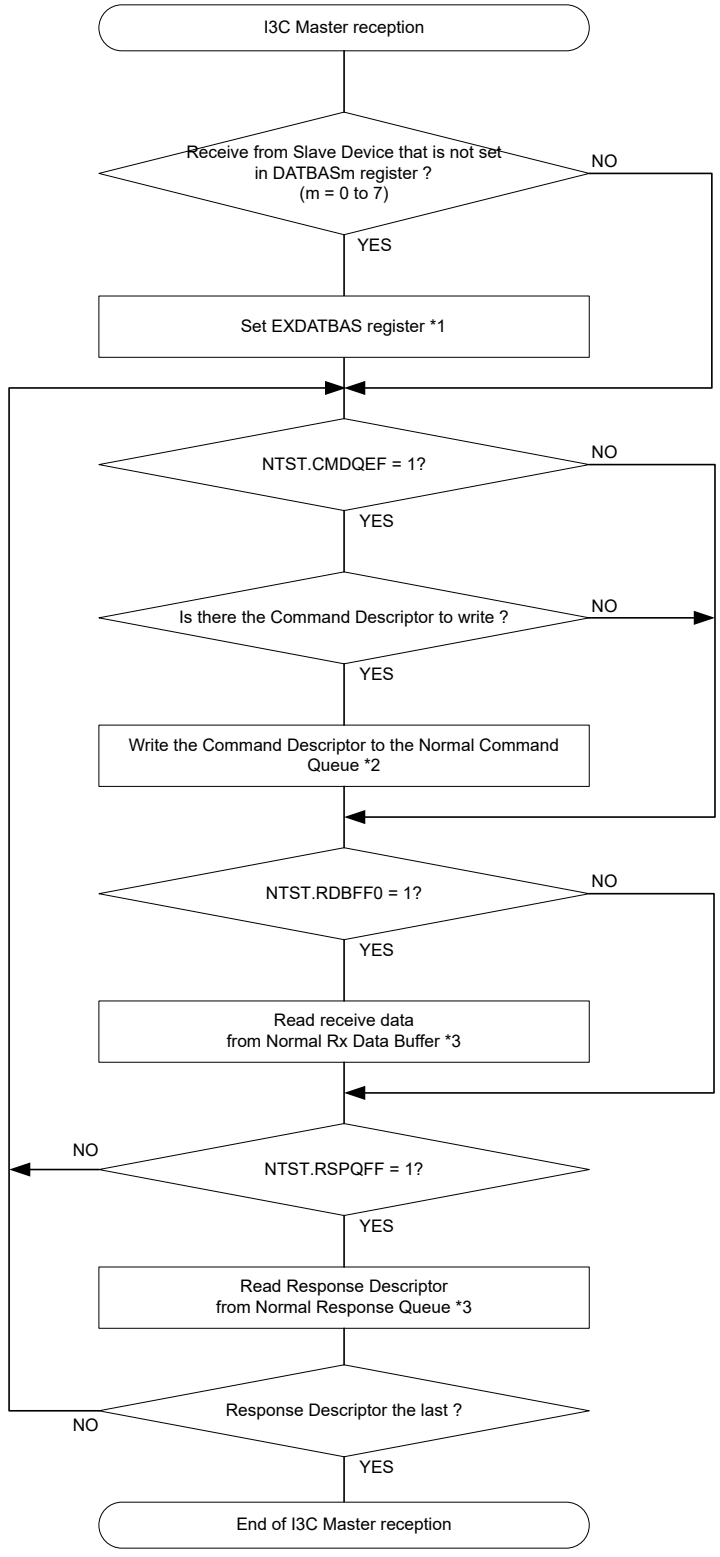


Note 1. Changing the EXDATBAS register is prohibited until all response descriptors of command for the Slave device set in the EXDATBAS register are stored.
Note 2. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 3. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 33.146 Example of I3C master transmission flowchart (normal FIFO buffer transfer)

33.3.3.3.4 I3C Master Reception Flow (Normal FIFO Buffer Transfer)

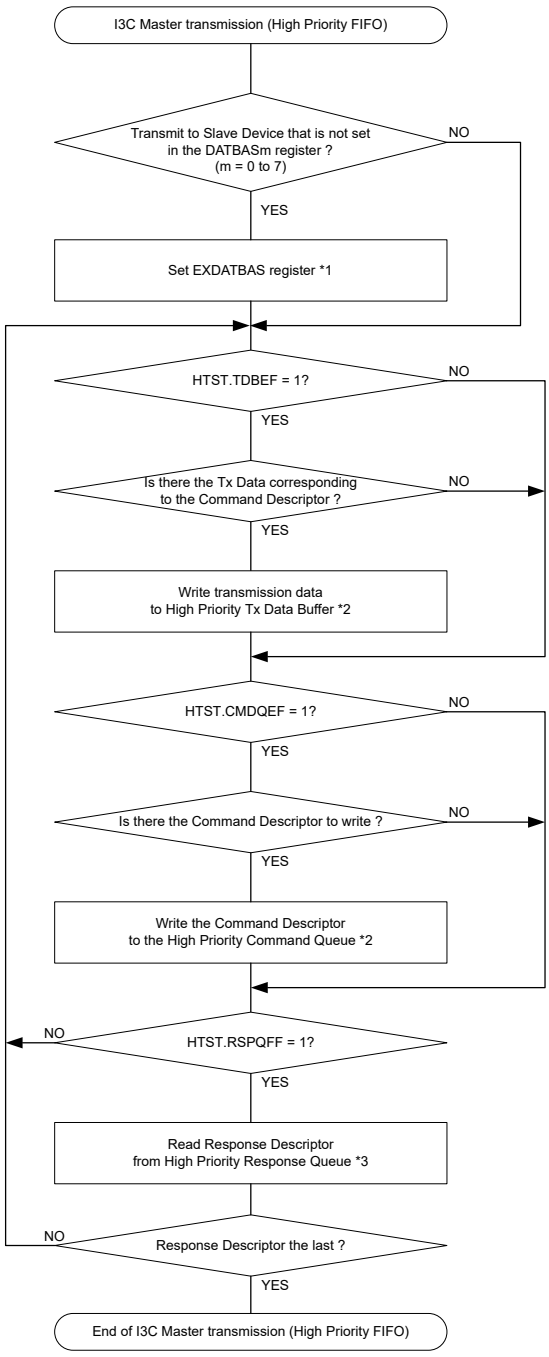
Master reception flow in I3C normal FIFO buffer transfer is common to Legacy I^2C and SDR (Private Transfer, Broadcast CCC, Direct CCC), HDR-DDR, and HDR-TSP/TSL.



Note 1. Changing the EXDATBAS register is prohibited until all response descriptors of command for the Slave device set in the EXDATBAS register are stored.
Note 2. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 3. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 33.147 Example of I3C master reception flowchart (normal FIFO buffer transfer)

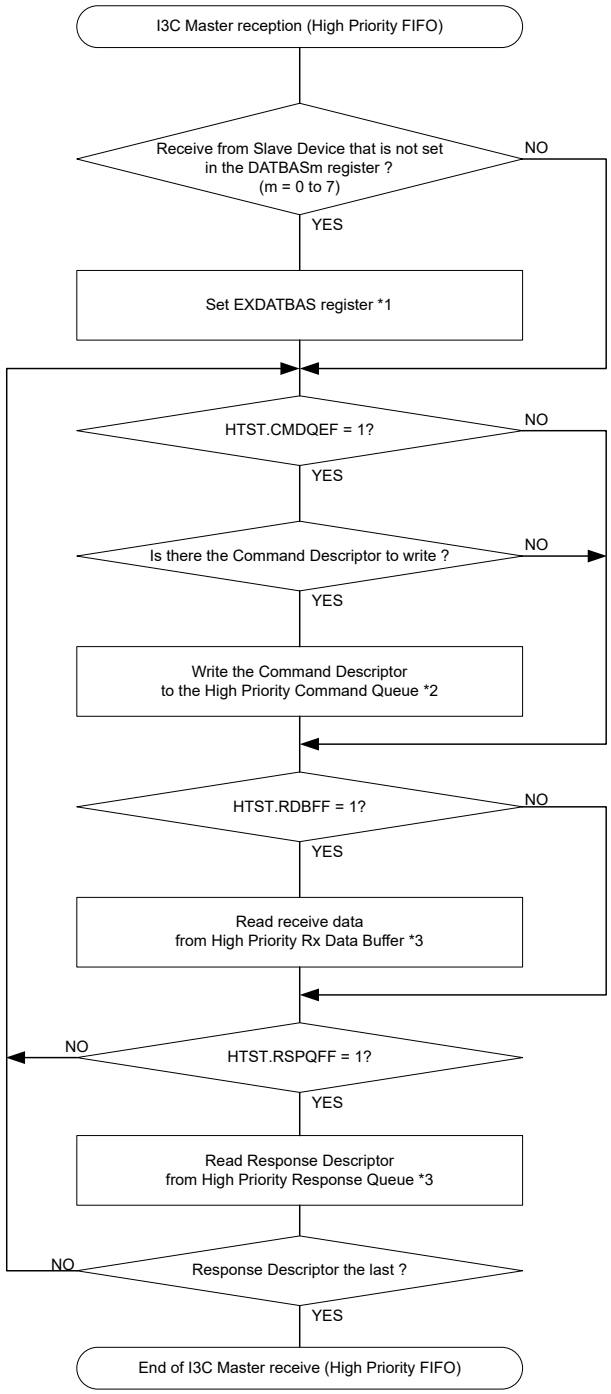
33.3.3.3.5 I3C Master Transmission Flow (High Priority FIFO Buffer Transfer)



Note 1. Changing the EXDATBAS register is prohibited until all response descriptors of command for the Slave device set in the EXDATBAS register are stored.
Note 2. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 3. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 33.148 Example of I3C master transmission flowchart (high priority FIFO buffer transfer)

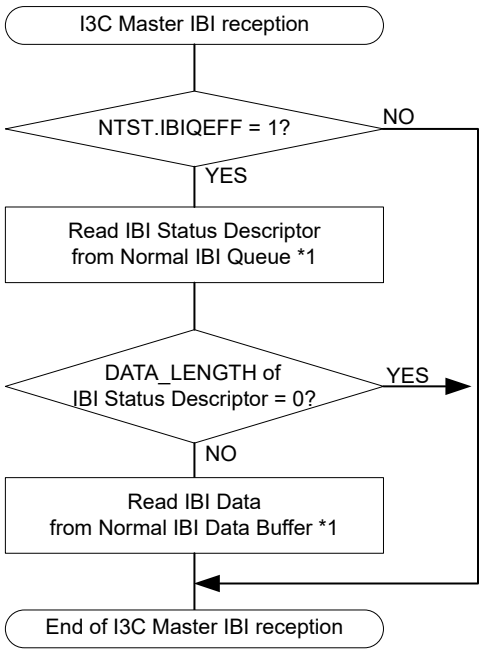
33.3.3.3.6 I3C Master Reception Flow (High Priority FIFO Buffer Transfer)



Note 1. Changing the EXDATBAS register is prohibited until all response descriptors of command for the Slave device set in the EXDATBAS register are stored.
Note 2. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 3. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 33.149 Example of I3C master reception flowchart (high priority FIFO buffer transfer)

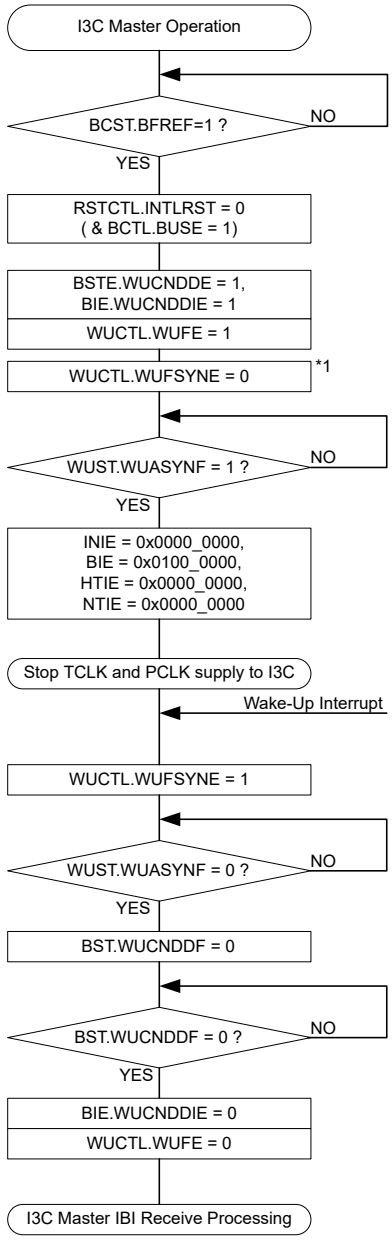
33.3.3.3.7 I3C Master IBI Reception Flow



Note 1. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 33.150 Example of I3C master IBI reception flowchart

33.3.3.3.8 I3C Master Wake-Up Flow



[1] Wait I3C bus free and I3C stay in the standby state.

[2] Negate internal reset if assert the internal reset.

[3] Wake-Up Interrupt is enabled.

[4] Wake-Up function is enabled.

[5] I3C of operating state is changed from PCLK/TCLK synchronous to PCLK/TCLK asynchronous.

[6] Interrupt requests are disabled except WUI.

[7] Stop TCLK supply to I3C, then stop PCLK supply to I3C.
(I3C will continue to receive)

[8] Start TCLK and PCLK supply to I3C by Wake-Up interrupt, then start PCLK supply to I3C.

[9] I3C of operating state is changed from PCLK/TCLK asynchronous to PCLK/TCLK synchronous.

[10] Write 0 to BST.WUCNDDF.
(Read and check whether clearing has been completed befor return from interrupt handling.)

[11] Wake-Up interrupt is disabled.

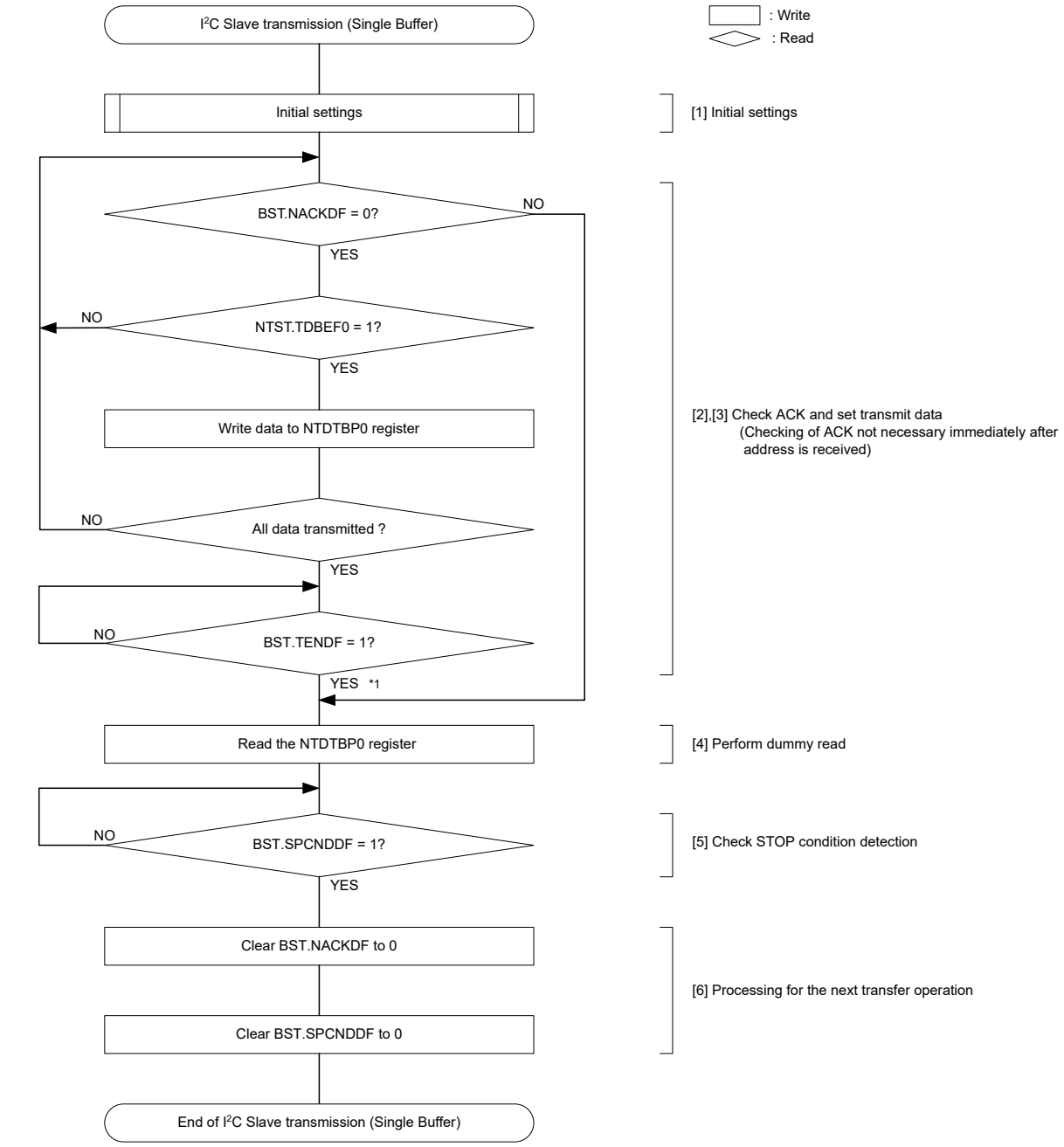
[12] Wake-Up function is disabled.

Note 1. After switching the I3C_SDA pin to the general I/O port (Input) in the I/O port function setting, set WUCTL.WUFSYNE to 1. Then wait until WUST.WUASYNF = 1, and switch the I3C_SDA pin to the peripheral function in the I/O port function setting.

Figure 33.151 Use case of I3C master wake-up

33.3.3.4 Slave Mode Communication Flow

33.3.3.4.1 I²C Slave Transmission Flow (Single Buffer Transfer)



Note 1. Normally, Master responds to the last byte with NACK, so when BST.TENDF=1, BST.NACKDF will also 1. If BST.NACKDF is 0 after BST.TENDF=1, the master may be hang up, so reset I3C with RSTCTL.INTLRST.

Figure 33.152 Example of I²C slave transmission flowchart (single buffer transfer)

33.3.3.4.2 I²C Slave Reception Flow (Single Buffer Transfer)

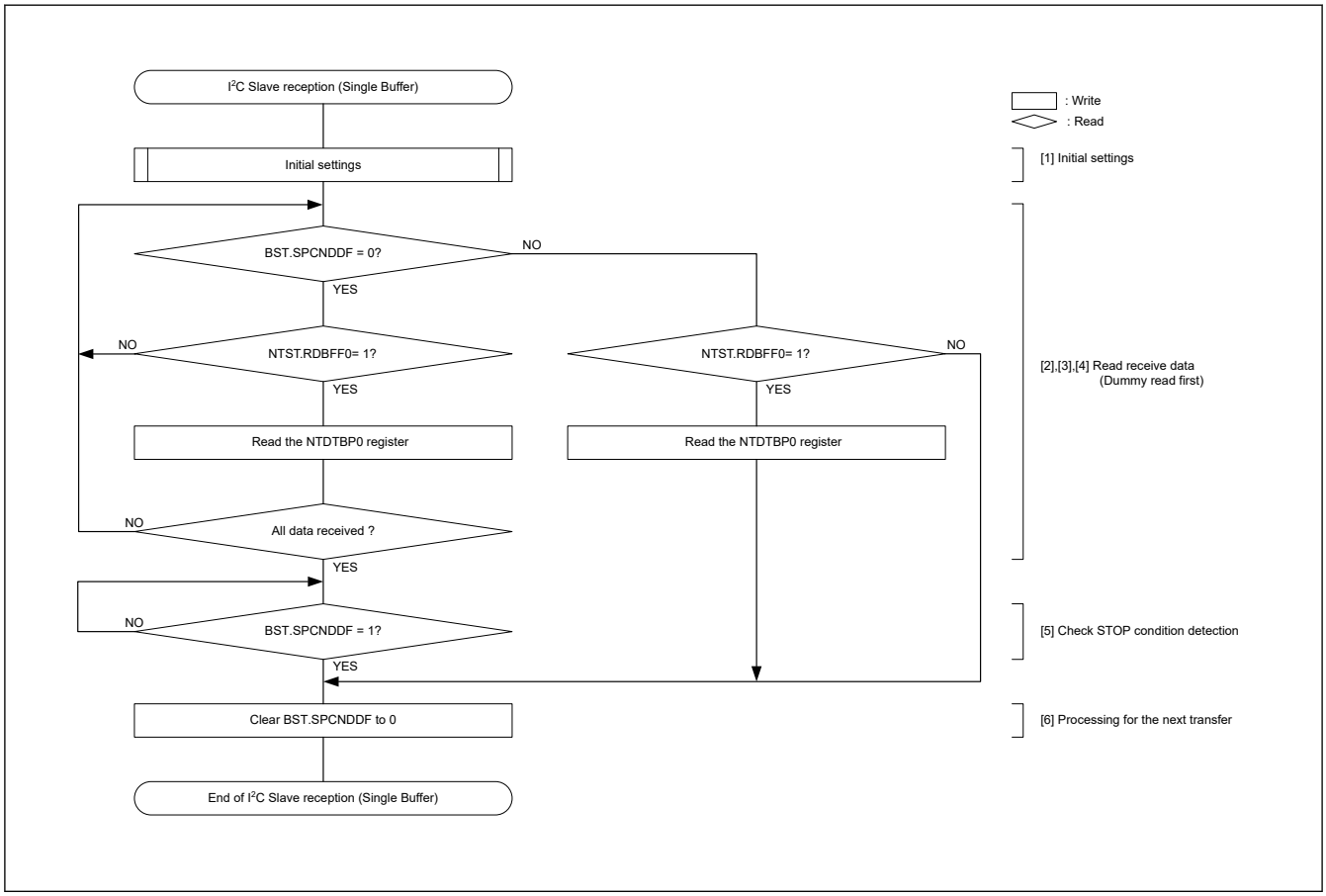
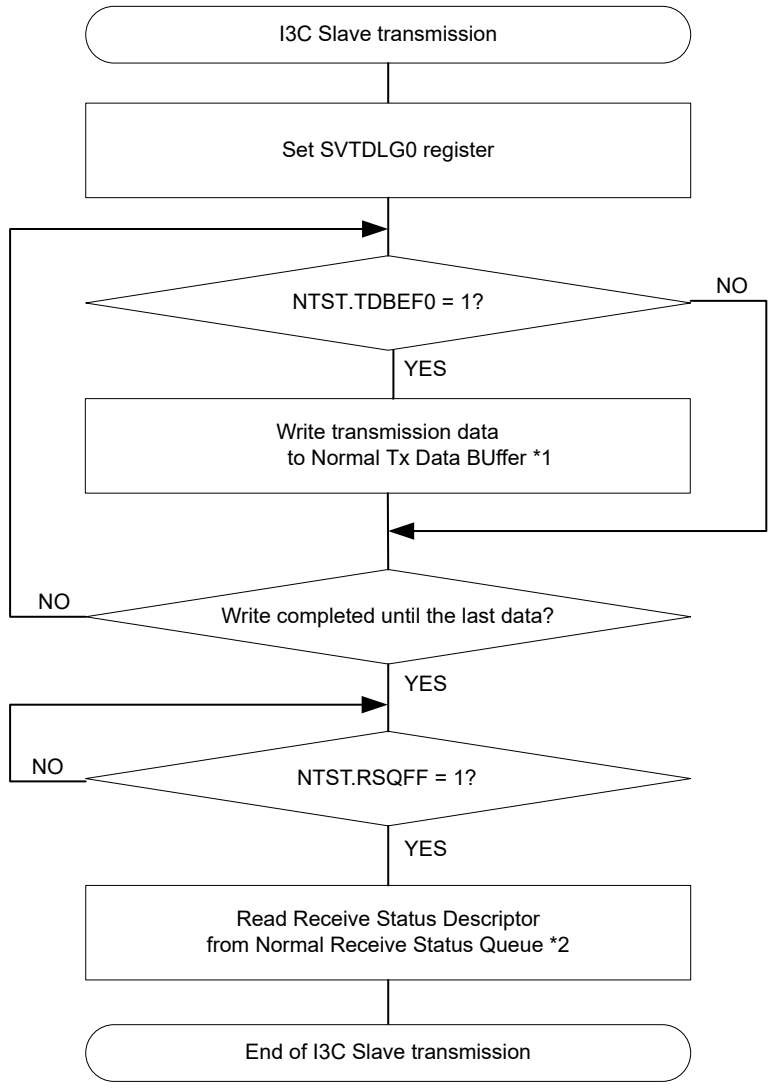


Figure 33.153 Example of I²C slave reception flowchart (single buffer transfer)

33.3.3.4.3 I3C Slave Transmission Flow (Normal FIFO Buffer Transfer)

Slave Transmission Flow in I3C normal FIFO buffer transfer is common to Legacy I²C, SDR (Private Transfer, Broadcast CCC, Direct CCC), HDR-DDR, and HDR-TSP/TSL.

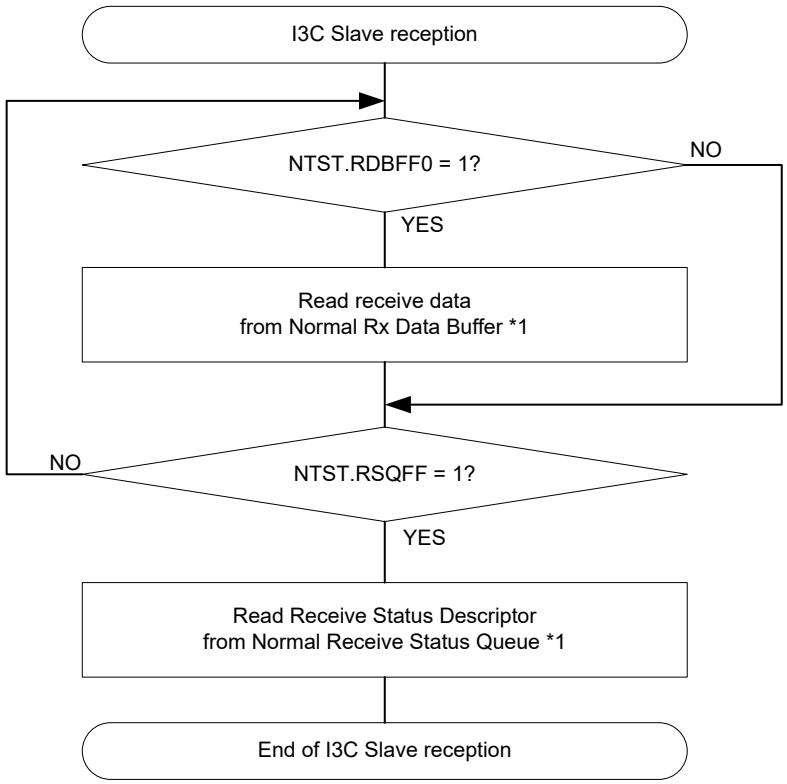


Note 1. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 2. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 33.154 Example of I3C slave transmission flowchart (normal FIFO buffer transfer)

33.3.3.4.4 I3C Slave Reception Flow (Normal FIFO Buffer Transfer)

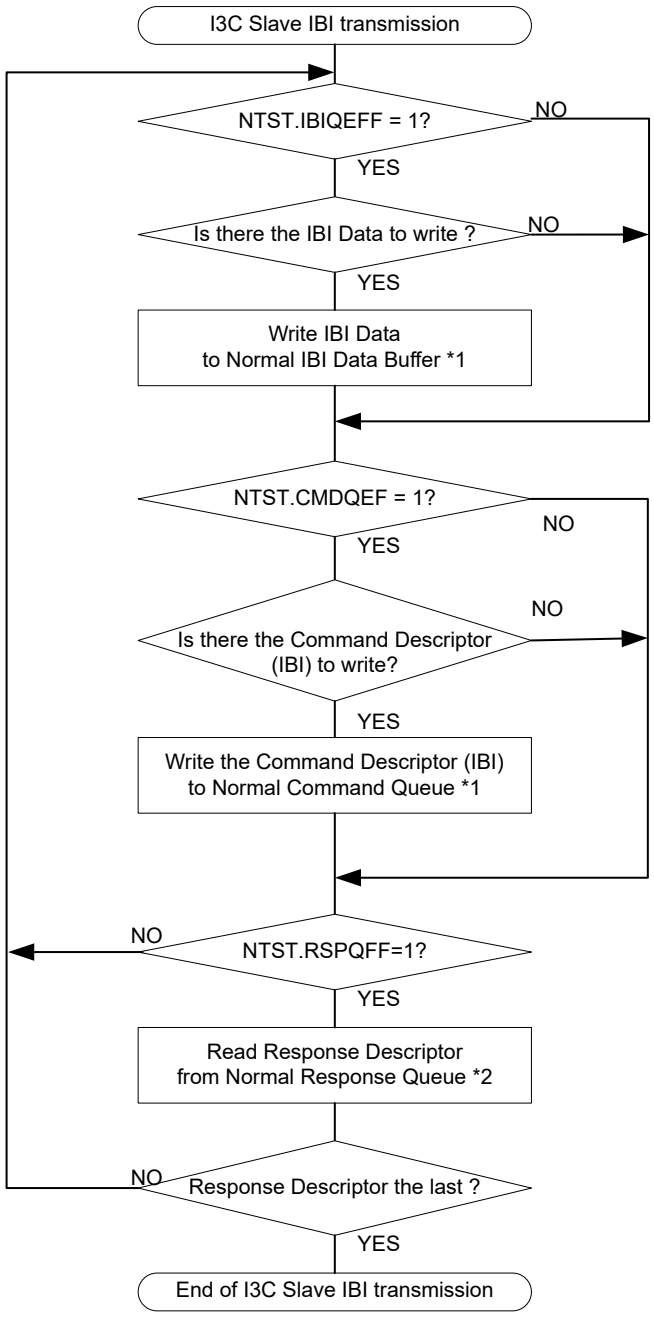
Slave Reception Flow in I3C normal FIFO buffer transfer is common to Legacy I^2C, SDR (Private Transfer, Broadcast CCC, Direct CCC), HDR-DDR, and HDR-TSP/TSL.



Note 1. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 33.155 Example of I3C slave reception flowchart (normal FIFO buffer transfer)

33.3.3.4.5 I3C Slave IBI Transmission Flow



Note 1. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 2. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 33.156 Example of I3C slave IBI transmission flowchart

33.3.3.4.6 I3C Slave Wake-Up Flow

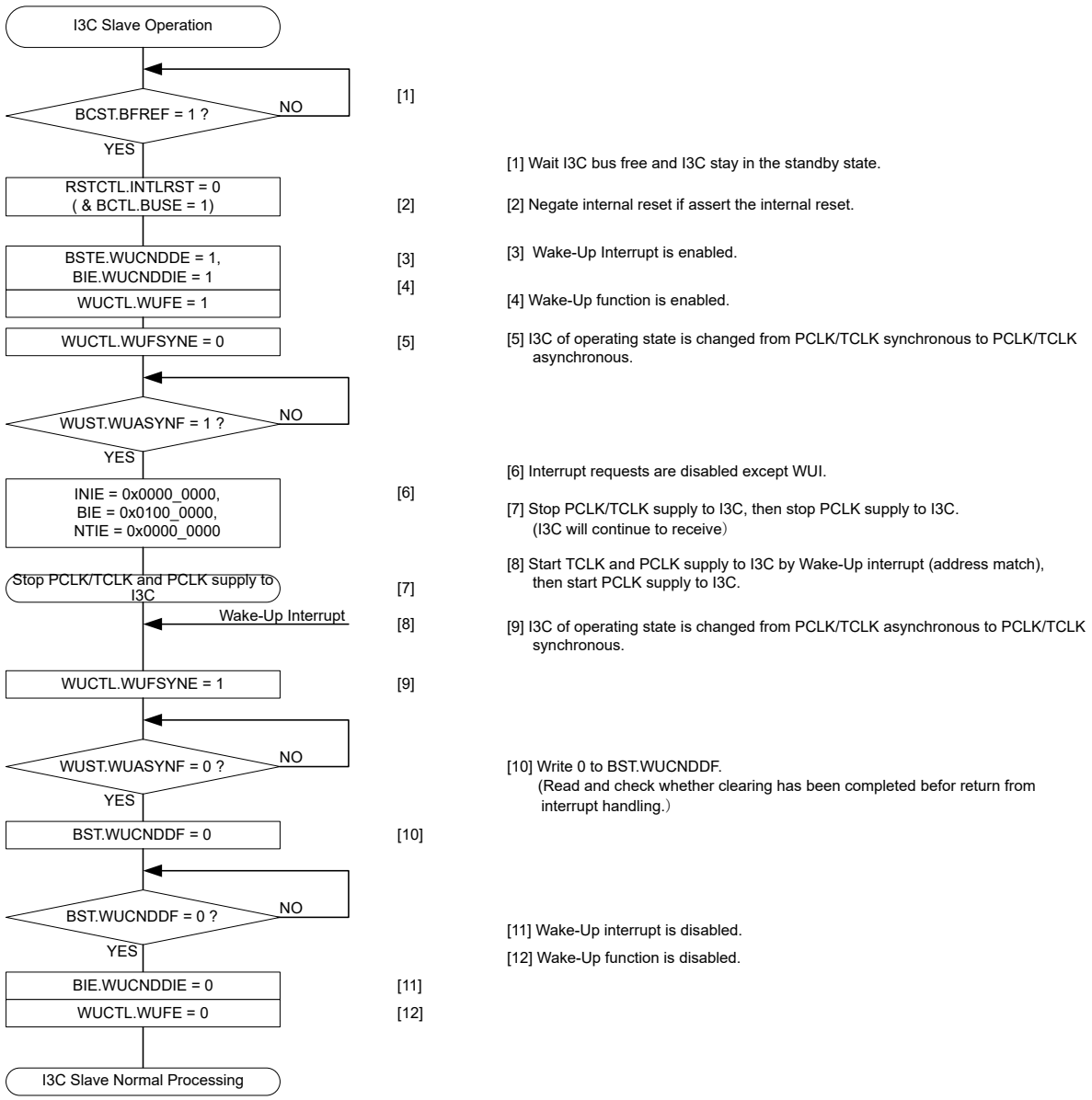


Figure 33.157 Use case of I3C slave wake-up (wake-up recovery by a wake-up interrupt triggered by the match of the slave address)

33.4 Interrupt Sources

I3C can generate the following interrupt requests:

33.4.1 Overview

The I3C has the interrupt factors shown in Table 33.18.

Table 33.18 Interrupt Generation (1 of 2)

Symbol	Interrupt source	Support			
		I2C	I3CM	I3C2M	I3CS
I3C_RESP	Normal Response Queue full	—	✓	✓	✓

Table 33.18 Interrupt Generation (2 of 2)

Symbol	Interrupt source	Support			
		I2C	I3CM	I3C2M	I3CS
I3C_CMD	Normal Command queue empty	—	✓	✓	✓
I3C_IBI	Normal IBI Queue Empty/Full	—	✓	✓	✓
I3C_RX	Normal Rx Data buffer full	✓	✓	✓	✓
I3C_TX	Normal Tx Data buffer empty	✓	✓	✓	✓
I3C_RCV	Normal Receive Status queue full	—	—	✓	✓
I3C_HRESP	High Priority Response queue full	—	✓	✓	—
I3C_HCMD	High Priority Command queue empty	—	✓	✓	—
I3C_HRX	High Priority Rx Data buffer full	—	✓	✓	—
I3C_HTX	High Priority Tx Data buffer empty	—	✓	✓	—
I3C_TEND	Transmit end	✓	—	—	—
I3C_EEI	Non-recoverable internal error	—	✓	✓	✓
	Normal Transfer Error	—	✓	✓	✓
	Normal Transfer Abort	—	✓	✓	✓
	High Priority Transfer Error	—	✓	✓	—
	High Priority Transfer Abort	—	✓	✓	—
	START condition detection	✓	✓	✓	✓
	STOP condition detection	✓	✓	✓	✓
	HDR Exit Pattern detection	—	✓	✓	✓
	NACK detection	✓	—	—	—
	Arbitration lost	✓	—	—	—
	Timeout detection	✓	✓	✓	✓
I3C_STEV	Synchronous Timing	—	✓	✓	✓
I3C_MREFOVF	MREF Counter Overflow	—	✓	✓	—
I3C_MREFCPT	MREF Capture	—	✓	✓	—
I3C_AMEV	Additional Master-initiated bus Event	—	✓	✓	—
I3C_WU	Wake-up condition detection	✓	✓	✓	✓

Note: ✓ : Support
— : Not support

Note: I^2C: I^2C Master/Slave (Single Buffer)
I3CM: I3C Master
I3C2M: I3C Secondary Master
I3CS: I3C Slave

Note: There is a delay time between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared, read the relevant flag again to check whether clearing has been completed, and then return from interrupt handling. Returning from interrupt handling without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.

For I2C Protocol mode:

- Since I3C_TX is an edge-detected interrupted, it does not require clearing. Furthermore, the NTST.TDBEF0 flag (a condition for I3C_TX) is automatically set to 0 when data for transmission are written to NTDTBP0 or a STOP condition is detected (SPCNDDF flag = 1 in BST).

- Since I3C_RX is an edge-detected interrupted, it does not require clearing. Furthermore, the NTST.RDBFF0 flag (a condition for I3C_RX) is automatically set to 0 when data are read from NTDTBP0.

For I3C Protocol mode:

For details, refer to the detailed explanation of each flag bit.

- The I3C_CMD, I3C_TX, I3C_HCMD, I3C_HTX and I3C_IBI (I3C Slave) interrupts are cleared under the following conditions.
  On completion of the last write access by DMAC/DTC.
  Write 0 to this bit after 1 state is read by CPU.
- The I3C_RESP, I3C_IBI (I3C Master), I3C_RX, I3C_RCV, I3C_HRESP and I3C_HRX interrupts are cleared under the following conditions.
  On completion of the last read access by DMAC/DTC.
  Write 0 to this bit after 1 state is read by CPU.

33.4.2 Buffer Operation for Buffer Full/Empty Interrupts

If the conditions for generating the each buffer full/empty interrupts are satisfied while the corresponding IR flag is 1, the interrupt request is not output for the ICU but retained internally (the capacity for internal retention is one request per source).

An interrupt request that was being retained within the ICU is output when the value of the IELSRn.IR flag becomes 0. Internally retained interrupt requests are automatically cleared under normal conditions of usage. Internally retained interrupt requests can also be cleared by writing 0 to the interrupt enable bit within the given peripheral module.

33.5 Event Link Output

I3C handles event output for the event link controller (ELC) corresponding to the following sources.

(1) Communication event

When a Communication event (arbitration-lost detection, detection of NACK, detection of timeout, detection of a START condition, or detection of a STOP condition) occurs, the corresponding event signal can be output for another module through the ELC.

(2) Rx Data buffer full

When a receive data register becomes full, the corresponding event signal can be output for another module through the ELC.

(3) Tx Data buffer empty

When a transmit data register becomes empty, the corresponding event signal can be output for another module through the ELC.

(4) Transmit end

On completion of transfer, the corresponding event signal can be output for another module through the ELC.

Other events are also available. For details, see section 18, Event Link Controller (ELC)18.2.3 ELSRn : Event Link Setting Register n.

33.5.1 Interrupt Handling and Event Linking

I3C module produces four kinds of interrupt: communication event (arbitration-lost detection, detection of NACK, detection of timeout, detection of a START condition, or detection of a STOP condition), Rx Data buffer full, Tx Data buffer empty, and transmit end interrupts. Each of these has an enable bit to control enabling and disabling of the interrupt signal. An interrupt request signal is output for the CPU when an interrupt source condition is satisfied while the setting of the corresponding enable bit is enabled.

The corresponding event link output signals are sent to other modules as event signals via the ELC when the interrupt source conditions are satisfied, regardless of the settings of the interrupt enable bits. For details on interrupt sources, see section 33.4.1. Overview.

33.6 Reset Descriptions

Table 33.19 Register states when issuing each condition (1) (1 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
PRTS	PRTMD	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BCTL	BUSE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSM	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ABT	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	BMDS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	INCBA	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
MSDVAD	MDYADV	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	MDYAD[6:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
RSTCTL	INTLRST	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HRDBRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HTDBRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HRSPQRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HCMDQRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSQRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	IBIQRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RDBRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDBRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RI3CRST	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
PRSST	PRSSTWP	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TRMD	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CRMS*1	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 33.19 Register states when issuing each condition (1) (2 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
INST	INEF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
INSTE	INEE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
INIE	INEIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
INSTFC	INEFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
DVCT	IDX[4:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
IBINCTL	NRSIRCTL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	NRMRCTL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BFCTL	HSME	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	FMPE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SMBS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SCSYNE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SALE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	NALE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	MALE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (The FIFO corresponding to this register is cleared)
Note 1. In I3C mode, CRMS is not reset by INTLRST. In I2C mode, CRMS is reset by INTLRST

Table 33.20 Register states when issuing each condition (2) (1 of 3)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
SVCTL	SVAE[2]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SVAE[1]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SVAE[0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HOAE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVIDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HSMCE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	GCAE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
REFCKCTL	IREFCKS [2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
STDBR	DSBRPO	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SBRHP[5:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SBRLP[5:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SBRHO[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SBRLO[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
EXTBR	EBRHP[5:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EBRLP[5:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EBRHO[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EBRLO[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BFRECDT	FRECYC [8:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BAVLCDT	AVLCYC[8:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BIDLCDT	IDLCYC [17:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 33.20 Register states when issuing each condition (2) (2 of 3)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
OUTCTL	SDODCS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SDOD[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EXCYC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SOCWP	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SCOC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SDOC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
INCTL	SDID[1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DNFE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DNFS[3:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
TMOCTL	TOMDS[1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TOHCTL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TOLCTL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TODTS[1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
WUCTL	WUFE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	WUFSYNE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	WUANFS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	WUACKS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
ACKCTL	ACKTWP	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ACKT	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ACKR	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SCSTRCTL	RWE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ACKTWE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 33.20 Register states when issuing each condition (2) (3 of 3)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
SCSTLCTL	ACKPE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	PARPE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	AAPE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	STLCYC [15:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SVTDLG0	STDLG[15:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (the FIFO corresponding to this register is cleared).

Table 33.21 Register states when issuing each condition (3) (1 of 3)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
STCTL	STOE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
ATCTL	CDIV[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	AMEOE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	MREFOE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ATTRGS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
ATTRG	ATSTRG	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
ATCCNTE	ATCE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CNDCTL	SPCND	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SRCND	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	STCND	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NCMDQP	NCMDQP [31:0]	In reset	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NRSPQP	NRSPQP [31:0]	In reset	In reset	In reset	Saved	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NTDTBP0	NTDTBP0 [31:0]	In reset	In reset	In reset	Saved	Saved	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
NIBIQP	NIBIQP[31:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved	Saved	Saved

Table 33.21 Register states when issuing each condition (3) (2 of 3)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
NRSQP	NRSQP [31:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved	Saved
HCMDQP	HCMDQP [31:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved
HRSPQP	HRSPQP [31:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	In reset	Saved	Saved
HTDTBP	HTDTBP [31:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	In reset	In reset
NQTHCTL	IBIQTH[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	IBIDSSZ[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQTH [7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQTH [7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NTBTHCTL0	RXSTTH [2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TXSTTH[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RXDBTH [2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TXDBTH [2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NRQTHCTL	RSQTH[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
HQTHCTL	RSPQTH [7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQTH [7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
HTBTHCTL	RXSTTH [2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TXSTTH[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RXDBTH [2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TXDBTH [2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 33.21 Register states when issuing each condition (3) (3 of 3)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
BST	WUCNDDF	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TODF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ALF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TENDF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	NACKDF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HDREXDF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SPCNDDF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	STCNDDF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BSTE	WUCNDDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TODE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ALE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TENDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	NACKDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HDREXDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SPCNDDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	STCNDDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (the FIFO corresponding to this register is cleared).

Table 33.22 Register states when issuing each condition (4) (1 of 3)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
BIE	WUCNDDIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TODIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ALIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TENDIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	NACKDIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HDREXDIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SPCNDDIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	STCNDDIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BSTFC	WUCNDDFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TODFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ALFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TENDFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	NACKDFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HDREXDFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SPCNDDFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	STCNDDFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 33.22 Register states when issuing each condition (4) (2 of 3)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
NTST	RSQFF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved	Saved
	TEF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TABTF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFF	In reset	In reset	In reset	Saved	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQEF	In reset	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	IBIQEFF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved	Saved	Saved
	RDBFF0	In reset	In reset	In reset	Saved	Saved	Saved	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	TDBEF0	In reset	In reset	In reset	Saved	Saved	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NTSTE	RSQFE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TEE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TABTE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQEE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	IBIQEFE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RDBFE0	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDBEE0	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 33.22 Register states when issuing each condition (4) (3 of 3)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
NTIE	RSQFIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TEIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TABTIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQEIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	IBIQEFIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RDBFIE0	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDBEIE0	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (the FIFO corresponding to this register is cleared).

Table 33.23 Register states when issuing each condition (5) (1 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
NTSTFC	RSQFFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TEFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TABTFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQEFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	IBIQEFFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RDBFFC0	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDBEFC0	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 33.23 Register states when issuing each condition (5) (2 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
HTST	TEF	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TABTF	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	In reset	Saved	Saved
	CMDQEF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved
	RDBFF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	In reset
	TDBEF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	In reset	Saved
HTSTE	TEE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TABTE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQEE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RDBFE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDBEE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
HTIE	TEIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TABTIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQEIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RDBFIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDBEIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (the FIFO corresponding to this register is cleared).

Table 33.24 Register states when issuing each condition (6) (1 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
HTSTFC	TEFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TABTFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQEFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RDBFFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDBEFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BCST	BIDLF	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	BAVLF	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	BFREF	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SVST	SVAF[2]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SVAF[1]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SVAF[0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HOAF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVIDF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HSMCF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	GCAF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
WUST	WUASYNF	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
MRCCPT	MRCCPT [31:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 33.24 Register states when issuing each condition (6) (2 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
DATBASm (m = 0 to 7)	DVTYP	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVNACK [1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVDYAD [7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVIBITS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVMRRJ	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVSIRRJ	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVIBIPL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVADLS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVSTAD [9:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (the FIFO corresponding to this register is cleared).

Table 33.25 Register states when issuing each condition (7) (1 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
EXDATBAS	EDTYP	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EDNACK [1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EDDYAD [7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EDADLS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EDSTAD [9:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SDATBASn (n = 0 to 2)	SDDYAD [6:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SDIBIPL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SDADLS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SDSTAD [9:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
MSDCTm (m = 0 to 7)	RBCRn	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 33.25 Register states when issuing each condition (7) (2 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
SVDCT	]TBCRn	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDCR[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (the FIFO corresponding to this register is cleared).

Table 33.26 Register states when issuing each condition (8) (1 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
SDCTPIDL	SDCTPIDL [31:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SDCTPIDH	SDCTPIDH [31:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SVDVADn (n = 0 to 2)	SDYADV	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SSTADV	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SADLG	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SVAD[9:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CSECMD	MSRQE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SVIRQE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CEACTST	ACTST[3:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CMWLG	MWLG[15:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CMRLG	IBIPSZ[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	MRLG[15:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CETSTMD	TSTMD[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CGDVST	VDRSV[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ACTMD[1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	PRTE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	PNDINT[3:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 33.26 Register states when issuing each condition (8) (2 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
CMDSPW	MSWDR[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CMDSPR	CDTTIM[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	MSRDR[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CMDSPT	MRTE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	MRTTIM [23:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (the FIFO corresponding to this register is cleared).

Table 33.27 Register states when issuing each condition (9) (1 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
CETSM	INAC[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	FREQ[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SPTASYN [3:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SPTSYN	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CGHDRCAP	TSLEN	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TSPEN	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DDREN	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CETSS	ICOVF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ASYNE[3:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SYNE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BITCNT	BCNT[4:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 33.27 Register states when issuing each condition (9) (2 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
NQSTLV	IBISCNT[4:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved	Saved	Saved
	IBIQLV[7:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved	Saved	Saved
	RSPQLV [7:0]	In reset	In reset	In reset	Saved	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQFLV [7:0]	In reset	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NDBSTLV0	RDBLV[7:0]	In reset	In reset	In reset	Saved	Saved	Saved	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	TDBFLV[7:0]	In reset	In reset	In reset	Saved	Saved	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NRSQSTLV	RSQLV[7:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved	Saved
HQSTLV	RSPQLV [7:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	In reset	Saved	Saved
	CMDQLV [7:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved
HDBSTLV	RDBLV[7:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	In reset
	TDBFLV[7:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	In reset	Saved
PRSTDBG	SDOLV	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SCOLV	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SDILV	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SCILV	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
MSERRCNT	M2ECNT [7:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SC1CPT	SC1C[15:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SC2CPT	SC2C[15:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CECTL	CLKE	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (the FIFO corresponding to this register is cleared).

33.7 Usage Notes

33.7.1 Settings for the Operating Clock

The following relation is required between the frequencies of the bus clock (PCLK) and transfer clock(TCLK).

$TCLK/2 \leq PCLK \leq TCLK$

33.7.2 Settings for the SCL frequency Clock

As described in Chapter 5.1.2.4 of MIPI I3C Specification V1.0, the SCL frequency setting is required depending on the bus configuration.

In Mixed Bus with both I3C and legacy I2C devices are present on the bus, If the Legacy I2C device has a 50ns Spike Filter, I3C Master can change the frequency by setting the SCL High period to less than 50ns (tDIG_H_MIXED).

33.7.3 Module-stop function

I3C operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The I3C module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

33.7.4 Notes on Rewriting the PRTS.PRTMD Bit

When the PRTS.PRTMD bit is rewritten, the SCL terminal (and SDA terminal) may output low. In this case, the bus line is unintentionally put into a busy state, and communication cannot be started even if the communication start setting is made after that.

Set the RSTCTL.INTRST bit to 1 and issue an internal software reset before rewriting the PRTMD bit. After rewriting, set the RSTCTL.INTRST bit to 0 to release the internal software reset.

However, no countermeasure is necessary if the following conditions are applicable:

- When the PRTS.PRTMD bit is not changed from its initial value of 1 (I^2C protocol mode) to 0 (I3C protocol mode)
- When the clock source selected by SCKSCR.CKSEL[2:0] and I3CCKCR.I3CCKSEL[3:0] are the same.
- In master mode, when the timeout error detection function is enabled under the following conditions and a software reset is set in the I3C error interruption routine to release unintended low output.
  1. Set the BSTE.TODE bit to 1 to enable the timeout error detection function.
  2. Select TMOCTL.TOMDS[1:0] = 00b or 01b as the timeout operation mode and set the TMOCTL.TOLCTL bit = 1 and TMOCTL.TOHCTL bit = 1 to treat the SCL line fixed at low and fixed at high as errors.
  3. In the I3C error interruption routine, set the RSTCTL.RI3CRST bit to 1 to perform an I3C software reset, or set the RSTCTL.INTLRST bit to 1 to perform an internal software reset.
- In slave mode, when the timeout error detection function is enabled under the following conditions and a software and TMOCTL.TOHCTL bit = 1 to treat the SCL line fixed at low and fixed at high as errors.
  1. Set the BSTE.TODE bit to 1 to enable the timeout error detection function.
  2. Select TMOCTL.TOMDS[1:0] = 01b as the timeout operation mode and set TMOCTL.TOLCTL = 1 and TMOCTL.TOHCTL = 1 to treat the SCL line fixed at low and fixed at high as errors.
  3. In the I3C error interruption routine, set the RSTCTL.RI3CRST bit to 1 to perform an I3C software reset, or set the RSTCTL.INTLRST bit to 1 to perform an internal software reset.

34. CAN with Flexible Data-rate (CANFD)

34.1 Overview

The CAN with Flexible Data-rate (CANFD) supports the following functions:

- CAN with Flexible Data-rate.*1

Note 1. This feature is not available in the classical CAN function.

The CANFD module has a flexible message buffer and FIFO structure that meet the requirements of various applications. It also provides test modes to achieve high testability of the module that can be useful for power-on testing.

This specification describes of the CANFD module.

The CANFD mode is only available in certain products that support it.

34.1.1 CANFD Module

Table 34.1 CANFD module specifications (1 of 2)

Parameter		Specifications
Communication		CAN functionality conforms to CANFD ISO 11898-1 (2015)
Protocol engine version		RS-CANFD_PE V3.0
Data transfer rate	CANFD*1 *2	Up to 1 Mbps for arbitration phase and up to 8 Mbps for data phase, individually for CAN channel
	Classical CAN	Up to 1 Mbps
Operation frequency of Peripheral module clock		120 MHz (PCLKA) RAM clock: 240 MHz (PCLKE)
Data Link Layer (DLL) clock		8 MHz to 80MHz The following clocks can be selected. ● CANMCLK : External Oscillator Clock ● CANFDCLK : CANFD Core Clock
Input/Output pins		CRXn/CTXn (n = 0, 1)
CAN channels		2 channels
Selectable ID type		11-bit Standard ID
		11-bit Standard ID + 18-bit Extended ID
Selectable frame type		Data frame (RTR = 0) (CAN and CANFD frames)
		Remote frame (RTR = 1) (only CAN frames)
Variable data byte count for data frames		DLC range: 0 to F
Message buffer		Up to 16 reception message buffers
		4 transmit message buffers per channel
		1 transmission queue per channel Automatic message transfer into transmission queues supported
FIFO number		2 reception FIFO buffers 1 COMMON FIFO individually configurable as: ● Reception FIFO ● Transmission FIFO
Automatic delay interval timer for transmission		The delay timer can be applied to: ● Transmission FIFO

Table 34.1 CANFD module specifications (2 of 2)

Parameter	Specifications
Enhanced reception filtering	Support of 11 bits and 29 bits CAN identifier
	Programmable 29 bits CAN identifier acceptance filter mask for each entry
	Programmable routing capability for each FIFO and reception message buffers (up to 2 routing destinations)
	RTR and IDE masking
	Data Length Code (DLC) filter
	Message buffer payload overload protection
	Payload filter
	Updating Acceptance Filter List (AFL) entry during communication
General software support	Automatic label information added to receive message (for upper software layer support)
Timer	TX and RX Time Stamp function
Power down function	Module start stop function for each CAN node (Channel and Global Sleep mode)
RAM	RAM ECC protected (2 bits error detection, 1-bit error correction)
Module-stop function	Module-stop state can be set for each channels to reduce power consumption.
TrustZone Filter	Security and Privilege attribution can be set for each channels.

Note 1. The CANFD mode is available only for CANFD supported product.
Note 2. The bit rate for communications depends on the board design and external environment. Determine if following sufficient evaluation.

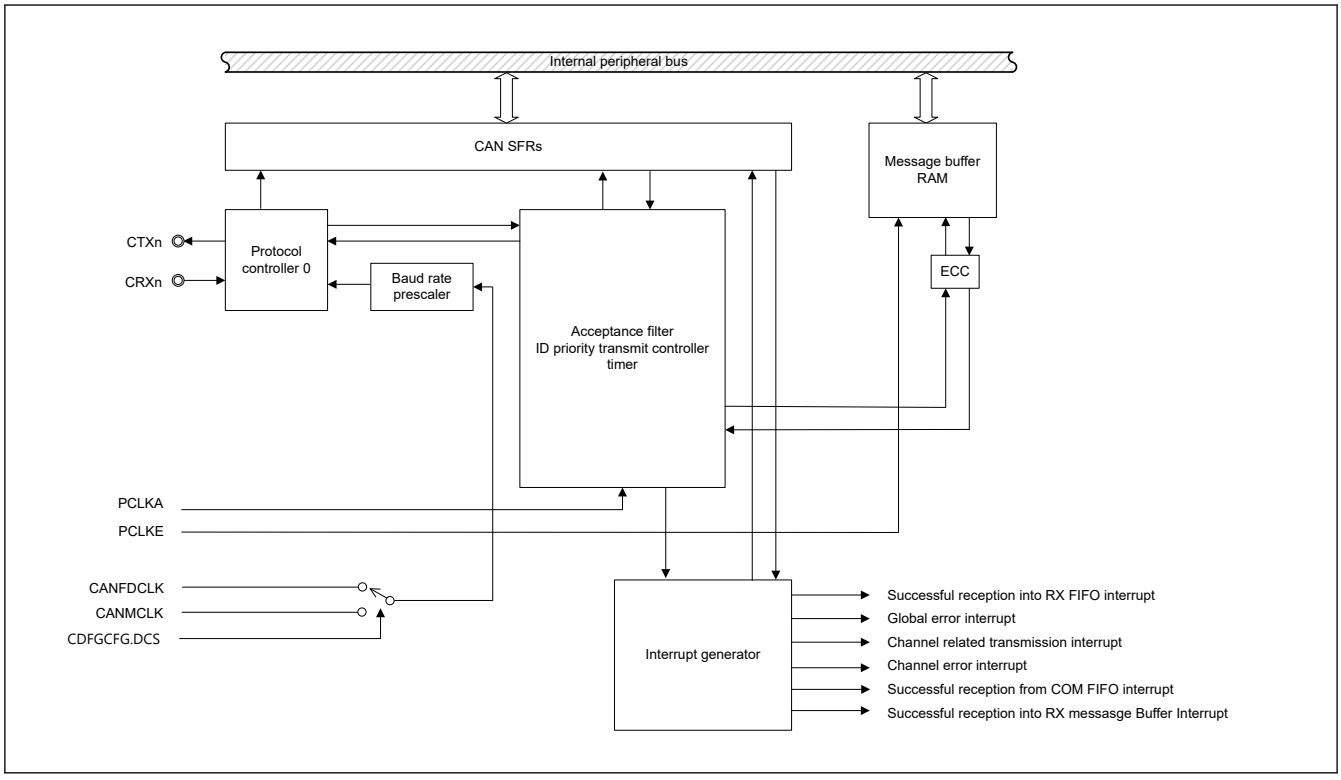


Figure 34.1 Overview of the CANFD module

- CRXn/CTXn:
  Input/Output pins of the CANFD module
- Protocol controller:

Handles CAN protocol processing such as bus arbitration, bit timing at transmission and reception, stuffing, error handling

- Message buffer RAM (MBRAM):
  This RAM is used to store messages after reception or for transmission using a normal message buffer or a FIFO. Each message entry has an individual ID, data length code, data field, message pointer for upper layer application usage and a time stamp.
  This RAM is used to store the message acceptance filtering entries. Each acceptance filter entry has an individual ID, data length code, data field, message pointer for upper layer application usage and message direction pointer.
- Acceptance filter list RAM (AFLRAM):
  Performs filtering of received messages. The entries in the Acceptance filter list RAM are used for the filtering process.
- Two timers:
  - Reception Timestamp function
  - Transmission separation time for FIFO buffers
- Interrupt generator:
  Generates several types of global and channel interrupts
- CAN Special Function Registers (SFRs):
  Registers associated with CAN. See section 34.2. Register Descriptions.

34.1.2 Clock Restriction

For the CAN communication the following restriction for the clocks should be satisfied:

- PCLKE / 2 = PCLKA ≥ CANFDCLK
- PCLKE / 2 = PCLKA ≥ CANMCLK

To avoid missing events the CAN engine clock (CANFDCLK or CANMCLK) frequency must be less than the PCLKA clock frequency.

To avoid loss of CAN message, the PCLKA should be set to a clock with a frequency depend on the CAN communication Baud Rate. The constraint of a baud rate and a PCLKA clock is shown in Table 34.2.

Table 34.2 Clock restriction

	Baud rate	PCLKA
CANFD	1 Mbps Nominal 8 Mbps Data*1	PCLKA ≥ 40 MHz
	500 Kbps Nominall 5 Mbps Data	PCLKA ≥ 32 MHz
Classical CAN	1 Mbps Data	PCLKA ≥ 32 MHz

Note 1. The bit rate for communications depends on the board design and external environment. Determine if following sufficient evaluation.

The frequency of CANFDCLK and CANMCLK depend on the required baud rate. For information how to configure the baud rate, refer to section 34.4.1.3. Baud Rate.

34.2 Register Descriptions

34.2.1 Register Table

The reset value shown for the RAM area, consisting of CFDGAFLIDr, CFDGAFLMr, CFDGAFLP0r, CFDGAFLP1r, CFDRMBCPb, CFDRFMBCPb, CFDCFMBCP0, CFDTMBCPb, CFDTHLACC0, CFDTHLACC1 and CFDRPGACCk is valid after initialization of a hardware reset. See section 34.4.2. CAN Module Configuration after Hardware Reset for details of the initialization process.

If a write access with a size of 8 or 16 bits is performed for the RAM area, then the CANFD module does a read-modify write-access to the RAM location, because the RAM requires a 32-bit access through the ECC module.

For single bit error, the correct data is written back. For multiple bit errors, unknown data is written back.

Do not access the space where the register is not assigned.

The read data from the space where the register is not assigned is unknown.

34.2.2 CFDC0NCFG : Nominal Bitrate Configuration Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0000

Bit position:	31						25	24						17	16						10	9						0
Bit field:	NTSEG2[6:0]							NTSEG1[7:0]								NSJW[6:0]							NBRP[9:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	NBRP[9:0]	Channel Nominal Baud Rate Prescaler Nominal baud rate prescaler division ratio	R/W
16:10	NSJW[6:0]	Resynchronization Jump Width 0x00: 1 Tq 0x01: 2 Tq ⋮ 0x7E: 127 Tq 0x7F: 128 Tq	R/W
24:17	NTSEG1[7:0]	Timing Segment 1 0x00: Reserved 0x01: 2 Tq 0x02: 3 Tq 0x03: 4 Tq ⋮ 0xFE: 255 Tq 0xFF: 256 Tq	R/W
31:25	NTSEG2[6:0]	Timing Segment 2 0x00: Reserved 0x01: 2 Tq ⋮ 0x7E: 127 Tq 0x7F: 128 Tq	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Tq means time quantum.

This register configures the transmission/reception nominal baud rate parameters of the channels.

NBRP[9:0] bits (Channel Nominal Baud Rate Prescaler)

The NBRP[9:0] bits are used to define the peripheral bus clock periods contained in a time quantum.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CANFD channel is in CH_RESET or CH_HALT mode.

NSJW[6:0] bits (Resynchronization Jump Width)

The NSJW[6:0] bits set the synchronization jump width. A value from 1 to 128 time quanta can be set.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CANFD channel is in CH_RESET or CH_HALT mode.

NTSEG1[7:0] bits (Timing Segment 1)

The NTSEG1[7:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. These bits contain the propagation segment.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CANFD channel is in CH_RESET or CH_HALT mode.

Additionally, configure a Tq value only between 2 and 256, inclusive. See section 34.4.1.2. CAN Bit Timing for more details.

NTSEG2[6:0] bits (Timing Segment 2)

The NTSEG2[6:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CANFD channel is in CH_RESET or CH_HALT mode.

Additionally, configure a Tq value only between 2 and 128, inclusive.

34.2.3 CFDC0CTR : Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ROM	BFT	—	—	—	CTMS[1:0]		CTME	ERRD	BOM[1:0]		—	TDCVFIE	SOCOIE	EOCOIE	TAIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	CSLPR	CHMDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
1:0	CHMDC[1:0]	Channel Mode Control 0 0: Channel operation mode request 0 1: Channel reset request 1 0: Channel halt request 1 1: Keep current value	R/W
2	CSLPR	Channel Sleep Request 0: Channel sleep request disabled 1: Channel sleep request enabled	R/W
3	RTBO	Return from Bus-Off 0: Channel is not forced to return from bus-off 1: Channel is forced to return from bus-off	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	BEIE	Bus Error Interrupt Enable 0: Bus error interrupt disabled 1: Bus error interrupt enabled	R/W
9	EWIE	Error Warning Interrupt Enable 0: Error warning interrupt disabled 1: Error warning interrupt enabled	R/W
10	EPIE	Error Passive Interrupt Enable 0: Error passive interrupt disabled 1: Error passive interrupt enabled	R/W
11	BOEIE	Bus-Off Entry Interrupt Enable 0: Bus-off entry interrupt disabled 1: Bus-off entry interrupt enabled	R/W
12	BORIE	Bus-Off Recovery Interrupt Enable 0: Bus-off recovery interrupt disabled 1: Bus-off recovery interrupt enabled	R/W
13	OLIE	Overload Interrupt Enable 0: Overload interrupt disabled 1: Overload interrupt enabled	R/W

Bit	Symbol	Function	R/W
14	BLIE	Bus Lock Interrupt Enable 0: Bus lock interrupt disabled 1: Bus lock interrupt enabled	R/W
15	ALIE	Arbitration Lost Interrupt Enable 0: Arbitration lost interrupt disabled 1: Arbitration lost interrupt enabled	R/W
16	TAIE	Transmission Abort Interrupt Enable 0: TX abort interrupt disabled 1: TX abort interrupt enabled	R/W
17	EOCOIE	Error Occurrence Counter Overflow Interrupt Enable 0: Error occurrence counter overflow interrupt disabled 1: Error occurrence counter overflow interrupt enabled	R/W
18	SOCOIE	Successful Occurrence Counter Overflow Interrupt Enable 0: Successful occurrence counter overflow interrupt disabled 1: Successful occurrence counter overflow interrupt enabled	R/W
19	TDCVFIE*1	Transceiver Delay Compensation Violation Interrupt Enable 0: Transceiver delay compensation violation interrupt disabled 1: Transceiver delay compensation violation interrupt enabled	R/W
20	—	This bit is read as 0. The write value should be 0.	R/W
22:21	BOM[1:0]	Channel Bus-Off Mode 0 0: Normal mode (comply with ISO 11898-1) 0 1: Entry to Halt mode automatically at bus-off start 1 0: Entry to Halt mode automatically at bus-off end 1 1: Entry to Halt mode (during bus-off recovery period) by software	R/W
23	ERRD	Channel Error Display 0: Only the first set of error codes displayed 1: Accumulated error codes displayed	R/W
24	CTME	Channel Test Mode Enable 0: Channel test mode disabled 1: Channel test mode enabled	R/W
26:25	CTMS[1:0]	Channel Test Mode Select 0 0: Basic test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (External loopback mode) 1 1: Self-test mode 1 (Internal loopback mode)	R/W
29:27	—	These bits are read as 0. The write value should be 0.	R/W
30	BFT	Bit Flip Test 0: First data bit of reception stream not inverted 1: First data bit of reception stream inverted	R/W
31	ROM*1	Restricted Operation Mode 0: Restricted operation mode disabled 1: Restricted operation mode enabled	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. These bits are not available in the classical CAN function.

Channel Control register controls the modes of the related channel. It is used to enable generation of interrupts if errors are detected on the CAN bus connected to this channel. It is also used to configure the channel in test mode.

CHMDC[1:0] bits (Channel Mode Control)

The CHMDC[1:0] bits can be used to configure modes of the CAN channel.

CAN mode transitions are described in more details in section 34.3.3. Channel Modes.

Setting CHMDC[1:0] bits to 11b has no effect. When the CANFD module is in GL_HALT mode, these bits can only be set to 10b or 01b. These bits cannot be set in CH_SLEEP mode.

These bits can change automatically when transitioning to Halt mode by the CFDC0CTR.BOM settings.

If CPU write access to CFDC0CTR.CHMDC occurs at the same time when the CAN channel enters Halt mode (at the start of bus-off when CFDC0CTR.BOM = 01b, or at the end of bus-off when CFDC0CTR.BOM = 10b), then the CPU write access has the highest priority.

The CAN channel changes the value of CFDC0CTR.CHMDC within the Channel Control Registers for the specified cases only if the CFDC0CTR.CHMDC value is 00b (Operation mode).

CSLPR bit (Channel Sleep Request)

When the CSLPR bit is 1, a Sleep mode request is generated for the corresponding CAN channel

When this bit is 0, a request to exit Sleep mode is generated for the related CANFD channel.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_SLEEP mode.

RTBO bit (Return from Bus-Off)

When the protocol controller of the CAN channel enters bus-off state, you can force a recovery from bus-off state by setting the RTBO bit in the Channel Control Register to 1.

The error state changes from bus-off state to integrating with a maximum delay of 1 CAN bit time.

When the RTBO bit is set to 1, the REC and TEC registers are initialized and the Bus-Off Status bit (Channel Bus-off Status, CFDC0STS.BOSTS) is set to 0.

Registers other than the REC and TEC registers are not initialized by this command. Even if CFDC0CTR.BORIE is set, a bus-off recovery interrupt is not generated by this recovery from the bus-off state.

The RTBO bit cannot be set in CH_SLEEP mode. Setting this bit in any state other than bus-off state has no effect and the bit is cleared immediately. The read value is always 0.

Return from the Bus-Off command should be used only when CFDC0CTR.BOM is set to 00b.

Only write to this bit when the related CANFD channel is in CH_OPERATION mode. This bit is automatically cleared when set by software.

BEIE bit (Bus Error Interrupt Enable)

When the BEIE and the CFDC0ERFL.BEF bits are both 1, an error interrupt request is generated.

This bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

EWIE bit (Error Warning Interrupt Enable)

When the EWIE and the CFDC0ERFL.EWF bits are both 1, an error interrupt request is generated.

The EWIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

EPIE bit (Error Passive Interrupt Enable)

An error interrupt request is generated when the EPIE bit and the CFDC0ERFL.EPF are both 1.

The EPIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

BOEIE bit (Bus-Off Entry Interrupt Enable)

When the BOEIE and the CFDC0ERFL.BOEF bits are both 1, an error interrupt request is generated.

The BOEIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

BORIE bit (Bus-Off Recovery Interrupt Enable)

When the BORIE and the CFDC0ERFL.BORF bits are both 1, an error interrupt request is generated.

The BORIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

OLIE bit (Overload Interrupt Enable)

When the OLIE and the CFDC0ERFL.OVLF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

BLIE bit (Bus Lock Interrupt Enable)

When the BLIE and the CFDC0ERFL.BLF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

ALIE bit (Arbitration Lost Interrupt Enable)

When the ALIE and the CFDC0ERFL.ALF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

TAIE bit (Transmission Abort Interrupt Enable)

When the TAIE bit is 1 and a transmission is successfully aborted from a TX MB belonging to the corresponding CAN channel, an interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

EOCOIE bit (Error Occurrence Counter Overflow Interrupt Enable)

When the EOCOIE bit is 1 and the CFDC0FDSTS.EOCO bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The EOCOIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

SOCOIE bit (Successful Occurrence Counter Overflow Interrupt Enable)

When the SOCOIE bit is 1 and the CFDC0FDSTS.SOCO bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The SOCOIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

TDCVFIE bit (Transceiver Delay Compensation Violation Interrupt Enable)

When the TDCVFIE bit is 1 and the CFDC0FDSTS.TDCVF bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The TDCVFIE bit cannot be set in CH_SLEEP mode.

Only write to this bit when the related CANFD channel is in CH_RESET mode. Do not set this bit when in Classical CAN mode.

Note: This bit is not available in the classical CAN function.

BOM[1:0] bits (Channel Bus-Off Mode)

The BOM[1:0]bits control the timing of the recovery from Bus-Off mode of the CANFD Channel.

Do not write to these bits in CH_SLEEP mode. Only write to these bits when the related CANFD channel is in CH_RESET mode.

Only write to these bits when the related CANFD channel is in CH_RESET mode.

ERRD bit (Channel Error Display)

The ERRD bit controls the display mode of the error flag bits [14:8] in the Channel Error Flag Register (CFDC0ERFL).

If the ERRD bit is 0 and more than one error occur at the same time, the error flag bits are set for all the errors that occurred at the same time. No further errors are flagged until CFDC0ERFL[14:8] is cleared.

Do not write to the ERRD bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

CTME bit (Channel Test Mode Enable)

The CTME bit enables the channel test modes.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_HALT mode.

CTMS[1:0] bits (Channel Test Mode Select)

The CTMS[1:0] bits are used to select the required test mode.

Do not write to these bits in CH_SLEEP or CH_RESET mode. Only write to these bits when the related CANFD channel is in CH_HALT mode.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

BFT bit (Bit Flip Test)

The BFT bit checks the internal CRC generator logic of the protocol controller.

It inverts the first bit (ID bit) of the CAN message data stream being received, so that the internal generated CRC result will not match the received CRC value of the frame. Refer to the bit stuffing rule, when using this feature, as there is the possibility of receiving a stuff error (due to the inversion) rather than a CRC error.

The internal generated CRC value is always observed in the following registers:

- CFDC0ERFL.CRCREG (Classical CAN frames)
- CFDC0FDCRC.CRCREG (CANFD frames).*1

Note 1. This feature is not available in the classical CAN function.

Some restriction exist when using this bit:

Other CAN node will send a reference message and the receiver node(s) can invert one bit of incoming bit stream.

Note: The transmitter and receiver modes share the same CRC generator, therefore it is not necessary to consider the modes separately when testing.

The Bit Flip test mode is enabled if the BFT (new control signal that inverts the first bit of the bit stream) and CTME bits are both 1 and CFDC0CTR.CTMS is 0x00.

If this function is used by a transmitting node, a bit error or an arbitration lost will occur.

Do not write to the BFT bit in CH_SLEEP mode. Users should not use this function when the Self test mode 1 (Internal Loop back mode). Only write to this bit when the related CANFD channel is in CH_HALT mode.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

ROM bit (Restricted Operation Mode)

When the ROM and CTME bits are both 1, the restricted operation mode is enabled. This mode should only be used in basic test mode (CFDC0CTR.CTMS[1:0] = 00b).

The ROM bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_HALT mode.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. Do not set this bit when in Classical CAN mode.

Note: This bit is not available in the classical CAN function.

34.2.4 CFDC0STS : Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TEC[7:0]								REC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ESIF	COMSTS	RECSTS	TRMSTS	BOSTS	EPSTS	CSLPSTS	CHLTSTS	CRSTSTS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
0	CRSTSTS	Channel Reset Status 0: Channel not in Reset mode 1: Channel in Reset mode	R
1	CHLTSTS	Channel Halt Status 0: Channel not in Halt mode 1: Channel in Halt mode	R
2	CSLPSTS	Channel Sleep Status 0: Channel not in Sleep mode 1: Channel in Sleep mode	R
3	EPSTS	Channel Error Passive Status 0: Channel not in error passive state 1: Channel in error passive state	R
4	BOSTS	Channel Bus-Off Status 0: Channel not in bus-off state 1: Channel in bus-off state	R
5	TRMSTS	Channel Transmit Status 0: Channel is not transmitting 1: Channel is transmitting	R
6	RECSTS	Channel Receive Status 0: Channel is not receiving 1: Channel is receiving	R
7	COMSTS	Channel Communication Status 0: Channel is not ready for communication 1: Channel is ready for communication	R
8	ESIF*1	Error State Indication Flag 0: No CANFD message has been received when the ESI flag was set 1: At least one CANFD message was received when the ESI flag was set	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
23:16	REC[7:0]	Reception Error Count These bits increment or decrement the counter value according to error status of the CAN channel during reception.	R
31:24	TEC[7:0]	Transmission Error Count These bits increment or decrement the counter value according to error status of the CAN channel during transmission.	R

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is not available in the classical CAN function.

Channel Status Register shows the mode, error and transmission or reception status of the related channel together with its reception and transmission error count values.

CRSTSTS bit (Channel Reset Status)

The CRSTSTS bit indicates whether the related CAN channel is in Reset mode.

This bit is set automatically when the related CAN channel enters Channel Reset mode. When the mode is changed from Reset mode to Sleep mode, the CRSTSTS bit remains 1.

This bit is cleared automatically when the related CAN channel exits the Channel Reset mode, except when changing to Sleep mode.

CHLTSTS bit (Channel Halt Status)

The CHLTSTS bit indicates whether the related CAN channel is in Halt mode.

This bit is set automatically when the related CAN module enters Halt mode, and is cleared automatically when the related CAN module exits Halt mode.

CSLPSTS bit (Channel Sleep Status)

The CSLPSTS bit indicates whether the related CAN channel is in Sleep mode.

This bit is set automatically when the related CANFD channel enters Sleep mode, and is cleared automatically when the related CANFD channel exits Sleep mode.

EPSTS bit (Channel Error Passive Status)

The EPSTS bit indicates whether the related CANFD channel has entered the error passive state.

This bit is set automatically when the value of the CAN Transmission or Reception Counter Register exceeds the value of 0x7F.

This bit is cleared automatically when the related CANFD channel exits the error passive state or enters Reset mode.

BOSTS bit (Channel Bus-Off Status)

The BOSTS bit indicates whether the related CANFD channel has entered the error bus-off state.

This bit is set automatically when the value of the related CAN Transmission Error Count Register exceeds 0xFF and the related CANFD channel is in the bus-off state (CAN Transmission Error Count Register > 0xFF).

This bit is cleared automatically when the related CANFD channel exits bus-off state.

TRMSTS bit (Channel Transmit Status)

The TRMSTS bit indicates whether the related CANFD channel is transmitting a message.

This bit is set automatically when the related CANFD channel is operating as a transmitter node or is in the bus-off state.

This bit is cleared automatically when the related CANFD channel is in the bus-idle state or starts operating as a receiver node.

RECSTS bit (Channel Receive Status)

The RECSTS bit indicates whether the related CANFD channel is receiving a message.

This bit is set automatically when the related CANFD channel is operating as a receiver node.

This bit is cleared automatically when the related CANFD channel is in the bus-idle state or starts operating as a transmitter node.

COMSTS bit (Channel Communication Status)

The COMSTS bit indicates whether the related CANFD channel is ready for communication.

This bit is set automatically when the related CANFD channel is ready to perform communication following the detection of 11 consecutive recessive bits after exiting the Reset or Halt mode.

This bit is cleared automatically when the related CANFD channel is in CH_RESET or CD_HALT mode.

Note: This bit is 1 during bus-off state.

ESIF bit (Error State Indication Flag)

The ESIF bit is set when the ESI bit is sampled recessively for a reception CAN message without any error. When in Loopback or Mirror mode, the self-transmitted messages are considered reception messages.

If a set from the CANFD channel occurs simultaneously with a clear by a write access, then the bit is set.

This bit is cleared by writing 0 to it. This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

Note: This bit is not available in the classical CAN function.

REC[7:0] bits (Reception Error Count)

The REC[7:0] bits increment or decrement the counter value according to error status of the CANFD channel during reception, and display the value of the REC error counter.

The value in bus-off state is indeterminate.

These bits are cleared automatically when the CANFD module enters GL_RESET or the CANFD channel is in CH_RESET mode.

TEC[7:0] bits (Transmission Error Count)

The TEC[7:0] bits increment or decrement the counter value according to error status of the CANFD channel during transmission, and display the value of the TEC error counter.

Only write to these bits when in test mode and CANFD channel is in CH_HALT mode.

These bits are cleared automatically when CANFD module is in GL_RESET or CANFD channel is in CH_RESET mode.

34.2.5 CFDC0ERFL : Error Flag Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x000C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	CRCREG[14:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ADERR	B0ERR	B1ERR	CERR	AERR	FERR	SERR	ALF	BLF	OVLF	BORF	BOEF	EPF	EWF	BEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BEF	Bus Error Flag 0: Channel bus error not detected 1: Channel bus error detected	R/W
1	EWF	Error Warning Flag 0: Channel error warning not detected 1: Channel error warning detected	R/W
2	EPF	Error Passive Flag 0: Channel error passive not detected 1: Channel error passive detected	R/W
3	BOEF	Bus-Off Entry Flag 0: Channel bus-off entry not detected 1: Channel bus-off entry detected	R/W
4	BORF	Bus-Off Recovery Flag 0: Channel bus-off recovery not detected 1: Channel bus-off recovery detected	R/W

Bit	Symbol	Function	R/W
5	OVLF	Overload Flag 0: Channel overload not detected 1: Channel overload detected	R/W
6	BLF	Bus Lock Flag 0: Channel bus lock not detected 1: Channel bus lock detected	R/W
7	ALF	Arbitration Lost Flag 0: Channel arbitration lost not detected 1: Channel arbitration lost detected	R/W
8	SERR	Stuff Error 0: Channel stuff error not detected 1: Channel stuff error detected	R/W
9	FERR	Form Error 0: Channel form error not detected 1: Channel form error detected	R/W
10	AERR	Acknowledge Error 0: Channel acknowledge error not detected 1: Channel acknowledge error detected	R/W
11	CERR	CRC Error 0: Channel CRC error not detected 1: Channel CRC error detected	R/W
12	B1ERR	Bit 1 Error 0: Channel bit 1 error not detected 1: Channel bit 1 error detected	R/W
13	B0ERR	Bit 0 Error 0: Channel bit 0 error not detected 1: Channel bit 0 error detected	R/W
14	ADERR	Acknowledge Delimiter Error 0: Channel acknowledge delimiter error not detected 1: Channel acknowledge delimiter error detected	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
30:16	CRCREG[14:0]	CRC Register value These bits show the CRC value calculated for the CAN2.0 CAN frame.	R
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Channel Error Flag register shows the status of various error conditions detectable regardless of the setting of the related CAN Channel Error Interrupt Enable Register. It also shows the status of the various bus errors detectable by the CAN channel. Refer to the CAN specification (ISO 11898-1) to check when each error condition occurs.

For this register, only a single bit can be cleared by software. Do not use the bit clear instruction to clear the bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

Example in assembler language to clear the CFDC0ERFL.BEF bit:

mov.b #0x0FE, CFDC0ERFL ;

BEF bit (Bus Error Flag)

The BEF bit indicates a detection of a CAN channel bus error state, flagged by bits [14:8] in this register.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when a bus error is detected, and is cleared automatically when the related CANFD channel is in CH_RESET mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

EWF bit (Error Warning Flag)

The EWF bit indicates whether an error warning condition has been detected for the CAN channel.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when either TEC or REC exceeds 0x5F.

The setting of this bit only occurs when the TEC or REC initially exceeds 0x5F. Therefore, if the TEC or REC remains > 0x5F and the EWF bit is cleared by software, it is not set again until both the TEC and REC go below 0x60 and either TEC or REC crosses over again from a value 0x5F to a value > 0x5F.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

EPF bit (Error Passive Flag)

The EPF bit indicates a detection of a CAN channel error passive state.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when the CAN error state becomes error passive state.

The setting of this bit only occurs when the TEC or REC initially exceeds 0x7F. Therefore, if the TEC or REC remains > 0x7F and the bit is cleared by software, it is not set again until both the TEC and REC go below 0x80 and either TEC or REC crosses over again from a value ≤ 0x7F to a value > 0x7F.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

BOEF bit (Bus-Off Entry Flag)

The BOEF bit indicates a detection of a CAN channel bus-off entry state.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when the CAN error state enters the bus-off state.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If a set condition occurs simultaneously with a clear condition, then the bit is set.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

BORF bit (Bus-Off Recovery Flag)

The BORF bit indicates a detection of a CAN channel bus-off recovery state.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically if CAN channel recovers from bus-off state in the following conditions:

- When CFDC0CTR.BOM is 00b and normal recovery (11 consecutive recessive bits x 128 times detected) occurs
- When CFDC0CTR.BOM is 10b and normal recovery (11 consecutive recessive bits x 128 times detected) occurs
- When CFDC0CTR.BOM is 11b and normal recovery (11 consecutive recessive bits x 128 times detected) occurs.

The bit is not set if CAN channel recovers from bus-off state in the following conditions:

- When CAN Reset mode is requested
- When CFDC0CTR.RTBO is set to 1 (the CAN channel returns to error active)
- When CFDC0CTR.BOM is 01b
- When CFDC0CTR.BOM is 11b and a halt request is asserted before the CAN channel reaches the end of the bus-off state.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If a set condition occurs simultaneously with a clear condition, the flag is set.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

OVLF bit (Overload Flag)

The OVLF flag indicates a detection of a CAN channel overload state.

The OVLF bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when an overload condition is detected. If a set condition occurs simultaneously with a clear condition, then the bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

BLF bit (Bus Lock Flag)

The BLF bit indicates a detection of a CAN channel bus lock condition.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when 32 consecutive dominant bits are detected on the CAN bus while the CAN channel is in Operation mode.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

It is cleared automatically when the related CANFD channel is in CH_RESET mode.

ALF bit (Arbitration Lost Flag)

The ALF bit indicates a detection of a CAN channel bus arbitration lost condition.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

The bit is set automatically when an arbitration lost condition is detected on the CAN bus while the CAN channel is in Operation mode.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

SERR bit (Stuff Error)

The SERR bit indicates a detection of a CAN stuff error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a stuff error is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

FERR bit (Form Error)

The FERR bit indicates a detection of a CAN form error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.

3. If yes, continue, else go back to step 1.

This bit is set automatically when a form error is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

AERR bit (Acknowledge Error)

The AERR bit indicates a detection of a CAN acknowledge error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when an acknowledge error is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION.

CERR bit (CRC Error)

The CERR bit indicates a detection of a CAN CRC error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a CRC error is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

B1ERR bit (Bit 1 Error)

The B1ERR bit indicates a detection of a recessive bit error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a recessive bit error (expected recessive bit, sampled as dominant bit) is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

B0ERR bit (Bit 0 Error)

The B0ERR bit indicates a detection of a dominant bit error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a dominant bit error (expected dominant bit, sampled as recessive bit) is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

ADERR bit (Acknowledge Delimiter Error)

The ADERR bit indicates a detection of an acknowledge delimiter bit error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a form error is detected during the acknowledge delimiter state of frame transmission. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

CRCREG[14:0] bits (CRC Register value)

The CRCREG[14:0] bits read the calculated CRC value when CFDC0CTR.CTME bit is 1 for the channel.

If CFDC0CTR.CTME bit is 0, then these bits are always read as 0.

These bits show the CAN2.0 CRC value calculated by the CANFD channel logic when the CTME bit is enabled.

The CFDC0ERFL.CRCREG value is updated in the first bit of the CRC field of the CAN frame (reception and transmission).

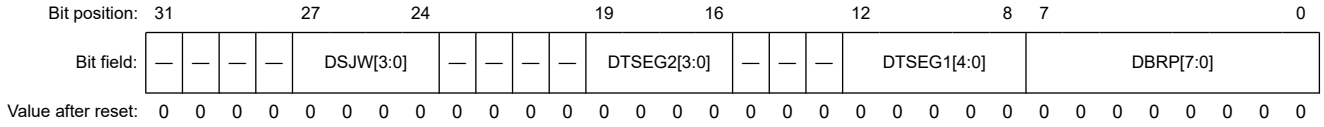
These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

34.2.6 CFDC0DCFG : Data Bitrate Configuration Register

This register is not available in the classical CAN function.

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0100



Bit	Symbol	Function	R/W
7:0	DBRP[7:0]	Channel Data Baud Rate Prescaler Data Baud Rate Prescaler division ratio	R/W
12:8	DTSEG1[4:0]	Timing Segment 1 0x00: Reserved 0x01: 2 Tq 0x02: 3 Tq 0x03: 4 Tq ⋮ 0x1E: 31 Tq 0x1F: 32 Tq	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
19:16	DTSEG2[3:0]	Timing Segment 2 0x0: Reserved 0x1: 2 Tq ⋮ 0xE: 15 Tq 0xF: 16 Tq	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
27:24	DSJW[3:0]	Resynchronization Jump Width 0x0: 1 Tq 0x1: 2 Tq ⋮ 0xF: 16 Tq	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Tq means time quantum.

The Data Bitrate Configuration Register configures the transmission/reception data baud rate parameters of the channels.

The channel of Classical CAN mode does not perform configuration of this register.

DBRP[7:0] bits (Channel Data Baud Rate Prescaler)

The DBRP[7:0] bits define the peripheral bus clock periods contained in a time quantum.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

DTSEG1[4:0] bits (Timing Segment 1)

The DTSEG1[4:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. A value from 2 to 32 time quanta can be set.

The DTSEG1[4:0] bits are also used to set the propagation segment.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not write any other value to these bits. See section 34.4.1.2. CAN Bit Timing for more details.

DTSEG2[3:0] bits (Timing Segment 2)

The DTSEG2[3:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error. A value from 2 to 16 time quanta can be set.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not write any other value to these bits.

DSJW[3:0] bits (Resynchronization Jump Width)

The DSJW[3:0] bits set the synchronization jump width. A value from 1 to 16 time quanta can be set.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

34.2.7 CFDC0FDCFG : CANFD Configuration Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0104

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	CLOE	REFE	FDOE	—	—	—	—	TDCO[7:0]							
Value after reset:	0	0/1*1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ESIC	TDCE	TDCOC	—	—	—	—	—	EOCCFG[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	EOCCFG[2:0]	Error Occurrence Counter Configuration 0 0 0: All transmitter or receiver CAN frames 0 0 1: All transmitter CAN frames 0 1 0: All receiver CAN frames 0 1 1: Reserved 1 0 0: Only transmitter or receiver CANFD data-phase (fast bits) 1 0 1: Only transmitter CANFD data-phase (fast bits) 1 1 0: Only receiver CANFD data-phase (fast bits) 1 1 1: Reserved	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	TDCOC*2	Transceiver Delay Compensation Offset Configuration 0: Measured + offset 1: Offset-only	R/W
9	TDCE*2	Transceiver Delay Compensation Enable 0: Transceiver delay compensation disabled 1: Transceiver delay compensation enabled	R/W
10	ESIC*2	Error State Indication Configuration 0: The ESI bit in the frame represents the error state of the node itself 1: The ESI bit in the frame represents the error state of the message buffer if the node itself is not in error passive. If the node is in error passive, then the ESI bit is driven by the node itself.	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
23:16	TDCO[7:0]*2	Transceiver Delay Compensation Offset	R/W
27:24	—	These bits are read as 0. The write value should be 0.	R/W
28	FDOE*2	FD-Only Enable 0: FD-only mode disabled 1: FD-only mode enabled	R/W

Bit	Symbol	Function	R/W
29	REFE	RX Edge Filter Enable 0: RX edge filter disabled 1: RX edge filter enabled	R/W
30	CLOE*2 *3	Classical CAN Enable 0: Classical CAN mode disabled 1: Classical CAN mode enabled	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The value after reset is 0 for products that support the CAN-FD protocol, and 1 for products that support only classical CAN protocol.
Note 2. These bits are not available in the classical CAN function.
Note 3. This bit can only be written for products that support the CAN-FD protocol. For products that support only classical CAN protocol, this bit is reserved and fixed to 1.

The CANFD Configuration Register configures which communication direction (transmitter/receiver) errors are counted.

EOCCFG[2:0] bits (Error Occurrence Counter Configuration)

The EOCCFG[2:0] bits select which type of CAN frame configuration and direction, including protocol errors are counted.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

TDCOC bit (Transceiver Delay Compensation Offset Configuration)*1

The TDCOC bit selects which offset is used when defining the position of the secondary sample point (SSP) for the CANFD channel. If the bit is set to 0, the position of the SSP is the measured transceiver delay plus the fixed offset. If the bit is 1, the position of the SSP is defined only by the offset.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical CAN mode.

TDCE bit (Transceiver Delay Compensation Enable)*1

The TDCE bit enables the transceiver delay compensation for the CANFD channel.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical CAN mode.

ESIC bit (Error State Indication Configuration)*1

The ESIC bit controls the transmission of either the ESI flag information or the message of ESI flag information (CFDCFFDCSTS.CFESI or CFDTMFDCTRb.TMESI).

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical CAN mode.

TDCO[7:0] bits (Transceiver Delay Compensation Offset)*1

The TDCO[7:0] bits set the secondary sample point offset. How this value is used, depends on the CFDC0FDCFG.TDCOC setting.

If CFDC0FDCFG.TDCOC = 0, the transceiver delay compensation result is equal to the Trv_Delay (measured delay) + the value in CFDC0FDCFG.TDCO, rounded down to the nearest integer number of time quanta. Otherwise, the result is equal to the value in CFDC0FDCFG.TDCO. See section 34.4.1.5. Transmitter Delay Compensation for details on how CFDC0FDCFG.TDCO is used.

The actual offset value is interpreted as TDCO + 1. For example, if 4 is set in TDCO, the offset is 5 clock cycles. Clock cycle is 1 cycle of CAN channel DLL clock.

Do not write to the TDCO[7:0] bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical CAN mode.

FDOE bit (FD-Only Enable)*1

The FDOE bit enables the reception and transmission of CANFD-only frames. If enabled, communication in Classical CAN frame format is disabled. Transmission of Classical CAN frames is not possible because the FDF bit of the message buffer is a don't care (CFDCFFDCSTS.CFFDF/CFDTMFDCTRb.TMFDF).

If messages with Classical CAN frame format are received, the protocol controller treats them as invalid frames and response with error frames. When a Classical CAN frame is configured for transmitting, the FDF bit is sent as recessive, therefore an FD frame is sent. If the data length code (DLC) is configured of greater than 8 bytes, the remaining data bytes are padded with 0xCC.

The FDOE bit cannot be written in CH_OPERATION, CH_HALT or CH_SLEEP mode.

Do not set CFDC0FDCFG.FDOE and CFDC0FDCFG.CLOE simultaneously.

REFE bit (RX Edge Filter Enable)

The REFE bit enables the RX edge filter during the IDLE detection (bus integration). When the bit is enabled, two consecutive dominant time quanta are required to detect a synchronization edge.

The REFE bit cannot be written in CH_OPERATION, CH_HALT and CH_SLEEP mode. Do not set this bit when in Classical CAN mode.

CLOE bit (Classical CAN Enable)*1

The CLOE bit enables the Classical CAN mode. If this bit is 1, the protocol controller can only send classical frames and response with a form or CRC error on FD frames.

Do not set CFDC0FDCFG.CLOE and CFDC0FDCFG.FDOE simultaneously.

CFDC0FDCFG.CLOE	CFDC0FDCFG.FDOE	Channel mode
0	0	CANFD mode
0	1	FD-only mode
1	0	Classical CAN mode
1	1	Reserved

The CANFD mode is available only for CANFD supported product.

Do not write to this bit in CH_OPERATION, CH_HALT or CH_SLEEP mode.

Only write to these bits when the CANFD channel is in CH_RESET mode.

Note 1. These bits are not available in the classical CAN function.

34.2.8 CFDC0FDCTR : CANFD Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0108

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOCCLR	EOCCLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EOCCLR	Error Occurrence Counter Clear 0: No error occurrence counter clear 1: Clear error occurrence counter	R/W
1	SOCCLR	Successful Occurrence Counter Clear 0: No successful occurrence counter clear 1: Clear successful occurrence counter	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The CANFD Control Register controls the error and successful occurrence counters.

EOCCLR bit (Error Occurrence Counter Clear)

The EOCCLR bit is used to clear the error occurrence counter.

Do not write to this bit in CH_SLEEP or CH_RESET mode. The read value is always 0.

This bit is cleared automatically by the CANFD module logic and when the related CANFD channel is in CH_RESET mode.

SOCCLR bit (Successful Occurrence Counter Clear)

The SOCCLR bit is used to clear the successful occurrence counter.

Do not write to this bit in CH_SLEEP or CH_RESET mode. The read value is always 0.

This bit is cleared automatically by the CANFD module logic and when the related CANFD channel is in CH_RESET mode.

34.2.9 CFDC0FDSTS : CANFD Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x010C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SOC[7:0]								EOC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TDCVF	—	—	—	—	—	SOCO	EOCO	TDCR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TDCR[7:0]*1	Transceiver Delay Compensation Result	R
8	EOCO	Error Occurrence Counter Overflow 0: Error occurrence counter has not overflowed 1: Error occurrence counter has overflowed	R/W
9	SOCO	Successful Occurrence Counter Overflow 0: Successful occurrence counter has not overflowed 1: Successful occurrence counter has overflowed	R/W
14:10	—	These bits are read as 0. The write value should be 0.	R/W
15	TDCVF*1	Transceiver Delay Compensation Violation Flag 0: Transceiver delay compensation violation has not occurred 1: Transceiver delay compensation violation has occurred	R/W
23:16	EOC[7:0]	Error Occurrence Counter These bits show the error occurrence counter value.	R

Bit	Symbol	Function	R/W
31:24	SOC[7:0]	Successful occurrence counter These bits show the successful occurrence counter value.	R

Note: S-TYPE-3, P-TYPE-3
Note 1. These bits are not available in the classical CAN function.

The CANFD Status Register indicates the transceiver compensation delay result and its related FIFO message lost status.

TDCR[7:0] bits (Transceiver Delay Compensation Result)

The TDCR[7:0] bits are set when the transceiver delay has been measured.

The measured delay is a multiple of the CAN channel DLL clock. The result depends on the CFDC0FDCFG.TDCOC configuration and the offset value in CFDC0FDCFG.TDCO. See section 34.4.1.5. Transmitter Delay Compensation for details on how this value is derived.

The TDCR[7:0] bits are updated at the falling edge between FDF and the RES bit when CFDC0FDCFG.TDCOC = 0 and the transceiver delay compensation is enabled (CFDC0FDCFG.TDCE = 1).

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

Note: These bits are not available in the classical CAN function.

EOCO bit (Error Occurrence Counter Overflow)

The EOCO bit indicates whether the related CAN channel error occurrence counter has overflowed. This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when CFDC0FDSTS.EOC is 0xFF and a CAN bus error is detected based on the configuration defined in CFDC0FDCFG.EOCCFG.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

SOCO bit (Successful Occurrence Counter Overflow)

The SOCO bit indicates whether the related CAN channel successful occurrence counter has overflowed. This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when CFDC0FDSTS.SOC is 0xFF and a successful message reception or successful message transmission occurs.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Write to this bit only when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

TDCVF bit (Transceiver Delay Compensation Violation Flag)

The CANFD module captures internally the transmitted data bit-by-bit. This data is then compared against the received CAN bus level which is delayed by the transceiver loop delay.

The transceiver delay has some variations depending on the physical parameters such as temperature. The result bit CFDC0FDSTS.TDCR is updated by each message. However, temporary maximum delay violation could be missed. Therefore, the TDCVF bit captures this violation.

This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when the transceiver delay compensation is greater than the maximum delay compensation (6 data bit times - 2 clk_dlc) and the internal bit is overrun.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

Note: This bit is not available in the classical CAN function.

EOC[7:0] bits (Error Occurrence Counter)

The EOC[7:0] bits are used together with the SOC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.

This higher error rate can be detected depending on the configuration of the CFDC0FDCFG.EOCCFG bits.

The EOC[7:0] bits are set only by CANFD module logic. These bits are cleared by writing 1 to CFDC0FDCTR.EOCCLR. Writing any other value has no effect.

These bits are updated when an error occurs, according to the configuration of the CFDC0FDCFG.EOCCFG bits. When the counter reaches the value of 0xFF, the update stops.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

SOC[7:0] bits (Successful occurrence counter)

The SOC[7:0] bits are used together with the EOC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.

The SOC[7:0] bits are set only by CANFD module logic. Writing any other value has no effect.

These bits are updated when the occurrence of any error-free messages on the bus is detected through reception or transmission. When the counter reaches the value of 0xFF, the update stops.

Note: In Loopback mode, the counter is incremented twice.

These bits are cleared by writing 1 to CFDC0FDCTR.SOCCLR.

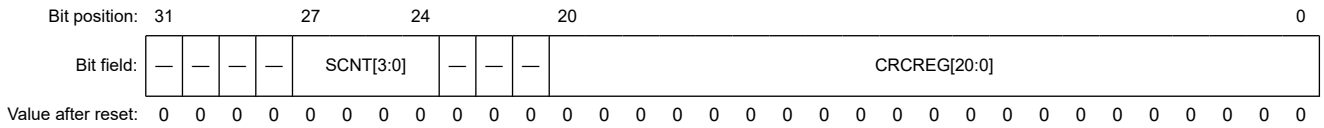
These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

34.2.10 CFDC0FDCRC : CANFD CRC Register

This register is not available in the classical CAN function.

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0110



Bit	Symbol	Function	R/W
20:0	CRCREG[20:0]	CRC Register value These bits show the CRC value calculated for the CANFD frame.	R
23:21	—	These bits are read as 0.	R
27:24	SCNT[3:0]	Stuff bit count These bits shows the stuff bit count (mod 8) for the CANFD frame.	R
31:28	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The CANFD CRC Register holds the CRC value calculated for the CANFD frame.

CRCREG[20:0] bits (CRC Register value)

The CRCREG[20:0] bits contain the CRC value calculated by the CANFD channel logic when the CFDC0CTR.CTME bit is enabled.

The CFDC0FDCRC.CRCREG value is updated in the first bit of the CRC field of the CANFD frame (reception and transmission).

When the CFDC0CTR.CTME bit is 0, the CRCREG[20:0] bits are always read as 0.

When bit 17th of the CRC field is used, CRCREG[20:17] are always read as 0.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

SCNT[3:0] bits (Stuff bit count)

The SCNT[3:0] bits contain the stuff count value of the CANFD frame. These bits indicate the number of inserted stuff bits (modulo 8, Graycoded) for a CANFD frame when the CFDC0CTR.CTME bit is enabled in CFDC0FDCRC.SCNT[3:1]. SCNT[0] is the parity bit.

When the CFDC0CTR.CTME bit is 0, the SCNT[3:0] bits are always read as 0.

The SCNT value is updated in the first bit of CRC field of the CANFD frame (reception and transmission).

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

34.2.11 CFDGCFG : Global Configuration Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ITRCP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TSSS	TSP[3:0]				—	—	CMPOC	DCS	MME	DRE	DCE	TPRI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TPRI	Transmission Priority 0: ID priority 1: Message buffer number priority	R/W
1	DCE	DLC Check Enable 0: DLC check disabled 1: DLC check enabled	R/W
2	DRE	DLC Replacement Enable 0: DLC replacement disabled 1: DLC replacement enabled	R/W
3	MME	Mirror Mode Enable 0: Mirror mode disabled 1: Mirror mode enabled	R/W
4	DCS	Data Link Controller Clock Select 0: CANFD core clock (CANFDCLK) 1: External oscillator clock (CANMCLK)	R/W
5	CMPOC*1	CANFD Message Payload Overflow Configuration 0: Message is rejected 1: Message payload is cut to fit to configured message size	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
11:8	TSP[3:0]	Timestamp Prescaler 0x0: Timestamp prescaler = 1 0x1: Timestamp prescaler = 2 0x2: Timestamp prescaler = 4 0x3: Timestamp prescaler = 8 ⋮ 0xD: Timestamp prescaler = 8192 0xE: Timestamp prescaler = 16384 0xF: Timestamp prescaler = 32768	R/W
12	TSSS	Timestamp Source Select 0: Source clock for timestamp counter is peripheral clock 1: Source clock for timestamp counter is bit time clock	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
31:16	ITRCP[15:0]	Interval Timer Reference Clock Prescaler FIFO interval timer prescaler value	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is not available in the classical CAN function.

The Global Configuration Register is used to select the transmission priority to be used for all the TX message buffers and the clock source for the CAN protocol engine of CAN channel. The CFDGCFG register is also used to select the source for the timestamp clock and to configure the frequency for the timestamp clock and interval timer reference clock.

TPRI bit (Transmission Priority)

The TPRI bit selects the transmission priority for CAN channel.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode.

Message buffer number priority should not be used together with TX queue transmission.

DCE bit (DLC Check Enable)

The DCE bit enables data length code (DLC) check for CAN channel.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode.

DRE bit (DLC Replacement Enable)

When the DRE bit is 1 and the DCE is 1, the CANFD stores the configured value (CFDGAFLP0r.GAFLDLC) of the DLC in the destination RX message buffer or FIFO buffer if the DLC check passes. Otherwise, the DLC value in the destination RX message buffer or FIFO buffer is unchanged.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode.

MME bit (Mirror Mode Enable)

The MME bit enables the Mirror mode for CAN channel.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode.

DCS bit (Data Link Controller Clock Select)

The DCS bit selects CANFDCLK or CANMCLK as the clock source for CAN communication.

Do not write to this bit in GL_SLEEP or GL_OPERATION mode. Only write to this bit when CANFD module is in GL_RESET mode.

CMPOC bit (CANFD Message Payload Overflow Configuration)

The CMPOC bit controls the message payload acceptance mechanism when the received payload is higher than the message buffer payload size CFDRMNB.RMPLS, CFDRFCCa.RFPLS, and CFDCFCC.CFPLS. The received message payload is always compared with the available message payload size in the message buffer.

Do not write to this bit in GL_SLEEP or GL_OPERATION mode. Only write to this bit when CANFD module is in GL_RESET mode.

When this bit is set and payload overflow occurs, the DLC value is stored in the RX message buffer or FIFO buffer unchanged.

Note: This bit is not available in the classical CAN function.

TSP[3:0] bits (Timestamp Prescaler)

The value configured in the TSP[3:0] bits defines the period of the clock source used for the timestamp counter.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode.

TSSS bit (Timestamp Source Select)

The TSSS bit allows the selection of the clock source for the timestamp counter.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode. Additionally, do not set this bit to 1 when CANFD communication is used.*1

Note: The bit time clock varies depending on the nominal and data rate bit configuration.

Note 1. This feature is not available in the classical CAN function.

ITRCP[15:0] bits (Interval Timer Reference Clock Prescaler)

The ITRCP[15:0] bits allow the definition of a reference clock for the FIFO interval timer source clock.

When these bits are 0x0000, the timer is disabled.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode.

34.2.12 CFDGCTR : Global Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CMPOFIE	THLEIE	MEIE	DEIE	—	—	—	—	—	GSLPR	GMDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
1:0	GMDC[1:0]	Global Mode Control 0 0: Global operation mode request 0 1: Global reset mode request 1 0: Global halt mode request 1 1: Keep current value	R/W
2	GSLPR	Global Sleep Request 0: Global sleep request disabled 1: Global sleep request enabled	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	DEIE	DLC Check Interrupt Enable 0: DLC check interrupt disabled 1: DLC check interrupt enabled	R/W
9	MEIE	Message Lost Error Interrupt Enable 0: Message lost error interrupt disabled 1: Message lost error interrupt enabled	R/W
10	THLEIE	TX History List Entry Lost Interrupt Enable 0: TX history list entry lost interrupt disabled 1: TX history list entry lost interrupt enabled	R/W

Bit	Symbol	Function	R/W
11	CMPOFIE*1	CANFD Message Payload Overflow Flag Interrupt Enable 0: CANFD message payload overflow flag interrupt disabled 1: CANFD message payload overflow flag interrupt enabled	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	TSRST	Timestamp Reset 0: Timestamp not reset 1: Timestamp reset	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is not available in the classical CAN function.

The Global Control Register controls the global mode of the CANFD module and the timestamp function. The register also enables and disables the global error interrupts.

GMDC bits (Global Mode Control)

The GMDC bits can be used to configure the modes for the CANFD module. Additionally, if CFDGCTR.GSLPR bit is 1 when the CANFD module is in Reset mode, the CANFD module enters Global Sleep mode.

Setting the GMDC bits to 11b has no effect. Mode transition is described in detail in section 34.3.2. Global Modes.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

GSLPR bit (Global Sleep Request)

The GSLPR bit globally selects the sleep request for CANFD module including CAN channels. Channel sleep request is set automatically for channels.

Only write to this bit when the CANFD module is in GL_RESET or GL_SLEEP mode.

DEIE bit (DLC Check Interrupt Enable)

When the DEIE bit is 1, an interrupt is generated if a DLC error is detected in the received frames.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

MEIE bit (Message Lost Error Interrupt Enable)

When the MEIE bit is 1, an interrupt is generated if a message lost condition occurs.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

THLEIE bit (TX History List Entry Lost Interrupt Enable)

When the THLEIE bit is 1, an interrupt is generated if a TX history list entry lost condition occurs.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

CMPOFIE bit (CANFD Message Payload Overflow Flag Interrupt Enable)

When the CMPOFIE bit is 1, an interrupt is generated when a CANFD message payload overflow condition occurs.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

Note: This bit is not available in the classical CAN function

TSRST bit (Timestamp Reset)

When the TSRST bit is 1, the Global Timestamp Register is reset to 0x0000.

Do not write to this bit when the CANFD module is in GL_SLEEP or GL_RESET mode.

Read value is always 0.

This bit is cleared automatically by the CANFD module logic.

34.2.13 CFDGSTS : Global Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x001C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	GRAM INIT	GSLP STS	GHLT STS	GRST STS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

Bit	Symbol	Function	R/W
0	GRSTSTS	Global Reset Status 0: Not in Reset mode 1: In Reset mode	R
1	GHLTSTS	Global Halt Status 0: Not in Halt mode 1: In Halt mode	R
2	GSLPSTS	Global Sleep Status 0: Not in Sleep mode 1: In Sleep mode	R
3	GRAMINIT	Global RAM Initialization 0: RAM initialization is complete 1: RAM initialization is ongoing	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The Global Status Register indicates the global status of the CANFD module.

GRSTSTS bit (Global Reset Status)

The GRSTSTS bit indicates the status of Global CANFD module Reset mode.

This bit is set automatically when the CANFD module enters GL_RESET mode. When the mode changes from GL_RESET mode to GL_SLEEP mode, this bit remains set.

This bit is cleared automatically when the CANFD module exits the GL_RESET mode.

GHLTSTS bit (Global Halt Status)

The GHLTSTS bit indicates the status of Global CANFD module Halt mode.

This bit is set automatically when the CANFD module enters GL_HALT mode.

This bit is cleared automatically when the CANFD module exits the GL_HALT mode.

GSLPSTS bit (Global Sleep Status)

The GSLPSTS bit indicates the status of Global CANFD module Sleep mode.

This bit is set automatically when the CANFD module enters GL_SLEEP mode.

This bit is cleared automatically when the CANFD module exits the GL_SLEEP mode.

GRAMINIT bit (Global RAM Initialization)

The GRAMINIT bit indicates the status of Global CANFD module RAM initialization.

This bit is set automatically when the CANFD module enters GL_SLEEP mode after a hardware reset.

This bit is cleared automatically when the CANFD module completed RAM initialization.

This bit is cleared when the test_mode input port is set to 1.

34.2.14 CFDGERFL : Global Error Flag Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—		EEF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	CMPOF	THLES	MES	DEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DEF	DLC Error Flag 0: DLC error not detected 1: DLC error detected	R/W
1	MES	Message Lost Error Status 0: Message lost error not detected 1: Message lost error detected	R
2	THLES	TX History List Entry Lost Error Status 0: TX history list entry lost error not detected 1: TX history list entry lost error detected	R
3	CMPOF*1	CANFD Message Payload Overflow Flag 0: CANFD message payload overflow not detected 1: CANFD message payload overflow detected	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
16	EEF0	ECC Error Flag 0: ECC error not detected during TX-SCAN 1: ECC error detected during TX-SCAN	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is not available in the classical CAN function.

The Global Error Flag register indicates the detection of global errors.

DEF bit (DLC Error Flag)

The DEF bit indicates the error status of the DLC.

Do not write to this bit when the CANFD module is in GL_SLEEP or GL_RESET mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when a DLC error is detected in a received frame.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set

The bit is cleared by writing 0 to it.

This bit is cleared automatically in GL_RESET mode.

MES bit (Message Lost Error Status)

The MES bit indicates status of the message lost error.

This bit is set automatically when a FIFO message lost error is detected.

This bit is cleared automatically when:

- All FIFO message lost flags are cleared
- The CANFD module is in GL_RESET mode.

THLES bit (TX History List Entry Lost Error Status)

The THLES bit indicates status of the TX history list entry lost error.

This bit is set automatically when a TX history list entry lost error is detected.

This bit is cleared automatically when:

- All TX history list entry lost flags are cleared
- The CANFD module is in GL_RESET mode.

CMPOF bit (CANFD Message Payload Overflow Flag)

The CMPOF bit is set automatically when a CANFD message payload overflow is detected on at least one channel.

Do not write to this bit when the CANFD module is in GL_SLEEP or GL_RESET mode.

This bit is cleared by writing 0 to it. Writing 1 to this bit has no effect.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is cleared automatically in GL_RESET mode.

Note: This bit is not available in the classical CAN function

EEF0 bit (ECC Error Flag)

The EEF0 bit specifies whether an ECC error has occurred.

Do not write to this bit when the CANFD module is in GL_SLEEP or GL_RESET mode. Writing 1 to this bit has no effect.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

The bit is cleared by writing 0 to it. This bit is cleared automatically in GL_RESET mode.

34.2.15 CFDGTINTSTS : Global TX Interrupt Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00A4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	THIF0	CFTIF0	TQIF0	TAI0	TSIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSIF0	TX Successful Interrupt Flag 0: Channel n TX Successful Interrupt flag not set 1: Channel n TX Successful Interrupt flag set	R
1	TAI0	TX Abort Interrupt Flag 0: Channel n TX Abort Interrupt flag not set 1: Channel n TX Abort Interrupt flag set	R

Bit	Symbol	Function	R/W
2	TQIF0	TX Queue Interrupt Flag 0: Channel n TX Queue Interrupt flag not set 1: Channel n TX Queue Interrupt flag set	R
3	CFTIF0	COM FIFO TX Mode Interrupt Flag 0: Channel n COM FIFO TX Mode Interrupt flag not set 1: Channel n COM FIFO TX Mode Interrupt flag set	R
4	THIF0	TX History List Interrupt 0: Channel n TX History List Interrupt flag not set 1: Channel n TX History List Interrupt flag set	R
31:5	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The Global TX Interrupt Status register indicates the detection of transmit specific interrupts.

TSIF0 bit (TX Successful Interrupt Flag)

The TSIF0 bit is set to 1 when the TX Successful Interrupt flag of the related channel is set (when the interrupt is enabled). This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

TAI0 bit (TX Abort Interrupt Flag)

The TAIF0 bit is set to 1 when the TX Abort Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

TQIF0 bit (TX Queue Interrupt Flag)

The TQIF0 bit is set to 1 when the TX Queue Interrupt flag of the related channel is set (when the interrupt is enabled). This bit is cleared automatically:

- When the related TX Queue Interrupt flag is cleared (when the interrupt is enable disabled)
- When in GL_RESET or CH_RESET mode.

CFTIF0 bit (COM FIFO TX Mode Interrupt Flag)

The CFTIF0 bit is set to 1 when the related COM TX FIFO Mode Interrupt flag (CFDCFSTS.CFTXIF) is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related COM TX FIFO Mode Interrupt flag (CFDCFSTS.CFTXIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

THIF0 bit (TX History List Interrupt)

The THIF0 bit is set to 1 when the related TX History List Interrupt flag (CFDTHLSTS.THLIF) is set (when the interrupt is enabled).

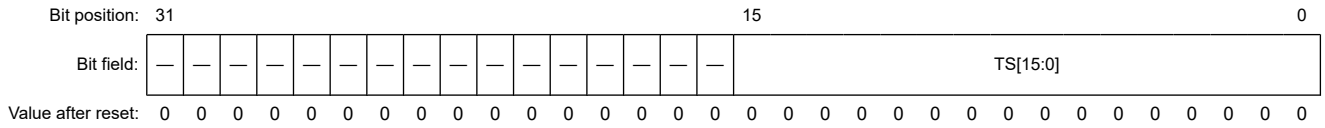
This bit is cleared automatically:

- When the related TX History List Interrupt flag (CFDTHLSTS.THLIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

34.2.16 CFDGTSC : Global Timestamp Counter Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0024



Bit	Symbol	Function	R/W
15:0	TS[15:0]	Timestamp value	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The Global Timestamp Counter register stores the timestamp based on the selected configuration.

TS[15:0] bits (Timestamp value)

The Timestamp value is stored in the Global Timestamp Counter register based on the configuration of TSSS, TSBTCS and TSP. The accuracy of the timestamp counter cannot be guaranteed when transitioning to halt state.

The Timestamp value is stored in this register based on the configuration of TSSS, TSBTCS and TSP.

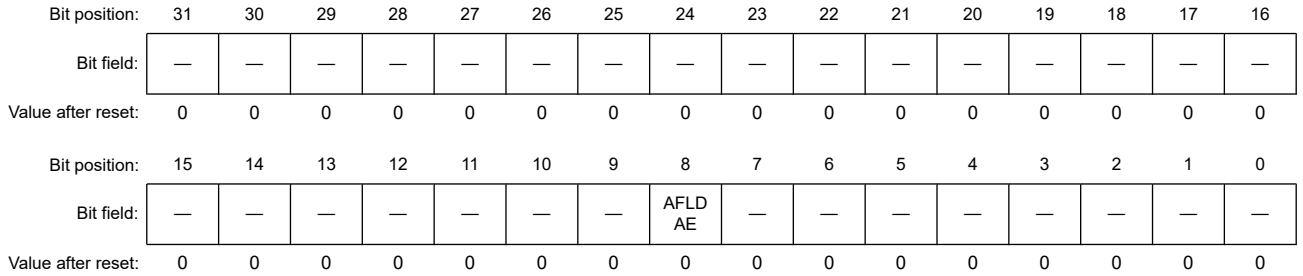
Do not write to bits TS[15:0] when the CANFD module is in GL_RESET or GL_SLEEP mode.

The TS[15:0] bits are cleared automatically in GL_RESET mode.

34.2.17 CFDGAFLECTR : Global Acceptance Filter List Entry Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0028



Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	AFLDAE	Acceptance Filter List Data Access Enable 0: Acceptance Filter List data access disabled 1: Acceptance Filter List data access enabled	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Acceptance Filter List Entry Control Register is used to select the Global Acceptance Filter List page for reading or writing entries into the Global Acceptance Filter List.

AFLDAE bit (Acceptance Filter List Data Access Enable)

The AFLDAE bit prevents write access to the Acceptance Filter List when cleared after configuration of the Acceptance Filter List.

Data can be read from the Acceptance Filter List independent of the status of this bit.

Do not write to this bit when the CANFD module is in GL_SLEEP mode. Set this bit to enable write access for the Acceptance Filter List.

34.2.18 CFDGAFLCFG : Global Acceptance Filter List Configuration Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x002C

Bit position:	31											20			16														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	RNC0[4:0]				—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
20:16	RNC0[4:0]	Rule Number Number of rules dedicated	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Acceptance Filter List Configuration Register is used to define the number of rules for entries in the Acceptance Filter List.

The total number of available entries in the Acceptance Filter List is 16.

RNC0[4:0] bits (Rule Number)

The RNC0[4:0] bits define the number of rules in the Acceptance Filter List.

Only write to these bits when the CANFD module is in GL_RESET mode. These bits can set to 5 bits for 16 rules.

34.2.19 CFDGAFLIDr : Global Acceptance Filter List ID Registers (r = 1 to 16)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0120 + 0x0010 × (r - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	GAFLIDE	GAFLRTR	GAFLLB	GAFLID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GAFLID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
28:0	GAFLID[28:0]	Global Acceptance Filter List Entry ID Field ID part of the Global Acceptance Filter List entry	R/W
29	GAFLLB	Global Acceptance Filter List Entry Loopback Configuration 0: Global Acceptance Filter List entry ID for acceptance filtering with attribute RX 1: Global Acceptance Filter List entry ID for acceptance filtering with attribute TX	R/W
30	GAFLRTR	Global Acceptance Filter List Entry RTR Field 0: Data frame 1: Remote frame	R/W

Bit	Symbol	Function	R/W
31	GAFLIDE	Global Acceptance Filter List Entry IDE Field 0: Standard identifier of rule entry ID is valid for acceptance filtering 1: Extended identifier of rule entry ID is valid for acceptance filtering	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Acceptance Filter List ID Registers are used to configure the ID field for the rules of entries in the Global Acceptance Filter List.

GAFLID[28:0] bits (Global Acceptance Filter List Entry ID Field)

The GAFLID[28:0] bits represent the CAN identifier (ID) field of each entry in the Global Acceptance Filter List.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLLB bit (Global Acceptance Filter List Entry Loopback Configuration)

The GAFLLB bit selects whether entry in the Global Acceptance Filter List gets the attribute RX or TX.

This attribute determines the validity of the entry in Mirror mode, Loopback test mode, and during standard (non-loopback) reception. See section 34.5.5. Loopback Modes for detailed description of the validity of the Global Acceptance Filter List entry depending on transmitter/receiver case, the type of loopback mode, and RX/TX attribute.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLRTR bit (Global Acceptance Filter List Entry RTR Field)

The GAFLRTR bit allows the configuration of the specified frame format (data frame or remote frame) for each entry of the Global Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the RTR bit of the received CAN message.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLIDE bit (Global Acceptance Filter List Entry IDE Field)

The GAFLIDE bit allows the configuration of the ID format (standard ID or extended ID) for each entry in the Global Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the IDE bit of the received CAN message.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

34.2.20 CFDGAFLMr : Global Acceptance Filter List Mask Registers (r = 1 to 16)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0124 + 0x0010 × (r - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	GAFLIDEM	GAFLRTRM	GAFLIFL1	GAFLIDM[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GAFLIDM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
28:0	GAFLIDM[28:0]	Global Acceptance Filter List ID Mask Field Global Acceptance Filter List Mask field bits for ID field	R/W
29	GAFLIFL1	Global Acceptance Filter List Information Label 1 Global Acceptance Filter List information label bit 1	R/W
30	GAFLRTRM	Global Acceptance Filter List Entry RTR Mask 0: RTR bit is not used for ID matching 1: RTR bit is used for ID matching	R/W
31	GAFLIDEM	Global Acceptance Filter List IDE Mask 0: IDE bit is not used for ID matching 1: IDE bit is used for ID matching	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Acceptance Filter List Mask Registers are used to configure the Mask field of each rule for entries in the Global Acceptance Filter List.

GAFLIDM[28:0] bits (Global Acceptance Filter List ID Mask Field)

GAFLIDM[28:0] bits are the filter mask bits for the related bits in the CAN Identifier field of each Global Acceptance Filter List entry.

0	Corresponding STD-ID/EXT-ID bit is not used for ID matching
1	Corresponding STD-ID/EXT-ID bit is used for ID matching

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLIFL1 bit (Global Acceptance Filter List Information Label 1)

The GAFLIFL1 bit allows the configuration of a 2-bit information label to be attached to a received message accepted by the associated entry in the Global Acceptance Filter List. This bit is a MSB bit of an information label.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

This bit is stored in the Information Label Field [1] (CFDRMFDSTSb.RMIFL [1], CFDRFFDSTSb.RFIFL [1], CFDCFFDCSTS.CFIFL [1]) of the storage location of an incoming message.

GAFLRTRM bit (Global Acceptance Filter List Entry RTR Mask)

The GAFLRTRM bit allows the configuration of the RTR mask bit for each entry in the Global Acceptance Filter List.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLIDEM bit (Global Acceptance Filter List IDE Mask)

The GAFLIDEM bit allows the configuration of the IDE mask bit for each entry in the Global Acceptance Filter List.

When the IDE mask bit is 0, the ID comparison depends on the received IDE bit.

If the received IDE bit is 0, the STD-ID comparison takes place.

If the received IDE bit is 1, the EXT-ID comparison takes place.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

34.2.21 CFDGAFLP0r : Global Acceptance Filter List Pointer 0 Registers (r = 1 to 16)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0128 + 0x0010 × (r - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	GAFLPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GAFLRMV	—	—	GAFLRMDP[4:0]					GAFLIFL0	—	—	—	GAFLDLC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	GAFLDLC[3:0]	Global Acceptance Filter List DLC Field Minimum number of data bytes in a data frame required for acceptance	R/W
6:4	—	These bits are read as 0. The write value should be 0.	R/W
7	GAFLIFL0	Global Acceptance Filter List Information Label 0	R/W
12:8	GAFLRMDP[4:0]	Global Acceptance Filter List RX Message Buffer Direction Pointer RX message buffer number for storage of received messages	R/W
14:13	—	These bits are read as 0. The write value should be 0.	R/W
15	GAFLRMV	Global Acceptance Filter List RX Message Buffer Valid 0: Single message buffer direction pointer is invalid 1: Single message buffer direction pointer is valid	R/W
31:16	GAFLPTR[15:0]	Global Acceptance Filter List Pointer	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Acceptance Filter List Pointer 0 Registers are used to configure the data length code (DLC), software pointer, single message buffer select, and message buffer direction pointer for each rule entry in the Global Acceptance Filter List.

GAFLDLC[3:0] bits (Global Acceptance Filter List DLC Field)

The GAFLDLC[3:0] bits allow the configuration of a minimum data length code (DLC) value for a message to be accepted by the associated entry in the Global Acceptance Filter List (automatic DLC filter function).

DLC filter process is only passed if the DLC value of the message accepted by an entry in the Global Acceptance Filter List is equal to or higher than the DLC value configured for this associated Global Acceptance Filter List entry. Automatic DLC filter function is disabled for the corresponding rule entry when this field is set to 0.

Table 34.3 shows DLC value that can be configured.

Table 34.3 Configuration of DLC value (1 of 2)

Format	DLC[3]	DLC[2]	DLC[1]	DLC[0]	Description
CAN and CANFD	0	0	0	0	DLC of received message = 0 or more (DLC filter check is disabled)
CAN and CANFD	0	0	0	1	DLC of received message = 1 or more
CAN and CANFD	0	0	1	0	DLC of received message = 2 or more
CAN and CANFD	0	0	1	1	DLC of received message = 3 or more
CAN and CANFD	0	1	0	0	DLC of received message = 4 or more
CAN and CANFD	0	1	0	1	DLC of received message = 5 or more

Table 34.3 Configuration of DLC value (2 of 2)

Format	DLC[3]	DLC[2]	DLC[1]	DLC[0]	Description
CAN and CANFD	0	1	1	0	DLC of received message = 6 or more
CAN and CANFD	0	1	1	1	DLC of received message = 7 or more
CAN	1	x	x	x	DLC of received message = 8 or more
CANFD	1	0	0	0	DLC of received message = 8 or more*1
CANFD	1	0	0	1	DLC of received message = 12 or more*1
CANFD	1	0	1	0	DLC of received message = 16 or more*1
CANFD	1	0	1	1	DLC of received message = 20 or more*1
CANFD	1	1	0	0	DLC of received message = 24 or more*1
CANFD	1	1	0	1	DLC of received message = 32 or more*1
CANFD	1	1	1	0	DLC of received message = 48 or more*1
CANFD	1	1	1	1	DLC of received message = 64*1

Note 1. This setting is not available in the classical CAN function.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLIFL0 bit (Global Acceptance Filter List Information Label 0)

The GAFLIFL0 bit allows the configuration of a 2-bit information label that can be attached to a received message accepted by the related Global Acceptance Filter List entry. This bit is a LSB bit of an information label.

You cannot write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to the bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

This bit is stored in Information Label Field[0] (CFDRMFDSTSb.RMIFL[0], CFDRFFDSTSb.RFIFL[0], CFDCFFDCSTS.CFIFL[0]) of the storage location of an incoming message.

GAFLRMDP[4:0] bits (Global Acceptance Filter List RX Message Buffer Direction Pointer)

The GAFLRMDP[4:0] bits allow the configuration of a single reception message buffer as the destination target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry. The value entered is the single destination message buffer number.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

CFDRMNB.NRXMB[4:0] is the value entered in the RX Message Buffer Number Register to configure the number of RX message buffers. The value to be entered in CFDGAFLP0r.GAFLRMDP[4:0] bits should only be between 0x00 and CFDMNB.NMB[4:0] to 1 less.

If CFDRMNB.NRXMB[4:0] = 0x00, the GAFLRMV bit should be configured as 0.

GAFLRMV bit (Global Acceptance Filter List RX Message Buffer Valid)

The GAFLRMV bit allows the enabling or disabling of a single reception message buffer as the target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLPTR[15:0] bits (Global Acceptance Filter List Pointer)

The GAFLPTR[15:0] bits allow the configuration of a 16-bit pointer to be attached to a received message accepted by the related Global Acceptance Filter List entry. The pointer is added during message storage in the Message Buffer area and can be used by the application as a support function. The pointer information can be used for example, to support PDU Identifier allocation for the received message in AUTOSAR systems.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

34.2.22 CFDGAFLP1r : Global Acceptance Filter List Pointer 1 Registers (r = 1 to 16)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x012C + 0x0010 × (r - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	GAFL FDP8	—	—	—	—	—	—	GAFL FDP1	GAFL FDP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	GAFLFDP0	Global Acceptance Filter List FIFO Direction Pointer FIFO direction pointer bits for received message storage 0: Disable RX FIFO 0 as target for reception 1: Enable RX FIFO 0 as target for reception	R/W
1	GAFLFDP1	Global Acceptance Filter List FIFO Direction Pointer FIFO direction pointer bits for received message storage 0: Disable RX FIFO 1 as target for reception 1: Enable RX FIFO 1 as target for reception	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	GAFLFDP8	Global Acceptance Filter List FIFO Direction Pointer FIFO direction pointer bits for received message storage 0: Disable Common FIFO as target for reception 1: Enable Common FIFO as target for reception	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Acceptance Filter List Pointer 1 registers are used to configure the FIFO direction pointer fields in each Rule Entry of the Global Acceptance Filter List.

GAFLFDP8, GAFLFDP1, GAFLFDP0 bits (Global Acceptance Filter List FIFO Direction Pointer)

These bits allow the configuration of FIFO Buffers as the target for a received message passing the acceptance check of the related Global Acceptance Filter List entry. Each bit of the GAFLFDP8, GAFLFDP1, GAFLFDP0 is configuring a dedicated FIFO.

Users cannot write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

For storage in Common FIFO, target for reception can only be those Common FIFO Buffers that are configured as RX FIFO.

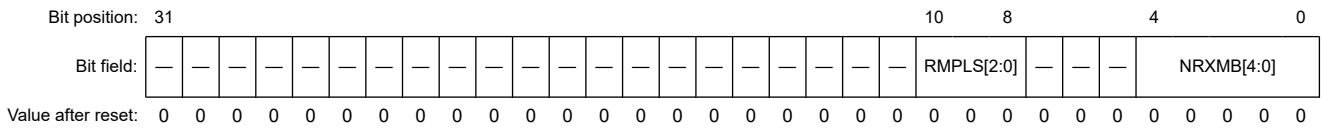
Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

Users should only configure up to 2 destination FIFO Buffers or 1 destination FIFO Buffers plus one RX Message Buffer.

34.2.23 CFDRMNB : RX Message Buffer Number Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0030



Bit	Symbol	Function	R/W
4:0	NRXMB[4:0]	Number of RX Message Buffers	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
10:8	RMPLS[2:0]	Reception Message Buffer Payload Data Size 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The RX Message Buffer Number register is used to configure the total number of RX message buffers allocated to channels.

NRXMB[4:0] bits (Number of RX Message Buffers)

The NRXMB[4:0] bits are used to configure the number of RX message buffers.

Only write to these bits when the CANFD module is in GL_RESET mode.

Enter only values between 0 and 16 inclusive, with 0x00 indicating that no RX message buffer is allocated.

RMPLS[2:0] bits (Reception Message Buffer Payload Data Size)

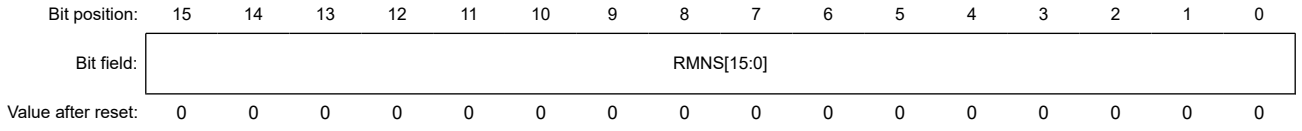
The RMPLS[2:0] bits are used to configure the message buffer payload data size.

Only write to these bits when the CANFD module is in GL_RESET mode.

34.2.24 CFDRMND : RX Message Buffer New Data Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0034



Bit	Symbol	Function	R/W
15:0	RMNS[15:0]	RX Message Buffer New Data Status 0: New data not stored in corresponding RX message buffer 1: New data stored in corresponding RX message buffer	R/W

Note: S-TYPE-3, P-TYPE-3

The RX Message Buffer New Data Status Register specifies the new data storage status of the RX message buffers.

RMNS[15:0] bits (RX Message Buffer New Data Status)

The RMNS[15:0] bits indicate the status of new data for the corresponding RX message buffer. RMNS bit [0] corresponds to RX message buffer [0] and so on.

The bit position of CFDRMND corresponds to the buffer number of RXMB.

Do not write to these bits when the CANFD module is in GL_RESET or GL_SLEEP mode. Writing 1 has no effect.

These bits cannot be cleared when message storage in the corresponding RX message buffer is in progress.
Do not use the bit clear instruction to clear these bits. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

These bits are set automatically when storage of new messages are in the corresponding RX message buffer. These bits are cleared by writing 0. These bits are cleared automatically when the CANFD module is in GL_RESET mode.

When CFDRMNB.RMPLS = 000b (maximum 8 bytes payload), the duration of message storage is 6 PCLKA cycles.

When CFDRMNB.RMPLS > 000b, the duration of message storage is 6 PCLKA cycles + 1 for each 4 bytes (maximum of 20 PCLKA cycles for 64 bytes).

Note: This feature is not available in the classical CAN function.

34.2.25 CFDRFCCa : RX FIFO Configuration/Control Registers a (a = 0 to 1)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x003C + 0x04 × a

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RFIGCV[2:0]			RFIM	—	RFDC[2:0]			—	RFPLS[2:0]			—	—	RFIE	RFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFE	RX FIFO Enable 0: FIFO disabled 1: FIFO enabled	R/W
1	RFIE	RX FIFO Interrupt Enable 0: FIFO interrupt generation disabled 1: FIFO interrupt generation enabled	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
6:4	RFPLS[2:0]*1	Rx FIFO Payload Data Size Configuration 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
10:8	RFDC[2:0]	RX FIFO Depth Configuration 0 0 0: FIFO Depth = 0 message 0 0 1: FIFO Depth = 4 messages 0 1 0: FIFO Depth = 8 messages 0 1 1: FIFO Depth = 16 messages 1 0 0: FIFO Depth = 32 messages 1 0 1: FIFO Depth = 48 messages 1 1 0: Reserved 1 1 1: Reserved	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	RFIM	RX FIFO Interrupt Mode 0: Interrupt generated when RX FIFO counter reaches RFIGCV value from values smaller than RFIGCV 1: Interrupt generated at the end of every received message storage	R/W
15:13	RFIGCV[2:0]	RX FIFO Interrupt Generation Counter Value 0 0 0: Interrupt generated when FIFO is 1/8th full 0 0 1: Interrupt generated when FIFO is 1/4th full 0 1 0: Interrupt generated when FIFO is 3/8th full 0 1 1: Interrupt generated when FIFO is 1/2 full 1 0 0: Interrupt generated when FIFO is 5/8th full 1 0 1: Interrupt generated when FIFO is 3/4th full 1 1 0: Interrupt generated when FIFO is 7/8th full 1 1 1: Interrupt generated when FIFO is full	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. These bits are not available in the classical CAN function.

The RX FIFO Configuration/Control Registers are used to configure and control the two RX FIFOs.

RFE bit (RX FIFO Enable)

The RFE bit enables the FIFO. When this bit is set to 0, the RX FIFO is cleared to empty.

Only write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode.

This bit can only be set if the configured FIFO depth is greater than 0x000 (CFDRFCCa.RFDC > 0x000) and less than 0x110.

Set the RFE bit with a separate write access to the CFDRFCCa register, after all the other bits in the CFDRFCCa register are set.

This bit is cleared automatically when the CANFD module is in GL_RESET mode.

RFIE bit (RX FIFO Interrupt Enable)

The RFIE bit enables generation of the FIFO interrupt.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

RFPLS[2:0] bits (Rx FIFO Payload Data Size Configuration)

The RFPLS[2:0] bits define the message data payload allocation in the RAM.

This is the maximum number of bytes which can be received by this FIFO.

Only write to these bits when the CANFD module is in GL_RESET mode.

Note: These bits are not available in the classical CAN function.

RFDC[2:0] bits (RX FIFO Depth Configuration)

The RFDC[2:0] bits select the depth of the FIFO in terms of the number of messages. If the FIFO depth is configured to 0 messages, the FIFO cannot be used.

Only write to these bits when the CANFD module is in GL_RESET mode.

RFIM bit (RX FIFO Interrupt Mode)

The RFIM bit selects the interrupt generation condition for the FIFO.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

Only write to this bit when the CANFD module is in GL_RESET mode.

RFIGCV[2:0] bits (RX FIFO Interrupt Generation Counter Value)

The RFIGCV[2:0] bits select the counter value of the FIFO for generation of FIFO interrupts. These values represent fractions of the FIFO depth for which an interrupt is generated.

Do not write to these bits when the CANFD module is in GL_SLEEP mode.

The setting of the RFIGCV[2:0] bits should be synchronized with the RFDC[2:0] bits.

Only write to these bits when the CANFD module is in GL_RESET mode.

34.2.26 CFDRFSTSa : RX FIFO Status Registers a (a = 0 to 1)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0044 + 0x04 × a

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	RFMC[5:0]						—	—	—	—	RFIF	RFMLT	RFFLL	RFEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	RFEMP	RX FIFO Empty 0: FIFO not empty 1: FIFO empty	R
1	RFFLL	RX FIFO Full 0: FIFO not full 1: FIFO full	R
2	RFMLT	RX FIFO Message Lost 0: No message lost in FIFO 1: FIFO message lost	R/W
3	RFIF	RX FIFO Interrupt Flag 0: FIFO interrupt condition not satisfied 1: FIFO interrupt condition satisfied	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
13:8	RFMC[5:0]	RX FIFO Message Count Number of messages stored in FIFO	R
31:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The RX FIFO Status Registers show the status of messages stored in the corresponding FIFO buffers.

RFEMP bit (RX FIFO Empty)

The RFEMP bit is set automatically when:

- The RFMC bit is 0
- RX FIFO is disabled by setting the CFDRFCCa.RFE bit to 0
- The CANFD module is in GL_RESET mode.

The RFEMP bit is cleared automatically when the first message is stored in the RX FIFO buffer.

RFFLL bit (RX FIFO Full)

The RFFLL bit is set automatically when the number of CAN messages stored in the FIFO buffer matches the configured FIFO depth.

The RFFLL is cleared automatically when:

- The number of CAN messages stored in the FIFO buffer is less than the configured FIFO depth
- RX FIFO is disabled by setting the CFDRFCCa.RFE bit to 0
- The CANFD module is in GL_RESET mode.

RFMLT bit (RX FIFO Message Lost)

Only write to the RFMLT bit when CANFD module is in GL_HALT or GL_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically whenever a message is lost due to attempted storage when the FIFO buffer is already full. If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The bit is cleared:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode.

RFIF bit (RX FIFO Interrupt Flag)

The RFIF bit is set automatically when the configured interrupt condition is satisfied. This bit is not automatically cleared when the RX FIFO buffer is disabled.

Only write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then this bit is set.

The bit is cleared by writing 0 to it. The bit is also cleared when CANFD module is in GL_RESET mode.

RFMC[5:0] bits (RX FIFO Message Count)

The RFMC[5:0] bits indicate the number of CAN messages stored in the RX FIFO buffer that can be read by the CPU.

These bits are cleared automatically when the FIFO is disabled and when the CANFD module is in GL_RESET mode.

34.2.27 CFDRFPCTRa : RX FIFO Pointer Control Registers a (a = 0 to 1)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x004C + 0x04 × a

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFPC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RFPC[7:0]	RX FIFO Pointer Control Increments read pointer of the corresponding RX FIFO buffers	W
31:8	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The RX FIFO Pointer Control Registers can be used to increment the read pointer of the corresponding RX FIFO buffers.

RFPC bits (RX FIFO Pointer Control)

When the value 0xFF is written to the RFPC bits, the pointer of the corresponding RX FIFO buffer is moved to the next FIFO entry. Only write 0xFF to these registers when the corresponding RX FIFO buffer is enabled and not empty.

The read value from these bits is always 0x00.

Only write to these bits when the CANFD module is in GL_HALT or GL_OPERATION mode.

Do not write to the RX FIFO Pointer Control registers when DMA is enabled.

34.2.28 CFDCFCC : Common FIFO Configuration/Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0054

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CFITT[7:0]								CFDC[2:0]			—	—	—	CFTML[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CFIGCV[2:0]			CFIM	CFITR	CFITSS	—	CFM	—	CFPLS[2:0]			—	CFTXIE	CFRXIE	CFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CFE	Common FIFO Enable 0: FIFO disabled 1: FIFO enabled	R/W
1	CFRXIE	Common FIFO RX Interrupt Enable 0: FIFO interrupt generation disabled for Frame RX 1: FIFO interrupt generation enabled for Frame RX	R/W
2	CFTXIE	Common FIFO TX Interrupt Enable 0: FIFO interrupt generation disabled for Frame TX 1: FIFO interrupt generation enabled for Frame TX	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	CFPLS[2:0]*1	Common FIFO Payload Data Size Configuration 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	CFM	Common FIFO Mode 0: RX FIFO mode 1: TX FIFO mode	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	CFITSS	Common FIFO Interval Timer Source Select 0: Reference clock (× 1 / × 10 period) 1: Bit time clock of related channel (FIFO is linked to fixed channel)	R/W
11	CFITR	Common FIFO Interval Timer Resolution 0: Reference clock period × 1 1: Reference clock period × 10	R/W

Bit	Symbol	Function	R/W
12	CFIM	Common FIFO Interrupt Mode 0: RX FIFO mode: RX interrupt generated when Common FIFO counter reaches CFIGCV value from a lower value TX FIFO mode: TX interrupt generated when Common FIFO transmits the last message successfully 1: RX FIFO mode: RX interrupt generated at the end of every received message storage TX FIFO mode: interrupt generated for every successfully transmitted message	R/W
15:13	CFIGCV[2:0]	Common FIFO Interrupt Generation Counter Value 0 0 0: Interrupt generated when FIFO is 1/8th full 0 0 1: Interrupt generated when FIFO is 1/4th full 0 1 0: Interrupt generated when FIFO is 3/8th full 0 1 1: Interrupt generated when FIFO is 1/2 full 1 0 0: Interrupt generated when FIFO is 5/8th full 1 0 1: Interrupt generated when FIFO is 3/4th full 1 1 0: Interrupt generated when FIFO is 7/8th full 1 1 1: Interrupt generated when FIFO is full	R/W
17:16	CFTML[1:0]	Common FIFO TX Message Buffer Link Transmission scan link position of the corresponding channel	R/W
20:18	—	These bits are read as 0. The write value should be 0.	R/W
23:21	CFDC[2:0]	Common FIFO Depth Configuration 0 0 0: FIFO Depth = 0 message 0 0 1: FIFO Depth = 4 messages 0 1 0: FIFO Depth = 8 messages 0 1 1: FIFO Depth = 16 messages 1 0 0: FIFO Depth = 32 messages 1 0 1: FIFO Depth = 48 messages 1 1 0: FIFO Depth = Reserved 1 1 1: FIFO Depth = Reserved	R/W
31:24	CFITT[7:0]	Common FIFO Interval Transmission Time Delay the start of transmission from the FIFO if configured in TX mode, delay is a multiple of basic Interval Timer Clock Source unit	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. These bits are not available in the classical CAN function.

CFE bit (Common FIFO Enable)

The CFE bit enables the FIFO when set. FIFO is disabled when this bit is cleared.

This bit can also be used, by clearing it, to abort transmission from Common FIFO when configured in TX mode, or to stop reception into the Common FIFO in RX mode.

Only write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode and the related CANFD channel is not in CH_RESET mode for FIFOs configured as TX FIFO.

This bit can only be set if the configured FIFO depth is greater than 0x000 (CFDCFCC.CFDC > 0x000) and less than 0x110 (0x110 > CFDCFCC.CFDC > 0x000).

Set the CFE bit with a separate write access to the CFDCFCC register, after all the other bits in this register are set.

This bit is cleared automatically when the CANFD module is in GL_RESET mode.

This bit is also cleared automatically when the related channel is in CH_RESET mode if the FIFO is configured in TX mode.

CFRXIE bit (Common FIFO RX Interrupt Enable)

The CFRXIE bit enables generation of FIFO interrupts when the interrupt flag is set after reception of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

CFTXIE bit (Common FIFO TX Interrupt Enable)

The CFTXIE bit enables generation of common FIFO interrupts when the interrupt flag is set after transmission of a frame from the corresponding FIFO buffer.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

CFPLS[2:0] bits (Common FIFO Payload Data Size Configuration)

The CFPLS[2:0] bits define the message data payload allocation in the RAM. This is the maximum number of bytes which can be received or transmitted by the FIFO buffer.

For details, see section 34.6. FIFO Buffers and Normal Message Buffer Configuration.

Only write to this bit when the CANFD module is in GL_RESET mode.

Note: These bits are not available in the classical CAN function.

CFM bit (Common FIFO Mode)

The CFM bit selects the mode of the FIFO. When a hardware reset is applied, all the Common FIFO buffers are configured in RX FIFO mode.

Do not write to these bits in GL_OPERATION or GL_SLEEP mode.

Only write to these bits when the CANFD module is in GL_RESET mode.

CFITSS bit (Common FIFO Interval Timer Source Select)

The CFITSS bit selects the basic clock source for the Interval Transmission Timer.

Do not write to this bit when the CANFD module is in GL_SLEEP mode. In addition, do not write to this bit when the CFE bit is set to 1.

Do not write 1 to this bit when CANFD communication is used.*1

Note: The bit time clock can vary depending on the nominal and data rate bit configuration.

Note 1. This feature is not available in the classical CAN function.

CFITR bit (Common FIFO Interval Timer Resolution)

The CFITR bit selects the resolution of the reference clock for the Interval Transmission Timer (peripheral clock is the source for the reference clock).

Do not write to this bit when the CANFD module is in GL_SLEEP mode. Also, do not write to this bit when the CFE bit is set to 1.

CFIM bit (Common FIFO Interrupt Mode)

The CFIM bit selects the interrupt generation condition for the FIFO buffer.

Do not write to this bit in GL_SLEEP mode.

Only write to this bit when the CANFD module is in GL_RESET mode.

CFIGCV[2:0] bits (Common FIFO Interrupt Generation Counter Value)

The CFIGCV[2:0] bits select the message counter value for the generation of FIFO interrupts. These values represent fractions of the FIFO depth at which the interrupt is to be generated.

Do not write to these bits when the CANFD module is in GL_SLEEP mode.

The setting of these bits should be synchronized with the CFDC[2:0] bits.

Only write to these bits when the CANFD module is in GL_RESET mode.

CFTML[1:0] bits (Common FIFO TX Message Buffer Link)

The CFTML[1:0] bits select the normal transmit message buffer position where the TX FIFO is linked to, for transmission scanning.

Do not write to these bits in GL_OPERATION or GL_SLEEP mode.

Only write to this bit when the CANFD module is in GL_RESET mode.

CFDC[2:0] bits (Common FIFO Depth Configuration)

The CFDC[2:0] bits select the depth of the common FIFO in terms of the number of messages. If the FIFO depth is configured to 0 message, the FIFO cannot be used.

Only write to these bits when the CANFD module is in GL_RESET mode.

CFITT[7:0] bits (Common FIFO Interval Transmission Time)

The CFITT[7:0] bits select the delay in the start of transmission for all messages transmitted from this FIFO buffer when configured in TX mode. The delay is a multiple of the basic interval timer clock source period (reference clock × 1, reference clock × 10, or bit time clock of the related CAN channel).

Do not write to these bits when the CANFD module is in GL_SLEEP mode.

Do not write to these bits when the CFE bit is set to 1.

When CFDGCFG.ITRCP[15:0] = 0x0000, set the CFITT[7:0] bits to 0x0000.

34.2.29 CFDCFSTS : Common FIFO Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0058

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CFMC[5:0]						—	—	—	CFTXIF	CFRXIF	CFMLT	CFFLL	CFEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	CFEMP	Common FIFO Empty 0: FIFO not empty 1: FIFO empty	R
1	CFFLL	Common FIFO Full 0: FIFO not full 1: FIFO full	R
2	CFMLT	Common FIFO Message Lost 0: Number of message lost in FIFO 1: FIFO message lost	R/W
3	CFRXIF	Common RX FIFO Interrupt Flag 0: FIFO interrupt condition not satisfied after frame reception 1: FIFO interrupt condition satisfied after frame reception	R/W
4	CFTXIF	Common TX FIFO Interrupt Flag 0: FIFO interrupt condition not satisfied after frame transmission 1: FIFO Interrupt condition satisfied after frame transmission	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
13:8	CFMC[5:0]	Common FIFO Message Count Number of messages stored in FIFO	R
31:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CFEMP bit (Common FIFO Empty)

The CFEMP bit is set automatically when:

- The CPU has read all messages from the FIFO configured in RX mode
- All messages have been transmitted from the FIFO configured in TX mode
- The FIFO is disabled by setting the CFE bit to 0
- The CANFD module is in GL_RESET mode

- The related CANFD channel is in CH_RESET when FIFO configured in TX mode.

The CFEMP bit is cleared automatically when:

- The first reception message is stored in the FIFO buffer when configured in RX mode
- The first message to be transmitted is stored in the FIFO buffer when configured in TX mode.

CFFLL bit (Common FIFO Full)

The CFFLL bit is set automatically when the number of CAN messages stored in the FIFO matches the configured FIFO depth.

The CFFLL bit is cleared automatically when:

- The number of CAN messages stored in the FIFO is less than the configured FIFO depth
- The FIFO is disabled by setting the CFE bit to 0
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode when FIFO buffer is configured in TX mode.

CFMLT bit (Common FIFO Message Lost)

The CFMLT bit is set automatically whenever a message is lost due to attempted storage of a new message when FIFO is already full in RX mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then this bit is set.

Only write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode and the related CANFD channel is not in CH_RESET mode for FIFO configured as TX FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

The CFMLT bit is cleared:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode
- When the related CANFD channel is in CH_RESET mode if the FIFO buffer is configured in TX mode.

CFRXIF bit (Common RX FIFO Interrupt Flag)

The CFRXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode and the related CANFD channel is not in CH_RESET mode for FIFO configured as TX FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the configured interrupt condition is satisfied for Common FIFO buffers when configured in RX mode.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

The CFRXIF bit is cleared:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode

CFTXIF bit (Common TX FIFO Interrupt Flag)

The CFTXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode and the related CANFD channel is not in CH_RESET mode for FIFO buffer configured as TX FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the configured interrupt condition is satisfied for Common FIFO buffers configured in TX mode.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

The CFTXIF bit is cleared:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode
- When the related CANFD channel is in CH_RESET mode if the FIFO buffer is configured in TX mode.

CFMC[5:0] bits (Common FIFO Message Count)

The CFMC[5:0] bits indicate the following:

- Number of CAN messages stored by the CPU in the FIFO buffer configured in TX mode pending for transmission
- Number of CAN messages stored in the FIFO buffer configured in RX mode by CANFD module to be read by the CPU

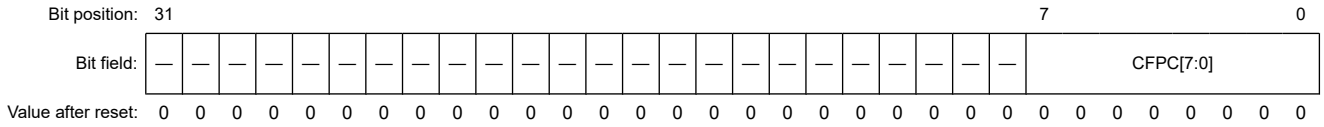
The CFMC[5:0] bits are cleared automatically when:

- The FIFO is disabled
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode if the FIFO buffer is configured in TX mode.

34.2.30 CFDCFPCTR : Common FIFO Pointer Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x005C



Bit	Symbol	Function	R/W
7:0	CFPC[7:0]	Common FIFO Pointer Control Increments read or write pointer of the corresponding Common FIFO buffers depending on the mode configuration.	W
31:8	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The Common FIFO Pointer Control Registers can be used to increment the read or write pointer of the corresponding Common FIFO buffer.

CFPC[7:0] bits (Common FIFO Pointer Control)

When the value 0xFF is written into the CFPC[7:0] bits, the read pointer of the corresponding Common FIFO buffer (when configured in RX mode), or the write pointer of the corresponding Common FIFO buffer (when configured in TX mode) moves to the next FIFO entry.

The read value from these bits is always 0x00.

Only write to these bits when the CANFD module is in GL_HALT or GL_OPERATION mode.

Only write 0xFF to this register when:

- The Common FIFO buffer is enabled and is not empty if configured in RX mode
- The Common FIFO buffer is enabled and is not full if configured in TX mode

Do not write to the Common FIFO Pointer Control registers when DMA is enabled.

34.2.31 CFDFESTS : FIFO Empty Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0060

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFEMP	—	—	—	—	—	—	RFXEMP[1:0]	
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
1:0	RFXEMP[1:0]	RX FIFO Empty Status 0: Corresponding FIFO not empty 1: Corresponding FIFO empty	R
7:2	—	These bits are read as 0.	R
8	CFEMP	Common FIFO Empty Status 0: Corresponding FIFO not empty 1: Corresponding FIFO empty	R
31:9	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The FIFO Empty Status register shows status of the empty bits of the FIFO buffers.

RFXEMP[1:0] bits (RX FIFO Empty Status)

The RFXEMP[1:0] bits are set when the CANFD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

CFEMP bit (Common FIFO Empty Status)

The CFEMP bits are set when the CANFD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

34.2.32 CFDFFSTS : FIFO Full Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0064

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFFLL	—	—	—	—	—	—	RFXFLL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RFXFLL[1:0]	RX FIF0 Full Status 0: Corresponding FIFO not full 1: Corresponding FIFO full	R
7:2	—	These bits are read as 0.	R
8	CFFLL	Common FIF0 Full Status 0: Corresponding FIFO not full 1: Corresponding FIFO full	R
31:9	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The FIFO Full Status Register shows status of the full bits of the FIFO buffers.

RFXFLL[1:0] bits (RX FIF0 Full Status)

The RFXFLL[1:0] bits are cleared when CANFD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

CFFLL bits (Common FIF0 Full Status)

The CFFLL bits are cleared when the CANFD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

34.2.33 CFDFMSTS : FIFO Message Lost Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0068

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFMLT	—	—	—	—	—	—	RFXMLT[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RFXMLT[1:0]	RX FIFO Message Lost Status 0: Corresponding FIFO Message Lost flag not set 1: Corresponding FIFO Message Lost flag set	R
7:2	—	These bits are read as 0.	R
8	CFMLT	Common FIFO Message Lost Status 0: Corresponding FIFO Message Lost flag not set 1: Corresponding FIFO Message Lost flag set	R
31:9	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The FIFO Message Lost Status Register shows status of the Msg Lost bits of the FIFO buffers.

RFXMLT[1:0] bits (RX FIFO Message Lost Status)

The RFXMLT[1:0] bits are cleared when the CANFD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

CFMLT bits (Common FIFO Message Lost Status)

The CFMLT bits are cleared when the CANFD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

34.2.34 CFDRFISTS : RX FIFO Interrupt Flag Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x006C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFXIF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RFXIF[1:0]	RX FIFO[x] Interrupt Flag Status 0: Corresponding RX FIFO Interrupt flag not set 1: Corresponding RX FIFO Interrupt flag set	R
31:2	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The FIFO Interrupt Flag Status Register shows status of the interrupt flag bits of the RX FIFO buffers.

RFXIF[1:0] bits (RX FIFO[x] Interrupt Flag Status)

Each bit is set automatically when the corresponding interrupt flag bit is set in the RX FIFO Status Registers.

The RFXIF[1:0] bits are cleared when the CANFD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding interrupt flag bit is cleared in the RX FIFO Status Registers.

34.2.35 CFDCDTCT : DMA Transfer Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00C8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFDMAE	—	—	—	—	—	—	RFDMAE1	RFDMAE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFDMAE0	DMA Transfer Enable for RXFIFO 0 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
1	RFDMAE1	DMA Transfer Enable for RXFIFO 1 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	CFDMAE	DMA Transfer Enable for Common FIFO 0 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The DMA Transfer Control Register controls the start and stop of DMA transfer operation.

RFDMAEe (e = 0 to 1) bit (DMA Transfer Enable for RXFIFO e)

The RFDMAEe bit cannot be set in GL_SLEEP or GL_RESET mode.

This bit is cleared when the CANFD module is in GL_RESET mode.

CFDMAE bit (DMA Transfer Enable for Common FIFO)

The CFDMAE bit enables or disables DMA transfer request for common FIFO

The CFDMAE bit cannot be set in GL_SLEEP or GL_RESET mode.

Do not enable a DMA transfer for a Common FIFO that is configured as TX FIFO.

This bit is cleared when the CANFD module is in GL_RESET mode.

34.2.36 CFDCDTSTS : DMA Transfer Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00CC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFDMASTS	—	—	—	—	—	—	RFDMASTS1	RFDMASTS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFDMASTS0	DMA Transfer Status for RX FIFO 0 0: DMA transfer stopped 1: DMA transfer on going	R
1	RFDMASTS1	DMA Transfer Status for RX FIFO 1 0: DMA transfer stopped 1: DMA transfer on going	R
7:2	—	These bits are read as 0.	R
8	CFDMASTS	DMA Transfer Status only for Common FIFO 0: DMA transfer stopped 1: DMA transfer on going	R
31:9	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The DMA Transfer Status Register shows the status of the DMA transfer.

RFDMASTSe (e = 0 to 1) bit (DMA Transfer Status for RX FIFO e)

Each bit is set automatically when the corresponding DMA enable bit is set and the corresponding DMA FIFO is not empty.

Each bit is cleared automatically when the DMA transfer stops either because the DMA is disabled or the DMA FIFO is empty.

When CFDCDTCT.RFDMAEe (see CFDCDTCT.RFDMAEe bit in section 34.2.35. CFDCDTCT : DMA Transfer Control Register) is set to 0 while DMA transfer for the corresponding FIFO is on going, the RFDMASTSe bit becomes 0 when the DMA transfer is complete.

This bit is cleared when the CANFD module is in GL_RESET mode.

CFDMASTS bit (DMA Transfer Status only for Common FIFO)

Each bit is set automatically when the corresponding DMA enable bit is set and the corresponding DMA FIFO is not empty.

Each bit is cleared automatically when the DMA transfer stops either because the DMA is disabled or the DMA FIFO is empty.

When CFDCDTCT.CFDMAE (see CFDCDTCT.CFDMAE bit in section 34.2.35. CFDCDTCT : DMA Transfer Control Register) is set to 0 while DMA transfer for the corresponding FIFO is on going, the CFDMASTS bit becomes 0 when the DMA transfer is complete.

This bit is cleared when the CANFD module is in GL_RESET mode.

34.2.37 CFDTMCi : TX Message Buffer Control Registers i (i = 0 to 3)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0070 + 0x01 × i

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TMOM	TMTAR	TMTR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMTR	TX Message Buffer Transmission Request 0: TX Message buffer transmission not requested 1: TX message buffer transmission requested	R/W
1	TMTAR	TX Message Buffer Transmission Abort Request 0: TX message buffer transmission request abort not requested 1: TX message buffer transmission request abort requested	R/W
2	TMOM	TX Message Buffer One-shot Mode 0: TX message buffer not configured in one-shot mode 1: TX message buffer configured in one-shot mode	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The TX Message Buffer Control Registers configure the TX message buffer functions.

TMTR bit (TX Message Buffer Transmission Request)

When the TMTR bit is set, the CANFD module logic tries to transmit the message stored in the corresponding message buffer.

Only write to this bit when the related CANFD module is in CH_HALT or CH_OPERATION mode.

Do not set this bit if the corresponding TX message buffer is linked to a COM FIFO in TX mode or is a part of TX Queue.

This bit cannot be directly cleared by a CPU write access.

This bit can only be set when the Transmission Result flag bits (CFDTMSTSj.TMTRF) in the CFDTMSTSj register corresponding to the message buffer are cleared to 00b.

The TMTR bit is automatically cleared by the:

- CANFD module logic at the end of a successful transmission
- CANFD module logic at the end of a transmission abort, requested by the corresponding CFDTMCi.TMTAR bit
- CANFD module logic when there is a detection of a CAN bus error or arbitration loss if CFDTMCi.TMOM bit is set for the message buffer
- CANFD module logic when the CANFD module is in GL_RESET mode or the related channel is in CH_RESET mode.

TMTAR bit (TX Message Buffer Transmission Abort Request)

When the TMTAR bit is set, the CANFD module logic tries to abort the transmission of the frame stored in the corresponding message buffer.

In most cases, transmission cannot be aborted if the internal scan for transmission is complete and the message buffer has already been selected for transmission. In this case, frame may be transmitted successfully from the message buffer. The message buffer selection is released by entering CH_HALT mode.

However, message buffer selected for transmission can be aborted by an abort request when the CAN node detects a new message on the bus (RX pin) before it starts transmission from the selected message buffer.

Only write to the TMTAR bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode. This bit can only be set when the related transmit request TMTR bit is set.

The TMTAR bit cannot be cleared by a CPU write access. Clearing of this bit by CANFD has priority over setting by a CPU write access.

The TMTAR bit is automatically cleared by:

- The CANFD module logic at the end of a successful transmission
- The CANFD module logic at the end of a transmission abort
- The CANFD module logic when there is detection of a CAN bus error or arbitration loss
- The CANFD module logic when the CANFD module is in GL_RESET mode or the related channel enters CH_RESET mode.

TMOM bit (TX Message Buffer One-shot Mode)

When the TMOM bit is set, the CANFD module logic tries to transmit the message only once.

If the transmission is successful, the CFDTMSTSj.TMTRF bits are set to 10b or 11b. Otherwise, the transmission is automatically aborted and CFDTMSTSj.TMTRF bits are set to 01b due to a bus error or a bus arbitration lost.

The TMOM bit remains set if the transmission has completed successfully or aborted due to an error or a loss of arbitration.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Set this bit at the same time as the TMTR bit. Clear this bit with a write access.

If a message has already been requested for transmission, do not write to this bit until the message has been successfully transmitted or transmission has been aborted.

The TMOM bit is automatically cleared by the CANFD module logic when the CANFD module is in GL_RESET mode or the related channel is in CH_RESET mode.

34.2.38 CFDTMSTSj : TX Message Buffer Status Registers j (j = 0 to 3)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0074 + 0x01 × j

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TMTARM	TMTRM	TMTRF[1:0]		TMTSTS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMTSTS	TX Message Buffer Transmission Status 0: No on-going transmission 1: On-going transmission	R
2:1	TMTRF[1:0]	TX Message Buffer Transmission Result Flag 0 0: No result 0 1: Transmission aborted from the TX message buffer 1 0: Transmission successful from the TX message buffer and transmission abort was not requested 1 1: Transmission successful from the TX message buffer and transmission abort was requested	R/W
3	TMTRM	TX Message Buffer Transmission Request Mirrored 0: TX message buffer transmission not requested 1: TX message buffer transmission requested	R
4	TMTARM	TX Message Buffer Transmission Abort Request Mirrored 0: TX message buffer transmission request abort not requested 1: TX message buffer transmission request abort requested	R
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The TX Message Buffer Status Registers show status of the transmission and transmission abort for the corresponding message buffers.

TMTSTS bit (TX Message Buffer Transmission Status)

The TMTSTS bit is set automatically at the start of the transmission from the corresponding TX message buffer.

This bit is cleared automatically when:

- Transmission stops
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode.

TMTRF[1:0] bits (TX Message Buffer Transmission Result Flag)

The TMTRF[1:0] bits show the result for the corresponding TX message buffer. The status is as follows:

- 00: Transmission in progress or has not been requested
- 01: Transmission has been aborted from the corresponding TX message buffer
- 10: Transmission was successful from the corresponding TX message buffer and the CFDTMCi.TMTAR bit was not set for this TX message buffer
- 11: Transmission was successful from the corresponding TX message buffer, but the CFDTMCi.TMTAR bit was set for this TX message buffer.

Only write to these bits when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

The TMTRF[1:0] bits are cleared automatically when the CANFD module is in GL_RESET mode or the related channel is in CH_RESET mode.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

TMTRM bit (TX Message Buffer Transmission Request Mirrored)

The TMTRM bit is set when the CFDTMCi.TMTR bit in the corresponding CFDTMCi register is set.

This bit is cleared when the CFDTMCi.TMTR bit in the corresponding CFDTMCi register is cleared.

TMTARM bit (TX Message Buffer Transmission Abort Request Mirrored)

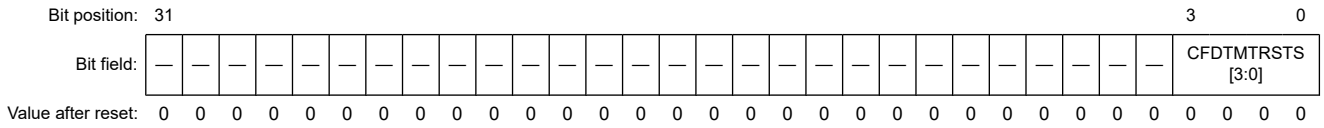
The TMTARM bit is set when the CFDTMCi.TMTAR bit in the corresponding CFDTMCi register is set.

This bit is cleared when the CFDTMCi.TMTAR bit in the corresponding CFDTMCi register is cleared.

34.2.39 CFDTMTRSTS : TX Message Buffer Transmission Request Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0078



Bit	Symbol	Function	R/W
3:0	CFDTMTRSTS[3:0]	TX Message Buffer Transmission Request Status 0: Transmission not requested for corresponding TX message buffer 1: Transmission requested for corresponding TX message buffer	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

These bits show the TX Message Buffer Transmission Request Status for the corresponding TX Message Buffer. The bit 0 of a CFDTMTRSTS register corresponds to the TX message buffer 0.

The bit position of CFDTMTRSTS corresponds to the buffer number of TX message buffer.

CFDTMTRSTS[3:0] bits (TX Message Buffer Transmission Request Status)

The CFDTMTRSTS[3:0] bits show status of the CFDTMCi.TMTR bits of the TX Message Buffer Control Registers.

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Control Registers (CFDTMCi), and only when the message buffer does not belong to a TX Queue.

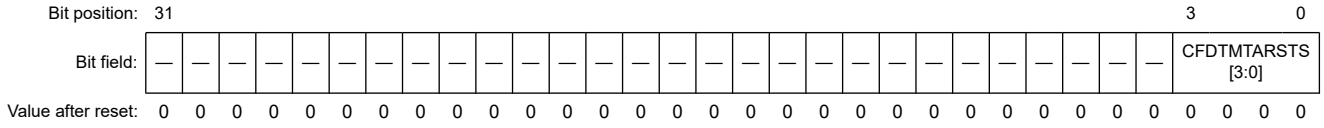
Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Control Registers
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode.

34.2.40 CFDTMTARSTS : TX Message Buffer Transmission Abort Request Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x007C



Bit	Symbol	Function	R/W
3:0	CFDTMTARSTS[3:0]	TX Message Buffer Transmission Abort Request Status 0: Transmission abort not requested for corresponding TX message buffer 1: Transmission abort requested for corresponding TX message buffer	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

These bits show the TX Message Buffer Transmission Abort Request Status for the corresponding TX Message Buffer. The bit 0 of a CFDTMTARSTS register corresponds to the TX message buffer 0.

The bit position of CFDTMTARSTS corresponds to the buffer number of TX message buffer.

CFDTMTARSTS[3:0] bits (TX Message Buffer Transmission Abort Request Status)

The CFDTMTARSTS[3:0] bits show status of the CFDTMCi.TMTAR bits of the TX Message Buffer Control Registers.

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Control Registers, and when the message buffer belongs to a TX Queue.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Control Registers
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode.

34.2.41 CFDTMTCSTS : TX Message Buffer Transmission Completion Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0080

Bit position:	31																										3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFDTMTCSTS [3:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CFDTMTCSTS[3:0]	TX Message Buffer Transmission Completion Status 0: Transmission not complete for corresponding TX message buffer 1: Transmission completed for corresponding TX message buffer	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

These bits show the TX Message Buffer Transmission Completion Status for the corresponding TX Message Buffer. The bit 0 of a CFDTMTCSTS register corresponds to the TX message buffer 0.

The bit position of CFDTMTCSTS corresponds to the buffer number of TX message buffer.

CFDTMTCSTS[3:0] bits (TX Message Buffer Transmission Completion Status)

The CFDTMTCSTS[3:0] bits show status of successful completion of the TX Message Buffer Status Registers.

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Status Registers.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Status Registers
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode.

If a CAN channel enters CH_RESET mode, then the bits related to that channel are cleared.

34.2.42 CFDTMTASTS : TX Message Buffer Transmission Abort Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0084

Bit position:	31																										3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFDTMTASTS [3:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CFDTMTASTS[3:0]	TX Message Buffer Transmission Abort Status 0: Transmission not aborted for corresponding TX message buffer 1: Transmission aborted for corresponding TX message buffer	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

These bits show the TX Message Buffer Transmission abort Status for the corresponding TX Message Buffer. The bit 0 of a CFDTMTASTS register corresponds to the TX message buffer 0.

The bit position of CFDTMTASTS corresponds to the buffer number of TX message buffer.

CFDTMTASTS[3:0] bits (TX Message Buffer Transmission Abort Status)

The CFDTMTASTS[3:0] bits show status of the successful transmission abort of the corresponding TX message buffer.

Each bit is set automatically when the CFDTMSTSj.TMTRF bits are set to 01b in the corresponding TX Message Buffer Status Register.

Each bit is cleared automatically when:

- The CFDTMSTSj.TMTRF bits are cleared in the corresponding TX Message Buffer Status Register
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode.

34.2.43 CFDTMIEC : TX Message Buffer Interrupt Enable Configuration Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0088

Bit position:	31																											3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TMIEg[3:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	TMIEg[3:0]	TX Message Buffer Interrupt Enable 0: TX message buffer interrupt disabled for corresponding TX message buffer 1: TX message buffer interrupt enabled for corresponding TX message buffer	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

These bits show the TX Message Buffer Interrupt Enable for the corresponding TX Message Buffer.

The bit 0 of a CFDTMIEC register corresponds to the TX message buffer 0.

The bit position of CFDTMIEC corresponds to the buffer number of TX message buffer.

g = [0…3]

TMIEg[3:0] bits (TX Message Buffer Interrupt Enable)

If the TMIEg[3:0] bits are set, an interrupt is generated at the end of a successful transmission from the corresponding message buffer.

See section 34.7. Interrupts and DMA for TX Message Buffer Interrupt specification.

Do not write to the TMIEg[7:0] bits when:

- The CANFD module is in GL_SLEEP mode
- The related CANFD channel is in CH_SLEEP mode
- The corresponding TX message buffer is linked to a Common FIFO with the CFDCFCC.CFTML bits.

34.2.44 CFDTXQCC : TX Queue Configuration/Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x008C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TXQDC[1:0]		TXQIM	—	TXQTXIE	—	—	—	—	TXQE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXQE	TX Queue Enable 0: TX Queue disabled 1: TX Queue enabled	R/W
4:1	—	These bits are read as 0. The write value should be 0.	R/W
5	TXQTXIE	TX Queue TX Interrupt Enable 0: TX Queue TX interrupt disabled 1: TX Queue TX interrupt enabled	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	TXQIM	TX Queue Interrupt Mode 0: When the last message is successfully transmitted 1: At every successful transmission	R/W
9:8	TXQDC[1:0]	TX Queue Depth Configuration 0x00: 0 messages 0x01: Reserved 0x10: 3 messages 0x11: 4 messages	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The TX Queue Configuration/Control Registers are used to configure the TX Queue transmission.

TXQ is composed of TXMB0 to TXMB3 (at the maximum) when TXQE is enabled.

TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 0x00 (CFDTXQCC.TXQDC == 0x00).

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH_RESET or CH_SLEEP mode.

The TXQE bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH_SLEEP mode.

TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQDC[1:0] bits (TX Queue Depth Configuration)

The TXQDC[1:0] bits select the depth of the transmission queue. The message buffer selection starts from MB[0] up to MB[3] depending on the configured depth.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

34.2.45 CFDTXQSTS : TX Queue Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0090

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TXQMC[2:0]			—	—	—	—	—	TXQTXIF	TXQFLL	TXQEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	TXQEMP	TX Queue Empty 0: TX Queue not empty 1: TX Queue empty	R
1	TXQFLL	TX Queue Full 0: TX Queue not full 1: TX Queue full	R
2	TXQTXIF	TX Queue TX Interrupt Flag 0: TX Queue interrupt condition not satisfied after a frame TX 1: TX Queue interrupt condition satisfied after a frame TX	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
10:8	TXQMC[2:0]	TX Queue Message Count Number of messages in the TX Queue.	R
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The TX Queue Status Registers show the status of the TX Queue of corresponding CAN channel.

TXQEMP bit (TX Queue Empty)

The TXQEMP bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

This bit is set automatically when:

- The last message is transmitted from the TX Queue
- The related CANFD channel is in CH_RESET mode.

The bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

TXQFLL bit (TX Queue Full)

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth
- The related CANFD channel is in CH_RESET mode.

TXQTXIF bit (TX Queue TX Interrupt Flag)

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

You cannot write to this bit when the related CANFD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

TXQMC[2:0][13:8] bits (TX Queue Message Count)

The TXQMC[2:0] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

34.2.46 CFDTXQPCTR : TX Queue Pointer Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0094

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQPC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TXQPC[7:0]	TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel	W
31:8	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The TX Queue Pointer Control Registers are used to confirm storage of a full message in the corresponding TX Queue buffers.

TXQPC[7:0] bits (TX Queue Pointer Control)

When the value 0xFF is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 0x00. Do not write to the FIFO control registers when DMA is enabled.

You cannot write to these bits when the related CANFD channel is in CH_SLEEP or CH_RESET mode.

Only write 0xFF to this register when:

- The corresponding TX Queue is enabled and not full

- The Common FIFO is enabled.

34.2.47 CFDTHLCC : TX History List Configuration/Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0098

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	THLDTE	THLIM	THLIE	—	—	—	—	—	—	—	THLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	THLE	TX History List Enable 0: TX History List disabled 1: TX History List enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	THLIE	TX History List Interrupt Enable 0: TX History List Interrupt disabled 1: TX History List Interrupt enabled	R/W
9	THLIM	TX History List Interrupt Mode 0: Interrupt generated if TX History List level reaches ¾ of the TX History List depth 1: Interrupt generated for every successfully stored entry	R/W
10	THLDTE	TX History List Dedicated TX Enable 0: TX FIFO + TX Queue 1: Flat TX MB + TX FIFO + TX Queue	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The TX History List Configuration/Control Register configures the TX History List functions.

THLE bit (TX History List Enable)

The THLE bit enables the TX History List buffer when it is set.

You cannot write to this bit when the related CANFD channel is in CH_RESET or CH_SLEEP mode.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

THLIE bit (TX History List Interrupt Enable)

The THLIE bit enables the generation of the TX History List interrupt when it is set.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

THLIM bit (TX History List Interrupt Mode)

The THLIM bit selects the interrupt generation condition for the FIFO.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode.

THLDTE bit (TX History List Dedicated TX Enable)

The THLDTE bit selects the condition for storing an entry in the TX History List after successful transmission.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode.

34.2.48 CFDTHLSTS : TX History List Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x009C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	THLMC[3:0]				—	—	—	—	THLIF	THLELT	THLFLL	THLEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	THLEMP	TX History List Empty 0: TX History List not empty 1: TX History List empty	R
1	THLFLL	TX History List Full 0: TX History List not full 1: TX History List full	R
2	THLELT	TX History List Entry Lost 0: No entry lost in TX History List 1: TX History List entry Lost	R/W
3	THLIF	TX History List Interrupt Flag 0: TX History List interrupt condition not satisfied 1: TX History List interrupt condition satisfied	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
11:8	THLMC[3:0]	TX History List Message Count Number of messages stored in TX History List	R
31:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The TX History List Status register shows the status of data stored in the TX History List buffer.

THLEMP bit (TX History List Empty)

The THLEMP bit is set automatically when the CPU has read all the entries from the TX History List buffer.

This bit is cleared automatically when the first entry is stored to the TX History List.

This bit is set automatically when:

- TX History List is disabled
- The related CANFD channel is in CH_RESET mode.

THLFLL bit (TX History List Full)

The THLFLL bit is set automatically when the number of entries in the TX History List buffer matches the TX History List depth.

Each TX History List can store up to 8 entries.

This bit is cleared automatically when:

- The number of entries in the TX History List buffer is less than the TX History List depth
- The TX History List is disabled
- The related CANFD channel is in CH_RESET mode.

THLELT bit (TX History List Entry Lost)

The THLELT bit is set when a new entry cannot be stored because the related TX History List buffer is already full.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

THLIF bit (TX History List Interrupt Flag)

The THLIF bit is set when the configured interrupt condition is satisfied.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

The bit is cleared by writing 0 to it.

This bit is automatically cleared in CH_RESET mode.

THLMC[3:0] bits (TX History List Message Count)

The THLMC[3:0] bits show the number of transmitted messages stored in the TX History List.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

34.2.49 CFDTHLACC0 : TX History List Access Register 0

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0740

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TMTS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	BN[1:0]		BT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	BT[2:0]	Buffer Type 0 0 1: Flat TX message buffer 0 1 0: TX FIFO message buffer number 1 0 0: TX Queue message buffer number	R
4:3	BN[1:0]	Buffer Number Number of the message buffer	R

Bit	Symbol	Function	R/W
15:5	—	These bits are read as 0.	R
31:16	TMTS[15:0]	Transmit Timestamp Transmit timestamp value for software drivers	R

Note: S-TYPE-3, P-TYPE-3

The TX History List Access Registers 0 provide access to the entry in the TX History List based on the read timestamp value.

BT[2:0] bits (Buffer Type)

The BT[2:0] bits indicate whether data has been stored following a transmission from a FIFO buffer, a TX Queue or a TX message buffer.

BN[1:0] bits (Buffer Number)

The BN[1:0] bits show the message buffer from which transmission was successfully completed. If a message from a Common FIFO is transmitted, then these bits show the message buffer that is linked to the Common FIFO for transmission.

TMTS[15:0] bits (Transmit Timestamp)

The TMTS[15:0] bits indicate the timestamp for use by software drivers.

34.2.50 CFDTHLACC1 : TX History List Access Register 1

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0744

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TIFL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TID[15:0]	Transmit ID These bits indicate that message buffer reference ID, TX FIFO reference ID, or AFL pointer field is stored for software drivers.	R
17:16	TIFL[1:0]	Transmit Information Label These bits indicate that message buffer information label, TX FIFO information label, or AFL information label is stored for software drivers.	R
31:18	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The TX History List Access Registers 1 provide access to entry in the TX History List based on the read pointer value.

TID[15:0] bits (Transmit ID)

The TID[15:0] bits indicate whether the message buffer reference ID (CFDTMFDCTRb.TMPTR) or the TX FIFO reference ID (CFDCFFDCSTS.CFPTR) is for use by software drivers.

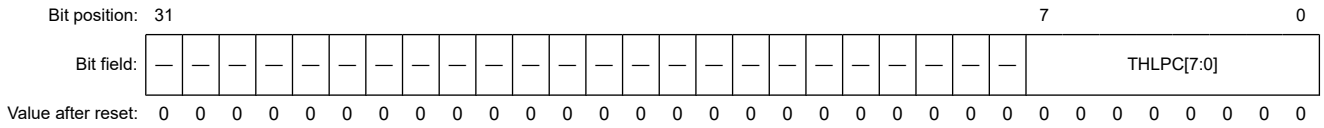
TIFL[1:0] bits (Transmit Information Label)

The TIFL[1:0] bits indicate whether the message buffer information label (CFDTMFDCTRb.TMIFL) or the TX FIFO information label (CFDCFFDCSTS.CFIFL) is for use by software drivers.

34.2.51 CFDTHLPCTR : TX History List Pointer Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00A0



Bit	Symbol	Function	R/W
7:0	THLPC[7:0]	TX History List Pointer Control Increments the write pointer to the TX History List in the corresponding channel	W
31:8	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The TX History List Pointer Control Registers are used to increment the read pointer of the TX History List.

THLPC[7:0] bits (TX History List Pointer Control)

When 0xFF is written to the THLPC[7:0] bits, the read pointer of the TX History List is moved to the next TX History List entry address.

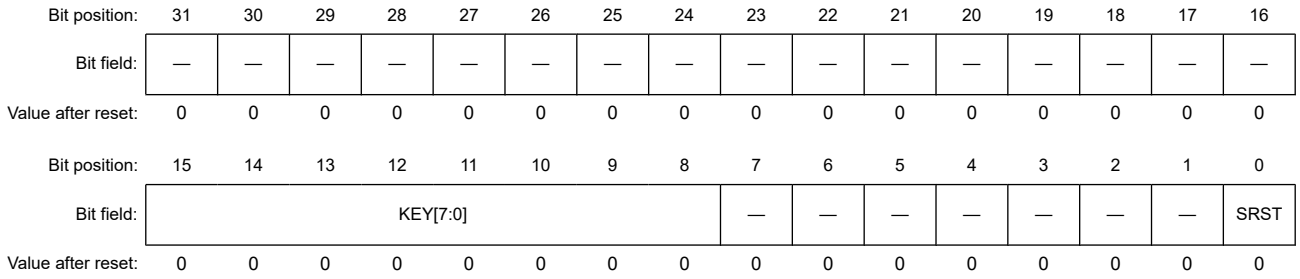
The read value from these bits is always 0x00. Only write to these bits when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Only write 0xFF to these registers when the corresponding TX History List is enabled and not empty.

34.2.52 CFDGRSTC : Global SW reset Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00D8



Bit	Symbol	Function	R/W
0	SRST	SW Reset 0: Normal state 1: SW reset state	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits control the validity of rewriting of a SRST bit.	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

SRST bit (SW Reset)

When the SRST bit is set, the CANFD module is in the same state as hardware reset. When a reset is required, write 1 then write 0 to this bit.

This bit is cleared when the CANFD module is in GL_SLEEP mode.

When this bit is cleared, the RAM initialization sequence does not operate. The configuration of RAM is performed by software.

The RAM is not initialized when software reset is performed during the initialization of RAM. Software must perform the initialization of RAM.

KEY[7:0] bits (Key Code)

When 0xC4 is written in the KEY[15:8] bits, a write to the SRST bit is valid.

The read value from these bits is always 0x00.

CFDGRSTC.SRST bit and the CFDGRSTC.KEY bit should be written simultaneously.

34.2.53 CFDGTSTCFG : Global Test Configuration Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00A8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	RTMPS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
19:16	RTMPS[3:0]	RAM Test Mode Page Select Select a RAM test mode page	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Test Configuration Register is used to configure the RAM test mode page.

RTMPS[3:0] bits (RAM Test Mode Page Select)

The RTMPS[3:0] bits select the RAM page mode for CPU read/write access when the CANFD module is configured in RAM test mode.

See section 34.9.2.1. RAM Test Mode for the RAM test mode specification.

Do not write to these bits when the CANFD module is in GL_RESET or GL_SLEEP mode.

Only enter values from 0 to 8 (0x008) for the message buffer RAM.

Only write to these bits when the CANFD module is in GL_HALT mode.

These bits are cleared automatically when the related CANFD channel is in GL_RESET mode.

34.2.54 CFDGTSTCTR : Global Test Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00AC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	RTME	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	RTME	RAM Test Mode Enable 0: RAM test mode disabled 1: RAM test mode enabled	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Test Control register is used to control the global test modes of the CANFD module.

RTME bit (RAM Test Mode Enable)

When the RTME bit is set, the CANFD module is configured in RAM test mode. See section 34.9.2.1. RAM Test Mode for RAM test mode specification.

Only write to this bit when the CANFD module is in GL_HALT mode.

Clear this bit when the CANFD module is in GL_HALT mode.

This bit is cleared automatically when the CANFD module is in GL_RESET mode.

34.2.55 CFDGFDCFG : Global FD Configuration Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00B0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TSCCFG[1:0]		—	—	—	—	—	—	—	RPED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RPED	RES Bit Protocol Exception Disable 0: Protocol exception event detection enabled 1: Protocol exception event detection disabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
9:8	TSCCFG[1:0]	Timestamp Capture Configuration 0 0: Timestamp capture at the sample point of SOF (start of frame) 0 1: Timestamp capture at frame valid indication 1 0: Timestamp capture at the sample point of RES bit 1 1: Reserved	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

RPED bit (RES Bit Protocol Exception Disable)

The RPED bit configures the protocol exception event handling according to ISO 11898-1.

When this bit is enabled, the protocol exception event detection is disabled, and the protocol controller transmits an error frame when the protocol exception event is detected (RES bit is sampled recessive).

Only write to this bit when the CANFD module is in GL_RESET mode.

TSCCFG[1:0] bits (Timestamp Capture Configuration)

The TSCCFG[1:0] bits configure the different capture points of the timestamp for transmission and reception.

When CFDGFDCFG.TSCCFG[1:0] = 10b, the timestamp capture is performed for CANFD frames at RES bit and for Classical frames at the start of frame.

Only write to these bits when the CANFD module is in GL_RESET mode.

34.2.56 CFDGLOCKK : Global Lock Key Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00B8

Bit position:	31															15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LOCK[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	LOCK[15:0]	Lock Key Key bits for unlocking the protection of test modes	W
31:16	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The Global Lock Key register is a write-only register that is used to unlock the protection for special test bits.

See section 34.9.2. Global Test Modes for Lock key specification.

LOCK[15:0] bits (Lock Key)

The unlock key sequence must be written in the LOCK[15:0] bits to configure the CANFD module in RAM test modes.

The read value from these bits is always 0x0000.

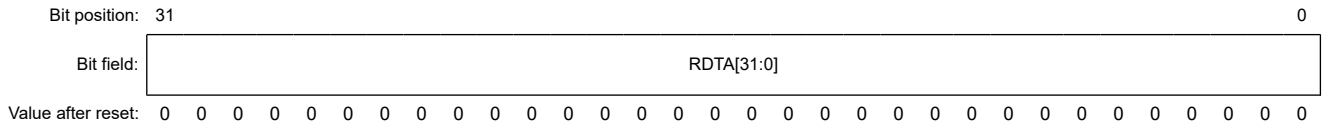
You cannot write to these bits when the CANFD module is in GL_SLEEP or GL_RESET mode.

Do not write to these bits when the CANFD module is in GL_OPERATION mode.

34.2.57 CFDRPGACCk : RAM Test Page Access Registers k (k = 0 to 63)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0280 + 0x0004 × k



Bit	Symbol	Function	R/W
31:0	RDTA[31:0]	RAM Data Test Access RAM data bytes	R/W

Note: S-TYPE-3, P-TYPE-3

RDTA[31:0] bits (RAM Data Test Access)

Data can be read from or written into the RDTA[31:0] bits when the CANFD module is configured in RAM test mode.

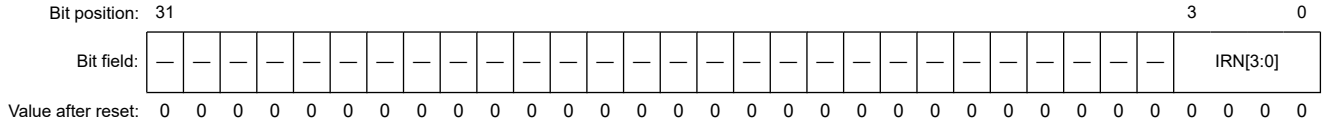
Only write to this bit when the CANFD module is in GL_HALT mode and RAM test mode is enabled.

Software data should be read/written in the RAM Test Page Access registers during RAM test mode.

34.2.58 CFDGAFLIGNENT : Global AFL Ignore Entry Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00C0



Bit	Symbol	Function	R/W
3:0	IRN[3:0]	Ignore Rule Number Define rule number which ignores an AFL entry.	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

IRN[3:0] bits (Ignore Rule Number)

The IRN[3:0] bits define the rule number which updates an AFL entry.

Enter only values between 0 and 15 (0x0F) inclusive.

Only write to these bits when the CFDGAFLIGNCTR.IREN bit is 0.

You cannot write to these bits when the CANFD module is in GL_SLEEP mode.

34.2.59 CFDGAFLIGNCTR : Global AFL Ignore Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00C4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	IREN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IREN	Ignore Rule Enable 0: AFL entry number is not ignored 1: AFL entry number is ignored	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits control the validity of rewriting the IREN bit.	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

IREN bit (Ignore Rule Enable)

When the IREN bit is set, the entry number (selected by CFDGAFLIGNENT register) is ignored.

This bit is cleared automatically when the CANFD module is in GL_RESET mode.

KEY[7:0] bits (Key Code)

When 0xC4 is written in the KEY[7:0] bits, a write to the IREN bit is valid.

The read value from these bits is always 0x00.

CFDGAFLIGNCTR.IREN bit and the CFDGAFLIGNCTR.KEY bit should be written simultaneously

34.2.60 CFDRMIEC : RX Message Buffer Interrupt Enable Configuration Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0038

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RMIE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RMIE[15:0]	RX Message Buffer Interrupt Enable 0: RX Message Buffer Interrupt disabled for corresponding RX message buffer 1: RX Message Buffer Interrupt enabled for corresponding RX message buffer	R/W

Note: S-TYPE-3, P-TYPE-3

These bits show the RX Message Buffer Interrupt Enable for the corresponding RX Message Buffer. CFDRMIEC bit 0 corresponds to RX Message Buffer 0 and so on.

The bit position of CFDRMIEC corresponds to the buffer number of RXMB.

RMIE[15:0] bit (RX Message Buffer Interrupt Enable)

If this bit is set, then an interrupt will be generated at the end of a successful reception from the corresponding Message Buffer.

For details, see section 34.7.1. Interrupts.

Users cannot write to this bit when the CANFD module is in GL_SLEEP mode.

34.2.61 Message Buffer Component Structure

34.2.61.1 Start Addresses

The start address for each of the Message Buffer component is calculated using the number of related Message Buffer components.

The start addresses for each register in the Message Buffer component are depicted in Table 34.4.

Table 34.4 Message Buffer Component Register Start Addresses

b = Message buffer component index	MBCP	p	Register	Start Address
[0…15] b = [0…7]	RMBCPb[0]	x	RMID	0x0920 + b × 0x004C
		x	RMPTR	0x0924 + b × 0x004C
		x	RMFDSTSb	0x0928 + b × 0x004C
		[1…15]	RMDFbp	0x092C + b × 0x004C + p × 0x0004
[0…15] b = [8…15]	RMBCPb[0]	x	RMIDb	0x0D20 + (b-8) × 0x004C
		x	RMPTRb	0x0D24 + (b-8) × 0x004C
		x	RMFDSTSb	0x0D28 + (b-8) × 0x004C
		[1…15]	RMDFbp	0x0D2C + (b-8) × 0x004C + p × 0x0004
[0…1]	RFMBCPb[0]	x	RFIDb	0x0520 + b × 0x004C
		x	RFPTRb	0x0524 + b × 0x004C
		x	RFFDSTSb	0x0528 + b × 0x004C
		[1…15]	RFDFbp	0x052C + b × 0x004C + p × 0x0004
[0]	CFMBCPb[0]	x	CFID	0x05B8
		x	CFPTR0	0x05BC
		x	CFFDCSTS0	0x05C0
		[1…15]	CFDFp0	0x05C4 + p × 0x0004
[0…3]	TMBCPb[0]	x	TMIDb	0x0604 + b × 0x004C
		x	TMPTRb	0x0608 + b × 0x004C
		x	TMFDCTRb	0x060C + b × 0x004C
		[1…15]	TMDFbp	0x0610 + b × 0x004C + p × 0x0004

The message buffer configuration consists of four types of Message Buffer components:

- RX Message Buffer Component (CFDRMBCPb[0])
- RX FIFO Access Message Buffer Component (CFDRFMBCPb[0])
- Common FIFO Access Message Buffer Component (CFDCFMBCP0[0])
- TX Message Buffer Component (CFDTMBCPb[0]).

Where b = the Message Buffer component index that has a range that varies based on the type of Message Buffer component.

For a summary of this configuration, see Figure 34.28. For a detailed description of the number of and the different types of message buffers, see section 34.6. FIFO Buffers and Normal Message Buffer Configuration.

As described in section 34.2. Register Descriptions, each Message Buffer component consists of the following registers:

- Identifier (ID)
- Pointer (PTR)
- Data Field (DFp).

Where p = the Data Field register index that has a range that varies based on the type of message buffer component.

Rc is the Message Buffer Component register where c = Message Buffer Component register index that has a range that varies based on the type of Message Buffer component.

A description of the registers, their associated bits and their accessibility are shown below the summary and detailed figures of each component.

In each of the figures, a cell that contains '-' means reserved and has the same behavio as reserved bits for registers in section 34.2.61. Message Buffer Component Structure.

34.2.61.2 CFDRMBCPb[0] : RX Message Buffer Component b (b = 0 to 15)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: See Table 34.4

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	Rc[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	Rc[15:0]	RX Message Buffer Component c Refer to Table 34.5, Table 34.6 and the descriptions that follow for a detailed description of each register and its related bits, contained within this message buffer component.	R/W

Note: S-TYPE-3, P-TYPE-3

Where the total number of CFDRMBCPb = 16 as shown in Figure 34.28 (c = RX Message Buffer Component Register index = [0…18])

Rc[15:0] bit (RX Message Buffer Component c)

The RX Message Buffer Component is made up of the following registers: CFDRMIDb, CFDRMPTRb, CFDRMFDSTSb, and CFDRMDFbp. Refer to Table 34.6 for details of how to interpret the structure of this buffer component and how to access the respective registers.

Table 34.5 RX Message Buffer Component Summary (1 of 2)

RX Message Buffer Component (RMBCP)	
Rc	**CANFD mode (CAN_FD_MODE = 1'b1)**
R0	RX Message Buffer (b) ID Registers
R1	RX Message Buffer (b) Pointer Registers
R2	RX Message Buffer (b) CANFD Status Registers
R3	RX Message Buffer (b) Data Field 0 Registers
R4	RX Message Buffer (b) Data Field 1 Registers
R5	RX Message Buffer (b) Data Field 2 Registers
R6	RX Message Buffer (b) Data Field 3 Registers

Table 34.5 RX Message Buffer Component Summary (2 of 2)

RX Message Buffer Component (RMBCP)	
Rc	**CANFD mode (CAN_FD_MODE = 1'b1)**
R7	RX Message Buffer (b) Data Field 4 Registers
R8	RX Message Buffer (b) Data Field 5 Registers
R9	RX Message Buffer (b) Data Field 6 Registers
R10	RX Message Buffer (b) Data Field 7 Registers
R11	RX Message Buffer (b) Data Field 8 Registers
R12	RX Message Buffer (b) Data Field 9 Registers
R13	RX Message Buffer (b) Data Field 10 Registers
R14	RX Message Buffer (b) Data Field 11 Registers
R15	RX Message Buffer (b) Data Field 12 Registers
R16	RX Message Buffer (b) Data Field 13 Registers
R17	RX Message Buffer (b) Data Field 14 Registers
R18	RX Message Buffer (b) Data Field 15 Registers
R[19…31]	—

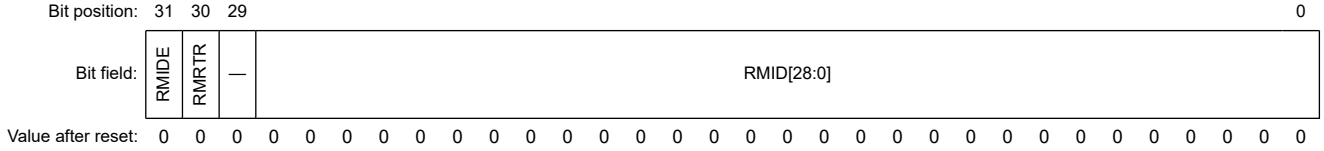
Table 34.6 RX Message Buffer Component (RMBCP) Detailed

Rc	p	Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0	x	CFDRMIDb	RMIDE	RMRTR	—	RMID																												
R1	x	CFDRMPTRb	RMDLC			—	—	—	—	—	—	—	—	—	—	—	—	—	RMTS															
R2	x	CFDRMFDSTSb	RMPTR																—	—	—	—	—	—	RMIFL		—	—	—	—	—	RMFDF	RMBRS	RMESI
R3	0	CFDRMDFbp	RMDB_HH								RMDB_HL								RMDB_LH								RMDB_LL							
R[4…18]	[1…15]	CFDRMDFbp	RMDB_HH								RMDB_HL								RMDB_LH								RMDB_LL							

34.2.61.3 CFDRMIDb : RX Message Buffer ID Registers (b = 0 to 15)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0920 + 0x004C × b (b = 0 to 7)
0x0D20 + 0x004C × (b - 8) (b = 8 to 15)



Bit	Symbol	Function	R/W
28:0	RMID[28:0]	RX Message Buffer ID Field STD-ID/EXT-ID fields	R
29	—	This bit is read as 0.	R

Bit	Symbol	Function	R/W
30	RMRTR	RX Message Buffer RTR Bit 0: Data frame 1: Remote frame	R
31	RMIDE	RX Message Buffer IDE Bit 0: STD-ID is stored 1: EXT-ID is stored	R

Note: S-TYPE-3, P-TYPE-3

The RX Message Buffer ID Register b (b = 0 to 15) store the ID field, IDE bit, and RTR bit of the received message.

RMID[28:0] bits (RX Message Buffer ID Field)

The RMID[28:0] are the bits of the STD-ID/EXT-ID fields of the message stored in the RX message buffer.

See section 34.2.61.1. Start Addresses for details on how to interpret the structure of this buffer component.

RMRTR bit (RX Message Buffer RTR Bit)

The RMRTR bit shows whether a data frame or a remote frame was stored in the RX message buffer.

Note: There are no remote frames in CANFD format. When a CANFD frame is received, the register reflects the state of the received value (the RRS bit in FD frame format).

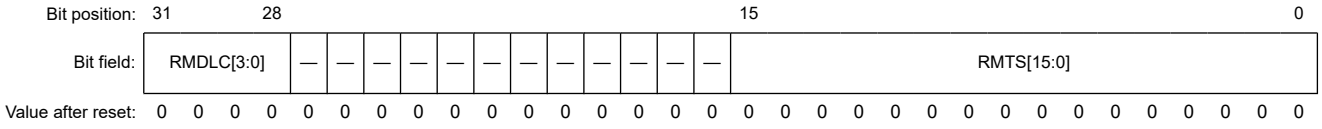
RMIDE bit (RX Message Buffer IDE Bit)

The RMIDE bit shows whether message with Standard Identifier or Extended Identifier was stored in the RX message buffer.

34.2.61.4 CFDRMPTRb : RX Message Buffer Pointer Registers (b = 0 to 15)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0924 + 0x004C × b (b = 0 to 7)
0x0D24 + 0x004C × (b - 8) (b = 8 to 15)



Bit	Symbol	Function	R/W
15:0	RMTS[15:0]	RX Message Buffer Timestamp Field Timestamp value stored for the message in the RX message buffer	R
27:16	—	These bits are read as 0.	R
31:28	RMDLC[3:0]	RX Message Buffer DLC Field Number of data bytes received in a CAN frame.	R

Note: S-TYPE-3, P-TYPE-3

The RX Message Buffer Pointer Register b (b = 0 to 15) store the DLC and Timestamp fields for the received message.

RMTS[15:0] bits (RX Message Buffer Timestamp Field)

The RMTS[15:0] bits store the timestamp value taken at the capture point as configured by CFDGFDCFG.TSCCFG of the received message.

RMDLC[3:0] bits (RX Message Buffer DLC Field)

The RMDLC[3:0] bits store the number of data bytes that were received in the RX message buffer.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes that were received.

Note: The maximum capacity of the buffer belongs to CFDRMNB.RMPLS and this is not available in the classical CAN function.

34.2.61.5 CFDRMFDSTSb : RX Message Buffer CANFD Status Registers (b = 0 to 15)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0928 + 0x004C × b (b = 0 to 7)
0x0D28 + 0x004C × (b - 8) (b = 8 to 15)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RMPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	RMIFL[1:0]		—	—	—	—	—	RMFDF	RMBRS	RMESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RMESI*1	Error State Indicator bit 0: CANFD frame received from error active node 1: CANFD frame received from error passive node	R
1	RMBRS*1	Bit Rate Switch bit 0: CANFD frame received with no bit rate switch 1: CANFD frame received with bit rate switch	R
2	RMFDF*1	CAN FD Format bit 0: Non CANFD frame received 1: CANFD frame received	R
7:3	—	These bits are read as 0.	R
9:8	RMIFL[1:0]	RX Message Buffer Information Label Field	R
15:10	—	These bits are read as 0.	R
31:16	RMPTR[15:0]	RX Message Buffer Pointer Field	R

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is not available in the classical CAN function.

The RX Message Buffer CANFD Status Register b (b = 0 to 15) show the status of the FDF, BRS and ESI bits, and pointer of the received CANFD frame.

RMESI bit (Error State Indicator bit)

The RMESI bit has the same value as the ESI bit of the received CANFD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RMBRS bit (Bit Rate Switch bit)

The RMBRS bit has the same value as the BRS bit of the received CANFD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RMFDF bit (CAN FD Format bit)

The RMFDF bit has the same value as the FDF bit of the received CANFD frame.

Note: This bit is not available in the classical CAN function.

RMIFL[1:0] bits (RX Message Buffer Information Label Field)

The RMIFL[1:0] bits store the information label value from the related Global Acceptance Filter List entry.

RMPTR[15:0] bits (RX Message Buffer Pointer Field)

The RMPTR[15:0] bits store the pointer value from the related Global Acceptance Filter List entry.

34.2.61.6 CFDRMDFb_p : RX Message Buffer Data Field p Registers (p = 0 to 15, b = 0 to 15)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x092C + 0x004C × b + 0x0004 × p (b = 0 to 7 , p = 0 to 15)
0x0D2C + 0x004C × (b - 8) + 0x0004 × p (b = 8 to 15 , p = 0 to 15)

Bit position:	31 … 24	23 … 16	15 … 8	7 … 0
Bit field:	RMDB_HH[7:0]	RMDB_HL[7:0]	RMDB_LH[7:0]	RMDB_LL[7:0]
Value after reset:	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	RMDB_LL[7:0]	RX Message Buffer Data Byte (p × 4)	R
15:8	RMDB_LH[7:0]	RX Message Buffer Data Byte ((p × 4) + 1)	R
23:16	RMDB_HL[7:0]	RX Message Buffer Data Byte ((p × 4) + 2)	R
31:24	RMDB_HH[7:0]*1	RX Message Buffer Data Byte ((p × 4) + 3)	R

Note: S-TYPE-3, P-TYPE-3
Note 1. These bits are not available in the classical CAN function.

The RX Message Buffer Data Field p Register b (p = 0 to 15, b = 0 to 15) store the data bytes (p × 4) to data bytes ((p × 4) + 3) of the received message.

RMDB_LL[7:0] bits (RX Message Buffer Data Byte (p × 4))

The RMDB_LL[7:0] bits store data bytes (p × 4) of the message in the RX message buffer.

Unused data bytes are filled with 0x00.

RMDB_LH[7:0] bits (RX Message Buffer Data Byte ((p × 4) + 1))

The RMDB_LH[7:0] bits store data bytes ((p × 4) + 1) of the message in the RX message buffer.

Unused Data Bytes will be filled with 0x00.

RMDB_HL[7:0] bits (RX Message Buffer Data Byte ((p × 4) + 2))

The RMDB_HL[7:0] bits store data bytes ((p × 4) + 2) of the message in the RX message buffer.

Unused data bytes are filled with 0x00.

RMDB_HH[7:0] bits (RX Message Buffer Data Byte ((p × 4) + 3))

The RMDB_HH[7:0] bits store data bytes ((p × 4) + 3) of the message in the RX message buffer.

Unused data bytes are filled with 0x00.

34.2.61.7 CFDRFMBCPb[0] : RX FIFO Access Message Buffer Component b (b = 0 to 1)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: see Table 34.4

Bit position:	31 … 0
Bit field:	Rc[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	Rc[31:0]	RX FIFO Access Message Buffer Component c See Table 34.7, Table 34.8 and the descriptions that follow for a detailed description of each register and its related bits, contained within this message buffer component.	R

Note: S-TYPE-3, P-TYPE-3

Where the total number of CFDRFMBCPb = 2 as shown in Figure 34.28 (c = RX FIFO Access Message Buffer Component Register index = [0…18])

Rc[31:0] bits (RX FIFO Access Message Buffer Component c)

The RX FIFO Access Message Buffer component comprises of the following registers:

- CFDRFIDb
- CFDRFPTRb
- CFDRFFDSTSb
- CFDRFDFbp

See Table 34.8 for details on how to interpret the structure of this buffer component and how to access the respective registers.

Table 34.7 RX FIFO Access Message Buffer component summary

Rc	
R0	RX FIFO Access ID Registers
R1	RX FIFO Access Pointer Register
R2	RX FIFO Access CANFD Status Registers
R3	RX FIFO Access Data Field 0 Registers
R4	RX FIFO Access Data Field 1 Registers
R5	RX FIFO Access Data Field 2 Registers
R6	RX FIFO Access Data Field 3 Registers
R7	RX FIFO Access Data Field 4 Registers
R8	RX FIFO Access Data Field 5 Registers
R9	RX FIFO Access Data Field 6 Registers
R10	RX FIFO Access Data Field 7 Registers
R11	RX FIFO Access Data Field 8 Registers
R12	RX FIFO Access Data Field 9 Registers
R13	RX FIFO Access Data Field 10 Registers
R14	RX FIFO Access Data Field 11 Registers
R15	RX FIFO Access Data Field 12 Registers
R16	RX FIFO Access Data Field 13 Registers
R17	RX FIFO Access Data Field 14 Registers
R18	RX FIFO Access Data Field 15 Registers
R[19...31]	—

Table 34.8 RX Message Buffer Component (RMBCP) Detailed (1 of 2)

Rc	p	Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0	x	CFDRMIDb	RMIDE	RMRTR	—	RMID																												

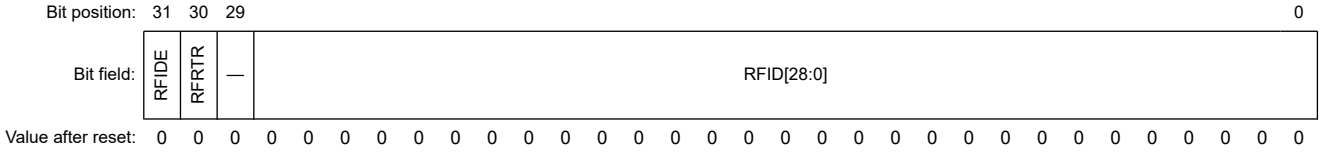
Table 34.8 RX Message Buffer Component (RMBCP) Detailed (2 of 2)

Rc	p	Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R1	x	CFDRM PTRb	RMDLC				—	—	—	—	—	—	—	—	—	—	—	—	RMTS															
R2	x	CFDRM FDSTSb	RMPTR																—	—	—	—	—	—	RMIFL		—	—	—	—	—	RMFDF	RMBRS	RMESI
R3	0	CFDRM DFbp	RMDB_HH								RMDB_HL								RMDB_LH								RMDB_LL							
R[4… 18]	[1… 15]	CFDRM DFbp	RMDB_HH								RMDB_HL								RMDB_LH								RMDB_LL							

34.2.61.8 CFDRFIDb : RX FIFO Access ID Register b (b = 0 to 1)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0520 + 0x004C × b



Bit	Symbol	Function	R/W
28:0	RFID[28:0]	RX FIFO Buffer ID Field STD-ID/EXT-ID fields	R
29	—	This bit is read as 0.	R
30	RFRTR	RX FIFO Buffer RTR bit 0: Data frame 1: Remote frame	R
31	RFIDE	RX FIFO Buffer IDE bit 0: STD-ID has been received 1: EXT-ID has been received	R

Note: S-TYPE-3, P-TYPE-3

The RX FIFO Access ID Registers b (b = 0 to 1) store the ID field, IDE bit and RTR bit of the message.

RFID[28:0] bits (RX FIFO Buffer ID Field)

The RFID[28:0] bits are the bits of the STD-ID/EXT-ID fields of the message in the FIFO buffer.

For alignment of these bits in standard and extended frame format, see Identifier Bits Alignment.

RFRTR bit (RX FIFO Buffer RTR bit)

The RFRTR bit shows whether a data frame or a remote frame was stored in the FIFO buffer.

Note: There are no remote frames in CANFD format. When a CANFD frame was received, the register reflects the state of the received value (RRS bit in FD frame format).

RFIDE bit (RX FIFO Buffer IDE bit)

The RFIDE bit shows whether message with the Standard Identifier or Extended Identifier was received in the FIFO buffer.

34.2.61.9 CFDRFPTRb : RX FIFO Access Pointer Register b (b = 0 to 1)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0524 + 0x004C × b

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RFDLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	RFTS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RFTS[15:0]	RX FIFO Timestamp Value Timestamp value of the received CAN frame	R
27:16	—	These bits are read as 0.	R
31:28	RFDLC[3:0]	RX FIFO Buffer DLC Field Number of data bytes received in a CAN frame	R

Note: S-TYPE-3, P-TYPE-3

The FIFO Access Pointer Registers b (b = 0 to 1) store the DLC and Timestamp fields for the received message.

RFTS[15:0] bits (RX FIFO Timestamp Value)

The RFTS[15:0] bits store the timestamp value taken at the capture point as configured by the CFDGFDCFG.TSCCFG bit of the received message.

RFDLC[3:0] bits (RX FIFO Buffer DLC Field)

The RFDLC[3:0] bits store the number of data bytes that were received in the RX FIFO buffer.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes that were received.

34.2.61.10 CFDRFFDSTSb : RX FIFO Access CANFD Status Register b (b = 0 to 1)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0528 + 0x004C × b

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CFDRFPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	RFIFL[1:0]		—	—	—	—	—	RFFDF	RFBRS	RFESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFESI*1	Error State Indicator bit 0: CANFD frame received from error active node 1: CANFD frame received from error passive node	R
1	RFBRS*1	Bit Rate Switch bit 0: CANFD frame received with no bit rate switch 1: CANFD frame received with bit rate switch	R
2	RFFDF*1	CAN FD Format bit 0: Non CANFD frame received 1: CANFD frame received	R
7:3	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
9:8	RFIFL[1:0]	RX FIFO Buffer Information Label Field	R
15:10	—	These bits are read as 0.	R
31:16	CFDRFPTR[15:0]	RX FIFO Buffer Pointer Field	R

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is not available in the classical CAN function.

The RX FIFO Access CANFD Status Registers b (b = 0 to 1) show the status of the FDF, BRS, and ESI bits, including the pointer of the received CANFD frame.

RFESI bit (Error State Indicator bit)

The RFESI bit has the same value as the ESI bit of the received CANFD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RFBRS bit (Bit Rate Switch bit)

The RFBRS bit has the same value as the BRS bit of the received CANFD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RFFDF bit (CAN FD Format bit)

The RFFDF bit has the same value as the FDF bit of the received CANFD frame.

Note: This bit is not available in the classical CAN function.

RFIFL[1:0] bits (RX FIFO Buffer Information Label Field)

The RFIFL[1:0] bits store the information label value from the related Global Acceptance Filter List entry.

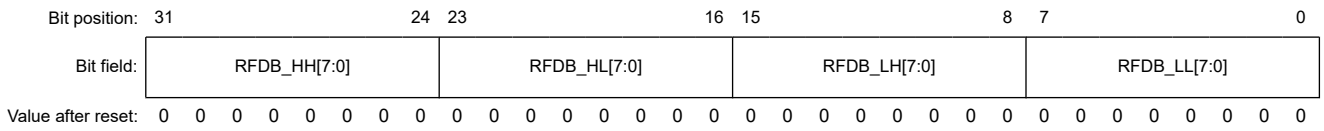
CFDRFPTR[15:0] bits (RX FIFO Buffer Pointer Field)

The CFDRFPTR[15:0] bits store the pointer value from the related Global Acceptance Filter List entry.

34.2.61.11 CFDRFDFb_p : RX FIFO Access Data Field p Register b (p = 0 to 15, b = 0 to 1)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x052C + 0x004 × p + 0x04C × b



Bit	Symbol	Function	R/W
7:0	RFDB_LL[7:0]	RX FIFO Buffer Data Byte (p × 4)	R
15:8	RFDB_LH[7:0]	RX FIFO Buffer Data Byte ((p × 4) + 1)	R
23:16	RFDB_HL[7:0]	RX FIFO Buffer Data Byte ((p × 4) + 2)	R
31:24	RFDB_HH[7:0]	RX FIFO Buffer Data Byte ((p × 4) + 3)	R

Note: S-TYPE-3, P-TYPE-3

The RX FIFO Access Data Field p Registers b (p = 0 to 15, b = 0 to 1) store data bytes ((p × 4) to data byte ((p × 4) + 3) of the received message.

RFDB_LL[7:0] bits (RX FIFO Buffer Data Byte (p × 4))

The RFDB_LL[7:0] bits store data bytes (p × 4) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00 according to the configured data payload size CFDRFCCa.RFPLS.

RFDB_LH[7:0] bits (RX FIFO Buffer Data Byte ((p × 4) + 1))

The RFDB_LH[7:0] bits store data bytes ((p × 4) + 1)) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00.

RFDB_HL[7:0] bits (RX FIFO Buffer Data Byte ((p × 4) + 2))

The RFDB_HL[7:0] bits store data bytes ((p × 4) + 2) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00.

RFDB_HH[7:0] bits (RX FIFO Buffer Data Byte ((p × 4) + 3))

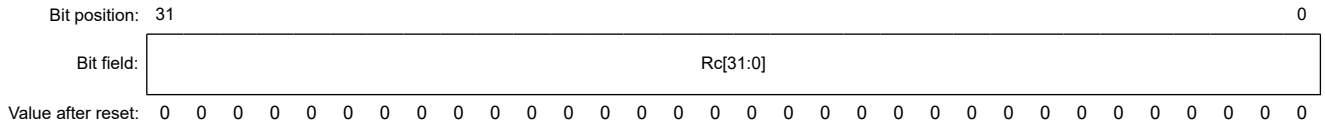
The RFDB_HH[7:0] bits store data bytes ((p × 4) + 3) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00.

34.2.61.12 CFDCFMBCP0[0] : Common FIFO Access Message Buffer Component

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: See Table 34.4



Bit	Symbol	Function	R/W
31:0	Rc[31:0]	Common FIFO Access Message Buffer Component c Refer to Table 34.9, Table 34.10 and the descriptions that follow for a detailed description of each register and its related bits, contained within this message buffer component.	R

Note: S-TYPE-3, P-TYPE-3

Where the total number of CFDCFMBCP0 = 1 as shown in Figure 34.28 (c = Common FIFO Message Buffer Component Register index = [0…18])

Rc[31:0] bit (Common FIFO Access Message Buffer Component c)

The Common FIFO Access Message Buffer Component is made up of the following registers: CFDCFID, CFDCFPTR, CFFDSTS0, and CFDCFDFp. Refer to Table 34.10 for details of how to interpret the structure of this buffer component and how to access the respective registers.

Table 34.9 Common FIFO Access Message Buffer Component Summary (1 of 2)

Common FIFO Access Message Buffer Component (CFMBCP)	
Rc	**CANFD mode (CAN_FD_MODE = 1)**
R0	Common FIFO Access ID Registers
R1	Common FIFO Access Pointer Register
R2	Common FIFO Access CANFD Status Registers
R3	Common FIFO Access Data Field 0 Registers
R4	Common FIFO Access Data Field 1 Registers
R5	Common FIFO Access Data Field 2 Registers
R6	Common FIFO Access Data Field 3 Registers
R7	Common FIFO Access Data Field 4 Registers

Table 34.9 Common FIFO Access Message Buffer Component Summary (2 of 2)

Common FIFO Access Message Buffer Component (CFMBCP)	
Rc	**CANFD mode (CAN_FD_MODE = 1)**
R8	Common FIFO Access Data Field 5 Registers
R9	Common FIFO Access Data Field 6 Registers
R10	Common FIFO Access Data Field 7 Registers
R11	Common FIFO Access Data Field 8 Registers
R12	Common FIFO Access Data Field 9 Registers
R13	Common FIFO Access Data Field 10 Registers
R14	Common FIFO Access Data Field 11 Registers
R15	Common FIFO Access Data Field 12 Registers
R16	Common FIFO Access Data Field 13 Registers
R17	Common FIFO Access Data Field 14 Registers
R18	Common FIFO Access Data Field 15 Registers
R[19…31]	—

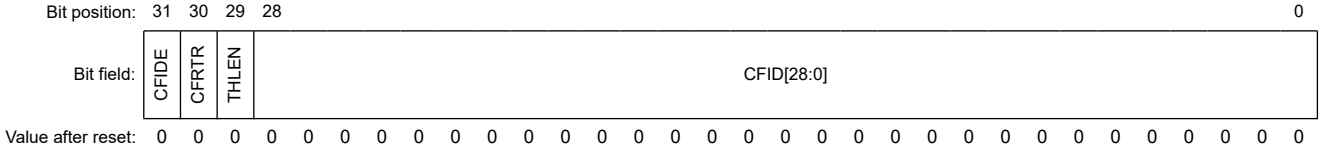
Table 34.10 Common FIFO Access Message Buffer Component (CFMBCP) Detailed

Rc	p	Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0	x	CFDCFID	CFIDE	CFRTR	THLEN	CFID																												
R1	x	CFDCFPTR	CFDLC				—	—	—	—	—	—	—	—	—	—	—	—	CFTS															
R2	x	CFDCFFDCSTS	CFPTR																—	—	—	—	—	—	CFIFL		—	—	—	—	—	CFFDF	CFBRS	CFESI
R3	0	CFDCFDFp	CFDB_HH								CFDB_HL								CFDB_LH								CFDB_LL							
R[4…18]	[1…15]	CFDCFDFp	CFDB_HH								CFDB_HL								CFDB_LH								CFDB_LL							
R[19…31]	x	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

34.2.61.13 CFDCFID : Common FIFO Access ID Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x05B8



Bit	Symbol	Function	R/W
28:0	CFID[28:0]	Common FIFO Buffer ID Field STD-ID / EXT-ID fields	R/W

Bit	Symbol	Function	R/W
29	THLEN	THL Entry enable TX FIFO Mode: 0: Entry will not be stored in THL after successful TX. 1: Entry will be stored in THL after successful TX. RX FIFO Mode: Reserved, this bit is read as 0	R/W
30	CFRTR	Common FIFO Buffer RTR Bit 0: Data Frame 1: Remote Frame	R/W
31	CFIDE	Common FIFO Buffer IDE Bit 0: STD-ID will be transmitted or has been received 1: EXT-ID will be transmitted or has been received	R/W

Note: S-TYPE-3, P-TYPE-3

The Common FIFO Access ID registers store the ID field, IDE bit and RTR bit of the message.

In TX mode, users can read data from the FIFO, only for the current entry based on the write pointer value, not for the other entries.

CFID[28:0] bit (Common FIFO Buffer ID Field)

These are the bits of the STD-ID / EXT-ID fields of the message in the FIFO Buffer.

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers.

THLEN bit (THL Entry enable)

This bit controls the storage of an entry corresponding to the transmitted message in the TX History list at the end of a successful transmission.

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers.

CFRTR bit (Common FIFO Buffer RTR Bit)

This bit selects whether a Data Frame or a Remote Frame will be transmitted from or was received in the FIFO Buffer.

Note: There are no remote frames in CANFD format. In case a CANFD frame was received (RX mode) the register reflects the state of the received value (RRS bit in FD frame format). In case of CANFD transmission (TX mode CFDCFID.CFFDF =1) the bit is always transmitted dominant (Data Frame).

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers.

CFIDE bit (Common FIFO Buffer IDE Bit)

This bit selects whether a message with EXT-ID or STD-ID will be transmitted from or was received in the FIFO Buffer.

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers.

34.2.61.14 CFDCFPTR : Common FIFO Access Pointer Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x05BC

Bit position:	31			28												15														0
Bit field:	CFDLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	CFTS[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CFTS[15:0]	Common FIFO Timestamp Value Timestamp value of the received CAN frame (FIFO in RX mode).	R/W
27:16	—	These bits are read as 0. The write value should be 0.	R/W
31:28	CFDLC[3:0]	Common FIFO Buffer DLC Field Number of data bytes received in a CAN frame, or to be transmitted in a CAN frame.	R/W

Note: S-TYPE-3, P-TYPE-3

The Common FIFO Access Pointer Registers store the DLC and Timestamp fields.

In TX mode, you can read data from the FIFO buffer, only for the current entry based on the write pointer value, and not for the other entries.

CFTS[15:0] bits (Common FIFO Timestamp Value)

The CFTS[15:0] bits store the timestamp value taken at the capture point as configured by the CFDGFDCFG.TSCCFG bit of the received message (if FIFO is configured in RX mode).

In TX mode, you can read and write from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

CFDLC[3:0] bits (Common FIFO Buffer DLC Field)

The CFDLC[3:0] bits store the number of data bytes that were received in the FIFO buffer or are to be transmitted.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes.

In TX mode, you can read and write from the FIFO buffers. Do not read data for the other entries in the FIFO when configured in TX mode.

In RX mode, you can only read data from the FIFO buffers.

34.2.61.15 CFDCFFDCSTS : Common FIFO Access CANFD Control/Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x05C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CFPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CFIFL[1:0]		—	—	—	—	—	CFFDF	CFBRS	CFESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CFESI*1	Error State Indicator bit 0: CANFD frame received or to transmit by error active node 1: CANFD frame received or to transmit by error passive node	R/W
1	CFBRS*1	Bit Rate Switch bit 0: CANFD frame received or to transmit with no bit rate switch 1: CANFD frame received or to transmit with bit rate switch	R/W
2	CFFDF*1	CAN FD Format bit 0: Non CANFD frame received or to transmit 1: CANFD frame received or to transmit	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
9:8	CFIFL[1:0]	COMMON FIFO Buffer Information Label Field	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
31:16	CFPTR[15:0]	Common FIFO Buffer Pointer Field	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is not available in the classical CAN function.

The Common FIFO Access CANFD Control/Status Registers show the status of the FDF, BRS and ESI bits, including the pointer of the received CANFD frame or the CANFD frame to transmit.

In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

CFESI bit (Error State Indicator bit)

In TX mode, you can read and write from FIFO buffers. In this mode, when the CANFD module is not in error passive, the CFESI bit equals the write value. Otherwise, it is a don't care and the bit is transmitted as 1 on the CAN bus, indicating this is an error passive node.

In RX mode, you can only read data from FIFO buffers.

In RX mode, the CFESI bit is updated with the ESI bit value of the CANFD frame when it has been received, indicating the error state of the transmitting node. In RX mode, 0 is stored to this bit when the received FDF bit is 0, this means a CAN 2.0 frame is received.

Note: This bit is not available in the classical CAN function.

CFBRS bit (Bit Rate Switch bit)

In TX mode, you can read and write from FIFO buffers. In this mode, the CANFD module either transmits a 0 to indicate no bit rate switch in the frame to be transmitted or a 1 to indicate a bit rate switch in the frame to be transmitted.

In RX mode, you can only read data from FIFO buffers.

In RX mode, the CFBRS bit is updated with the BRS bit value of the CANFD frame when it has been received, indicating whether there is a bit rate switch (1) or (0) on the CANFD frame.

In RX mode, 0 is stored to the CFBRS bit when the received FDF bit is 0, this means a CAN 2.0 frame is received.

Note: This bit is not available in the classical CAN function.

CFFDF bit (CAN FD Format bit)

In TX mode, you can read and write from FIFO buffers. In this mode, the CANFD module either transmits a 0 to indicate a CAN 2.0 frame is to be transmitted or a 1 to indicate a CANFD frame is to be transmitted.

In RX mode, you can only read data from FIFO buffers.

In RX mode, the CFFDF bit is updated with the FDF bit value of the CAN frame when it has been received, indicating whether it is a CAN 2.0 frame (0) or a CANFD frame (1).

Note: This bit is not available in the classical CAN function.

CFIFL[1:0] bits (COMMON FIFO Buffer Information Label Field)

If the Common FIFO is configured in TX mode, the value programmed in CFDCFFDCSTS.CFIFL[1:0] is stored together with additional message information, to the TX History List after successful transmission of the message.

The information label value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX mode).

In TX mode, you can read and write from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

CFPTR[15:0] bits (Common FIFO Buffer Pointer Field)

If the Common FIFO is configured in TX mode, the value programmed in CFDCFFDCSTS.CFPTR[15:0] is stored together with additional message information, to the TX History List after successful transmission of the message.

The pointer value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX mode).

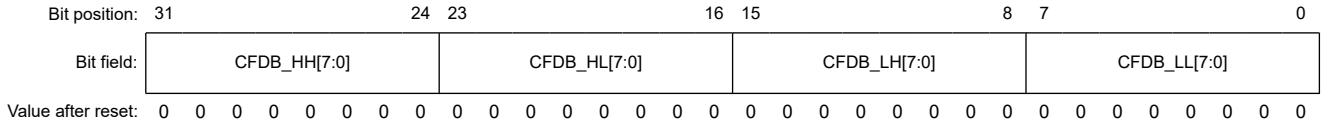
In TX mode, you can read and write from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

34.2.61.16 CFDCFDFp : Common FIFO Access Data Field p Registers (p = 0 to 15)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x05C4 + 0x004 × p



Bit	Symbol	Function	R/W
7:0	CFDB_LL[7:0]	Common FIFO Buffer Data Bytes (p × 4)	R/W
15:8	CFDB_LH[7:0]	Common FIFO Buffer Data Bytes ((p × 4) + 1)	R/W
23:16	CFDB_HL[7:0]	Common FIFO Buffer Data Bytes ((p × 4) + 2)	R/W
31:24	CFDB_HH[7:0]	Common FIFO Buffer Data Bytes ((p × 4) + 3)	R/W

Note: S-TYPE-3, P-TYPE-3

The FIFO Access Data Field p Registers (p = 0 to 15) store data bytes (p × 4) to data bytes ((p × 4) + 3) of the message.

In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

CFDB_LL[7:0] bits (Common FIFO Buffer Data Bytes (p × 4))

The CFDB_LL[7:0] bits store data bytes (p × 4) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In RX mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCC.CFPLS.*1

CFDB_LH[7:0] bits (Common FIFO Buffer Data Bytes ((p × 4) + 1))

The CFDB_LH[7:0] bits store data bytes ((p × 4) + 1) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In RX mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCC.CFPLS.*1

CFDB_HL[7:0] bits (Common FIFO Buffer Data Bytes ((p × 4) + 2))

The CFDB_HL[7:0] bits store data bytes ((p × 4) + 2) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In RX mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCC.CFPLS.*1

CFDB_HH[7:0] bits (Common FIFO Buffer Data Bytes ((p × 4) + 3))

The CFDB_HH[7:0] bits store data bytes ((p × 4) + 3) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

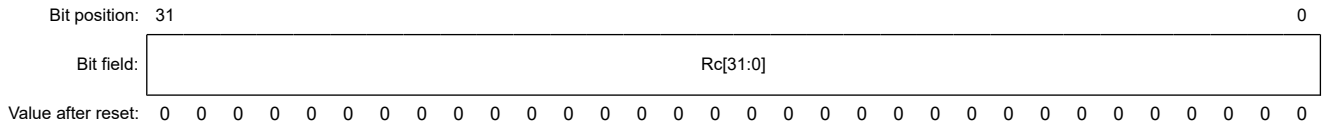
In RX mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCC.CFPLS.*1

Note 1. In RX mode, unused data bytes are filled with 0x00 according to the configured data payload size CFDCFCC.CFPLS, which is a CANFD feature not found in classical CAN.

34.2.61.17 CFDTMBCPb[0] : TX Message Buffer Component b (b = 0 to 3)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: See Table 34.4



Bit	Symbol	Function	R/W
31:0	Rc[31:0]	TX Message Buffer Component c Refer to Table 34.11, Table 34.12 and the descriptions that follow for a detailed description of each register and its related bits, contained within this message buffer component.	R

Note: S-TYPE-3, P-TYPE-3

Where the total number of CFDTMBCPn = 4 as shown in Figure 34.28 (c = TX Message Buffer Component Register index = [0…18])

Rc[31:0] bit (TX Message Buffer Component c)

TX Message Buffer Component c

The TX Message Buffer Component is made up of the following registers: CFDTMIDb, CFDTMPTRb, CFDTMFDCTRb, and CFDTMDFbp. Refer to Table 34.12 for details of how to interpret the structure of this buffer component and how to access the respective registers.

Table 34.11 TX Message Buffer Component Summary

TX Message Buffer Component (TMBCP)	
Rc	**CANFD mode (CAN_FD_MODE = 1b)**
R0	TX Message Buffer (b) ID Registers
R1	TX Message Buffer (b) Pointer Registers
R2	TX Message Buffer (b) CANFD Status Registers
R3	TX Message Buffer (b) Data Field 0 Registers
R4	TX Message Buffer (b) Data Field 1 Registers
R5	TX Message Buffer (b) Data Field 2 Registers
R6	TX Message Buffer (b) Data Field 3 Registers
R7	TX Message Buffer (b) Data Field 4 Registers
R8	TX Message Buffer (b) Data Field 5 Registers
R9	TX Message Buffer (b) Data Field 6 Registers
R10	TX Message Buffer (b) Data Field 7 Registers
R11	TX Message Buffer (b) Data Field 8 Registers
R12	TX Message Buffer (b) Data Field 9 Registers
R13	TX Message Buffer (b) Data Field 10 Registers
R14	TX Message Buffer (b) Data Field 11 Registers
R15	TX Message Buffer (b) Data Field 12 Registers
R16	TX Message Buffer (b) Data Field 13 Registers
R17	TX Message Buffer (b) Data Field 14 Registers
R18	TX Message Buffer (b) Data Field 15 Registers
R[19…31]	—

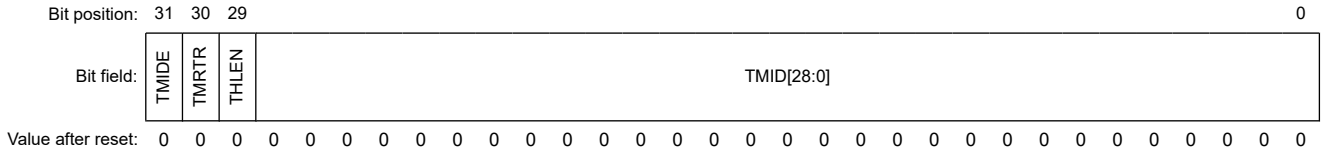
Table 34.12 TX Message Buffer Component (TMBCP) Detailed

Rc	p	Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0	x	CFDTMIDb	TMIDE	TMRTR	THLEN	TMID																												
R1	x	CFDTMPTRb	TMDLC				—	—	—	—	—	—	—	—	—	—	—	—	CFTS															
R2	x	CFDTMFDCTRb	TMPTR																—	—	—	—	—	—	TMIFL		—	—	—	—	—	TMFDF	TMBRS	TMESI
R3	0	CFDTMDFbp	TMDB_HH								TMDB_HL								TMDB_LH								TMDB_LL							
R[4…18]	[1…15]	CFDTMDFbp	TMDB_HH								TMDB_HL								TMDB_LH								TMDB_LL							

34.2.61.18 CFDTMIDb : TX Message Buffer ID Registers (b = 0 to 3)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0604 + 0x004C × b



Bit	Symbol	Function	R/W
28:0	TMID[28:0]	TX Message Buffer ID Field STD-ID/EXT-ID fields	R/W
29	THLEN	Tx History List Entry 0: Entry not stored in THL after successful TX 1: Entry stored in THL after successful TX	R/W
30	TMRTR	TX Message Buffer RTR bit 0: Data frame 1: Remote frame	R/W
31	TMIDE	TX Message Buffer IDE bit 0: STD-ID is transmitted 1: EXT-ID is transmitted	R/W

Note: S-TYPE-3, P-TYPE-3

Each TX Message Buffer ID Register b (b = 0 to 3) are used to store the ID, IDE, RTR fields and history configuration of the message to be transmitted from the associated buffer.

TMID[28:0] bits (TX Message Buffer ID Field)

The TMID[28:0] bits are bits of the STD-ID/EXT-ID fields of the message stored in this TX message buffer.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

THLEN bit (Tx History List Entry)

The THLEN bit controls the storage of an entry corresponding to the transmitted message in the TX History list at the end of a successful transmission.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

TMRTR bit (TX Message Buffer RTR bit)

The TMRTR bit selects whether a data frame or remote frame is to be transmitted from this TX message buffer.

Note: There are no remote frames in CANFD format. For a CANFD transmission (CFDTMFDCTRb.CFFDF = 1), this bit is always transmitted dominant (data frame).

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

TMIDE bit (TX Message Buffer IDE bit)

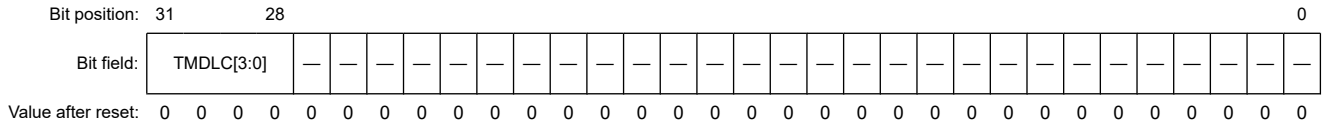
The TMIDE bit selects whether a message with EXT-ID or STD-ID is to be transmitted from this TX message buffer.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

34.2.61.19 CFDTMPTRb : TX Message Buffer Pointer Register (b = 0 to 3)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0608 + 0x004C × b



Bit	Symbol	Function	R/W
27:0	—	The read values are undefined. The write value should be 0.	R/W
31:28	TMDLC[3:0]	TX Message Buffer DLC Field Number of data bytes to be transmitted in a CAN frame.	R/W

Note: S-TYPE-3, P-TYPE-3

Each TX Message Buffer Pointer Register b (b = 0 to 3) is used to store the DLC fields of the message to transmit from the associated buffer.

TMDLC[3:0] bits (TX Message Buffer DLC Field)

The TMDLC[3:0] bits select the number of data bytes to be transmitted from this TX message buffer when the corresponding TMRTR bit is configured as 0.

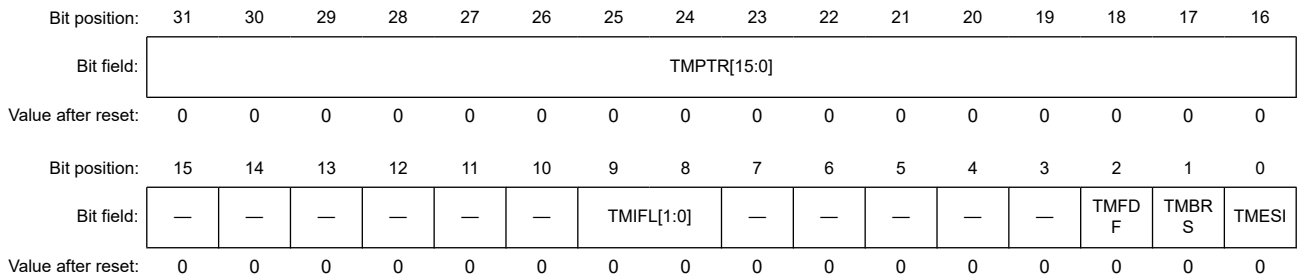
See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes to be transmitted.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

34.2.61.20 CFDTMFDCTRb : TX Message Buffer CANFD Control Register (b = 0 to 3)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x060C + 0x004C × b



Bit	Symbol	Function	R/W
0	TMESI*1	Error State Indicator bit 0: CANFD frame to transmit by error active node 1: CANFD frame to transmit by error passive node	R/W
1	TMBRS*1	Bit Rate Switch bit 0: CANFD frame to transmit with no bit rate switch 1: CANFD frame to transmit with bit rate switch	R/W
2	TMFDF*1	CAN FD Format bit 0: Non CANFD frame to transmit 1: CANFD frame to transmit	R/W
7:3	—	The read values are undefined. The write value should be 0.	R/W
9:8	TMIFL[1:0]	TX Message Buffer Information Label Field	R/W
15:10	—	The read values are undefined. The write value should be 0.	R/W
31:16	TMPTR[15:0]	TX Message Buffer Pointer Field	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is not available in the classical CAN function.

The TX Message Buffer CANFD Control Registers b (b = 0 to 3) show the status of the FDF, BRS and ESI bits, including the pointer fields of the CANFD frame to be transmitted.

TMESI bit (Error State Indicator bit)

If the channel is not in error passive, then the TMESI bit equals the write value, otherwise it is a don't care and the bit is transmitted as 1 on the CAN bus, indicating this is an error passive node.

Do not write to the TMESI bit when the related CANFD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

TMBRS bit (Bit Rate Switch bit)

Do not write to the TMBRS bit when the related CANFD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

TMFDF bit (CAN FD Format bit)

Do not write to the TMFDF bit when the related CANFD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

TMIFL[1:0] bits (TX Message Buffer Information Label Field)

The TMIFL[1:0] bits store the information label value to be copied, together with additional message information, in the TX History List after successful transmission of the message.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

TMPTR[15:0] bits (TX Message Buffer Pointer Field)

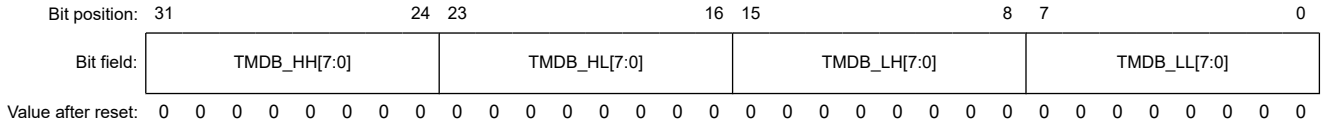
The TMPTR[15:0] bits store the pointer value to be copied, together with additional message information in the TX History List after successful transmission of the message.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

34.2.61.21 CFDTMDFb_p : TX Message Buffer Data Field Register (p= 0 to 15 , b= 0 to 3)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0610 + 0x004 × p + 0x004C × b



Bit	Symbol	Function	R/W
7:0	TMDB_LL[7:0]	TX Message Buffer Data Byte (p × 4)	R/W
15:8	TMDB_LH[7:0]	TX Message Buffer Data Byte ((p × 4) + 1)	R/W
23:16	TMDB_HL[7:0]	TX Message Buffer Data Byte ((p × 4) + 2)	R/W
31:24	TMDB_HH[7:0]	TX Message Buffer Data Byte ((p × 4) + 3)	R/W

Note: S-TYPE-3, P-TYPE-3

Each TX Message Buffer Data Field p Register b (p = 0 to 15, b = 0 to 3) is used to store data bytes (p × 4) to data bytes ((p × 4) + 3) of the message to transmit from the associated buffer.

TMDB_LL[7:0] bits (TX Message Buffer Data Byte (p × 4))

TMDB_LL[7:0] bits store data bytes (p × 4) of the message in the TX message buffer.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

TMDB_LH[7:0] bits (TX Message Buffer Data Byte ((p × 4) + 1))

TMDB_LH[7:0] bits store data bytes ((p × 4) + 1) of the message in the TX message buffer.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

TMDB_HL[7:0] bits (TX Message Buffer Data Byte ((p × 4) + 2))

TMDB_HL[7:0] bits store data bytes ((p × 4) + 2) of the message in the TX message buffer.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

TMDB_HH[7:0] bits (TX Message Buffer Data Byte ((p × 4) + 3))

TMDB_HH[7:0] bits store data bytes ((p × 4) + 3) of the message in the TX message buffer.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

34.3 Modes of Operation

34.3.1 Overview

The modes of the CANFD module can be classified into 2 groups:

- Global modes
- Channel modes

34.3.2 Global Modes

These modes are applicable for the complete CANFD module and therefore are called Global modes. The global modes of the CANFD module are:

- Global Sleep
- Global Reset
- Global Halt

- Global Operation.

Figure 34.2 shows the possible transitions between the Global modes.

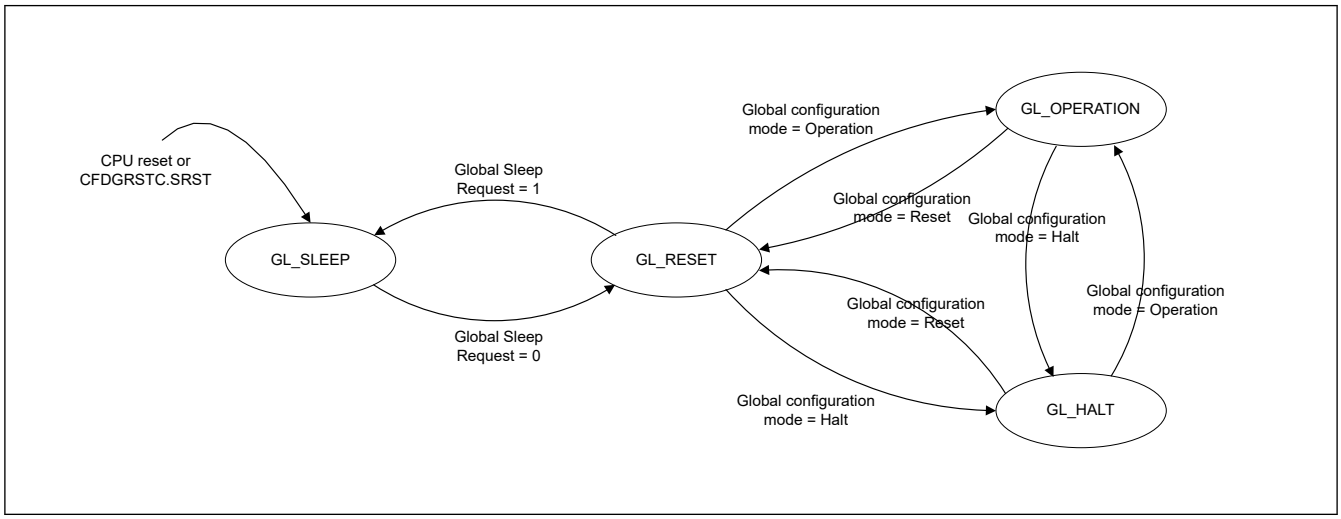


Figure 34.2 Transition between CANFD Global modes

Change in the Global mode can affect the Channel mode. Table 34.13 shows the effect of a Global mode transition on a Channel mode.

Table 34.13 Possible CANFD Channel modes and Global modes

Current Global mode	Target Global mode			
	Sleep	**Reset**	**Halt**	**Operation**
Sleep		Ch-Sleep: Keep Ch-Reset: N/A Ch-Halt: N/A Ch-Oper: N/A		
Reset	Ch-Sleep: Keep Ch-Reset: → Ch-Sleep Ch-Halt: N/A Ch-Oper: N/A		Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: N/A Ch-Oper: N/A	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: N/A Ch-Oper: N/A
Halt		Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: → Ch-Reset Ch-Oper: N/A		Ch-Sleep:Keep Ch-Reset: Keep Ch-Halt: Keep Ch-Oper: N/A
Operation		Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: → Ch-Reset Ch-Oper: → Ch-Reset	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: Keep Ch-Oper: → Ch-Halt	

34.3.2.1 Global Sleep Mode

After the release of a hardware reset or after setting and clearing a CFDGRSTC.SRST bit, the CANFD module automatically enters Global Sleep mode.

The CANFD module also enters the Global Sleep mode when the Global Sleep Request bit is set while it is in Global Reset mode. This control bit cannot be set in Global Halt mode or Global Operation mode.

Setting the Global Sleep Request bit sets Channel Sleep Request bit and forces the channel into the Channel Sleep mode.

Sleep mode is used for power saving purpose. When CANFD module is in Global Sleep mode, only the clock for CPU write access to the Global Sleep Mode Request bit is active. All other clocks are stopped and all other functions of the CANFD module are suspended.

Read access from all registers is still possible and all register values are preserved.

After setting the Global Sleep Request bit, it is necessary to confirm that the Global Sleep status has been updated, indicating successful transition to Global Sleep mode before the Global Sleep Request bit can be cleared again.

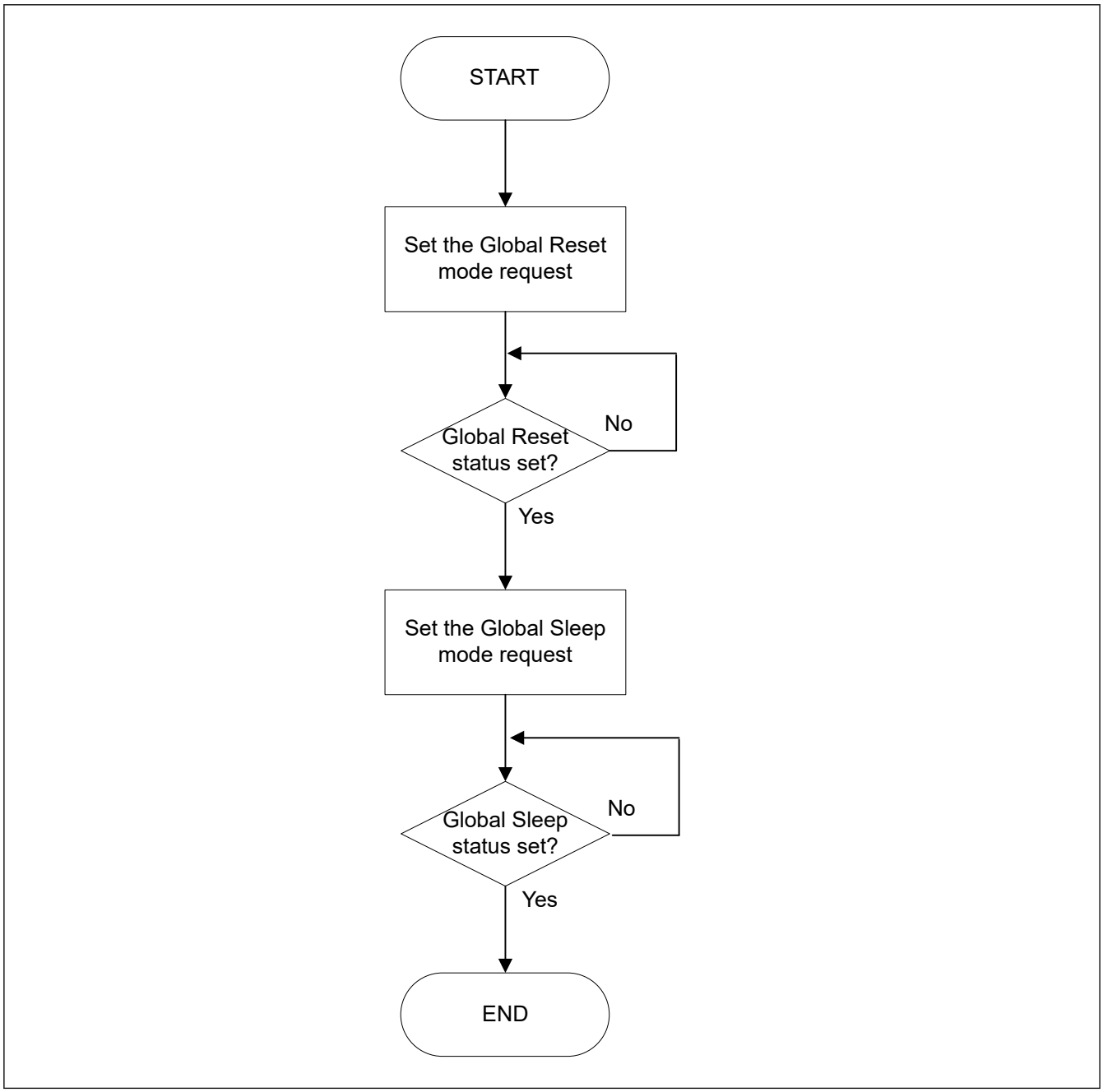


Figure 34.3 Procedure for entering Global Sleep mode

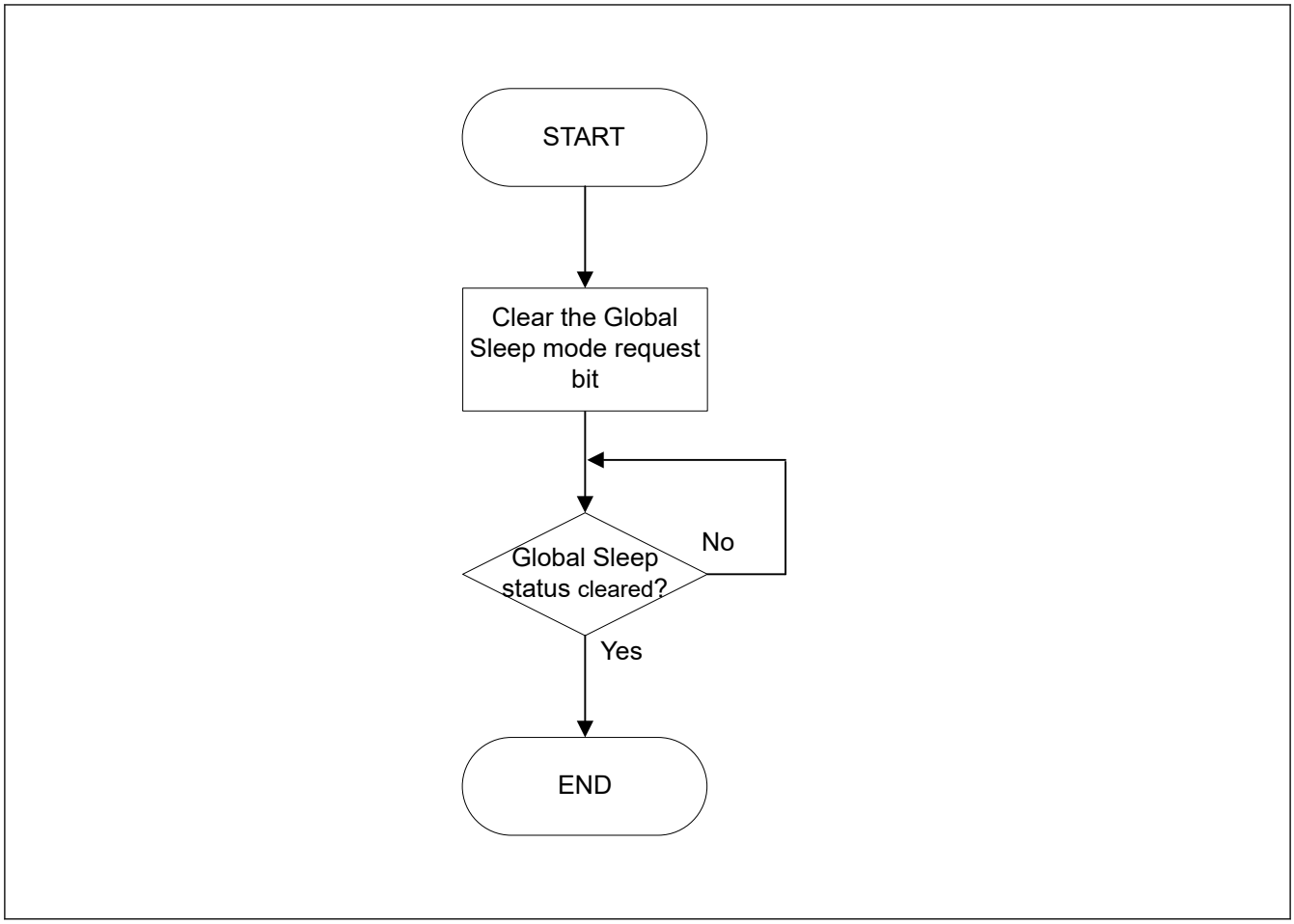


Figure 34.4 Procedure for exiting Global Sleep mode

34.3.2.2 Global Reset Mode

The CANFD module enters this mode in the following ways:

- Global Mode Control bit CFDGCTR.GMDC in the Global Control Register is configured for Global Reset mode while the CANFD module is in Global Halt or Global Operation mode
- Global Sleep Mode Request bit is cleared while CANFD module is in Global Sleep mode.

In Global Reset mode, all CANFD module functions are suspended and all status and flag registers are initialized.

Additionally all FIFOs and TX Queues are disabled and transmission control bits are cleared.

Configuration registers (except the test mode registers) are not initialized in this mode to their MCU reset values and the CANFD module can be configured.

See section 34.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Global Reset mode is performed.

Setting the Global mode to Reset by setting the Global Mode Control bits CFDGCTR.GMDC in the Global Control Register to 01b sets Channel Mode Control bits CFDC0CTR.CHMDC in the Channel Control Registers to 01b and forces the channel into the Channel Reset mode.

For channels that are already in Channel Reset mode or Channel Sleep mode, this automatic transition is not performed (CFDC0CTR.CHMDC of related channel already set to 01b).

After setting Global Mode Control bit CFDGCTR.GMDC to Reset mode, it is necessary to confirm that the Reset Mode Status bit CFDGSTS.GRSTSTS in the Global Status Register has been updated, indicating successful transition to Global Reset mode before CFDGCTR.GMDC can be changed again.

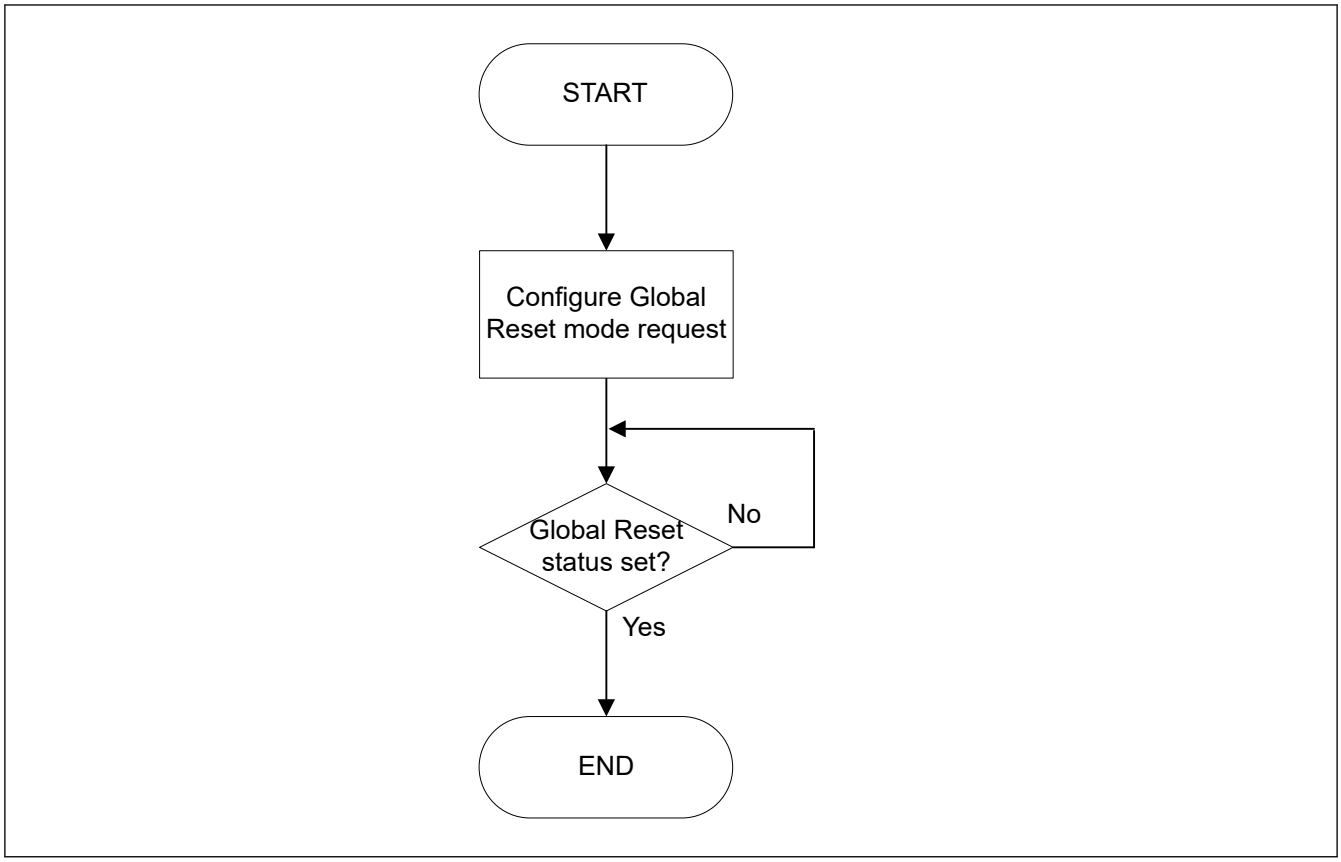


Figure 34.5 Procedure for entering Global Reset mode

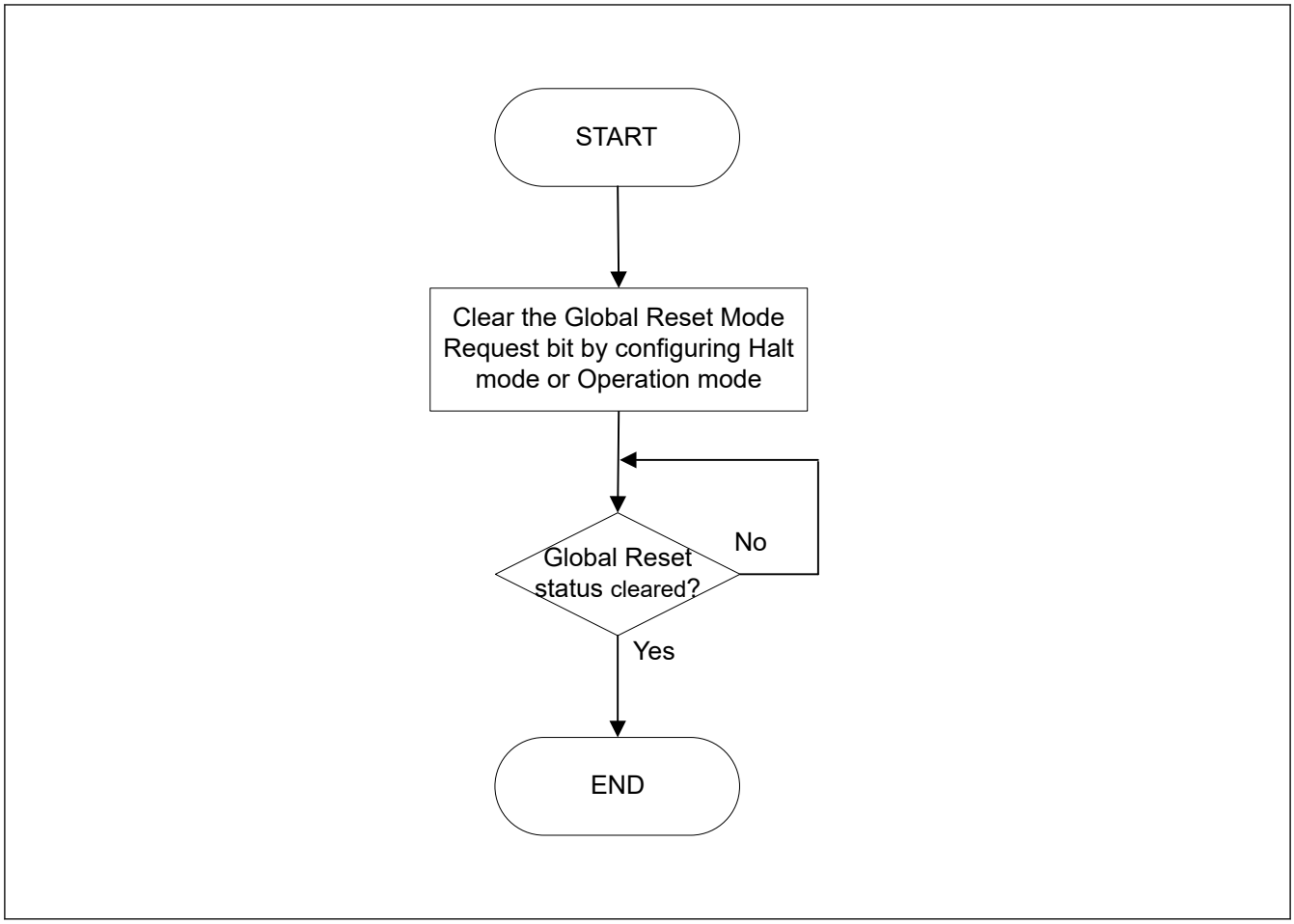


Figure 34.6 Procedure for exiting Global Reset mode

34.3.2.3 Global Halt Mode

The CANFD module enters this mode in the following ways:

- Global Mode Control bit CFDGCTR.GMDC in the Global Control Register is configured for Global Halt mode while the CANFD module is in Global Reset mode:
  - the channel in either Channel Reset or Channel Sleep mode remains in this mode
- Global Mode Control bit CFDGCTR.GMDC in the Global Control Register is configured for Global Halt mode while the CANFD module is in Global Operation mode:
  - the channel in Channel Reset, Channel Halt, or Channel Sleep mode remains in this mode
  - the channel in Channel Operation mode transitions to Channel Halt mode
  - Global Halt Mode Status bit is set when the channel has left Channel Operation mode.

If a transmission or reception is ongoing for a channel, the transition to Channel Halt mode is delayed until completion of the communication.

Similarly, if a channel is in bus-off, the full bus-off recovery sequence may be delayed depending on the channel configuration.

In Global Halt mode, all communications are suspended and CANFD logic does not cause any change to the Status and Flag registers (only when a channel is in the bus-off that its REC and TEC values are cleared). Additionally, the test mode configuration and control registers are not initialized in this mode.

The Global Halt mode should be used to configure global module test modes.

See section 34.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Global Halt mode is performed.

Setting the Global mode to Halt by setting the Global Mode Control bit CFDGCTR.GMDC in the Global Control Register to 10b sets Channel Mode Control bits CFDC0CTR.CHMDC in the Channel Control Registers to 10b for the channel that are in Channel Operation mode and forces these channels into the Channel Halt mode.

For the channel that are already in Channel Reset, Channel Halt, or Channel Sleep mode, this automatic transition is not performed.

Therefore, the Global Halt mode request can be used to shut down all CANFD channel communications without loss of messages and disruption on the related CAN bus (no interruption of reception/transmission processes on the channels).

After setting the Global Mode Control bit CFDGCTR.GMDC to Halt mode, it is necessary to confirm that the Halt Mode Status bit CFDGSTS.GHLTSTS in the Global Status Register has been updated to indicate a successful transition to Global Halt mode. Do not specify any other SFR setting until confirming CFDGSTS.GHLTSTS is set.

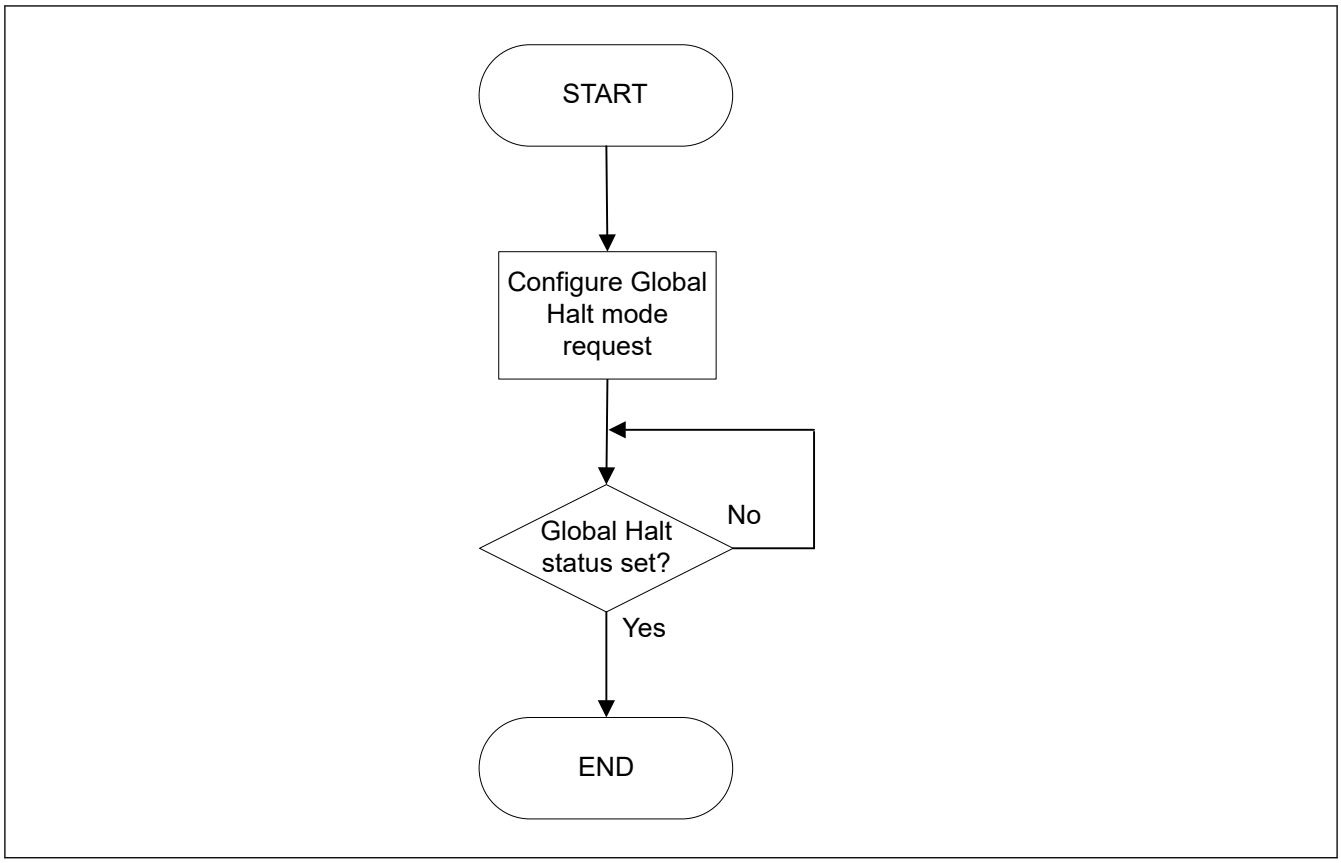


Figure 34.7 Procedure for entering Global Halt mode

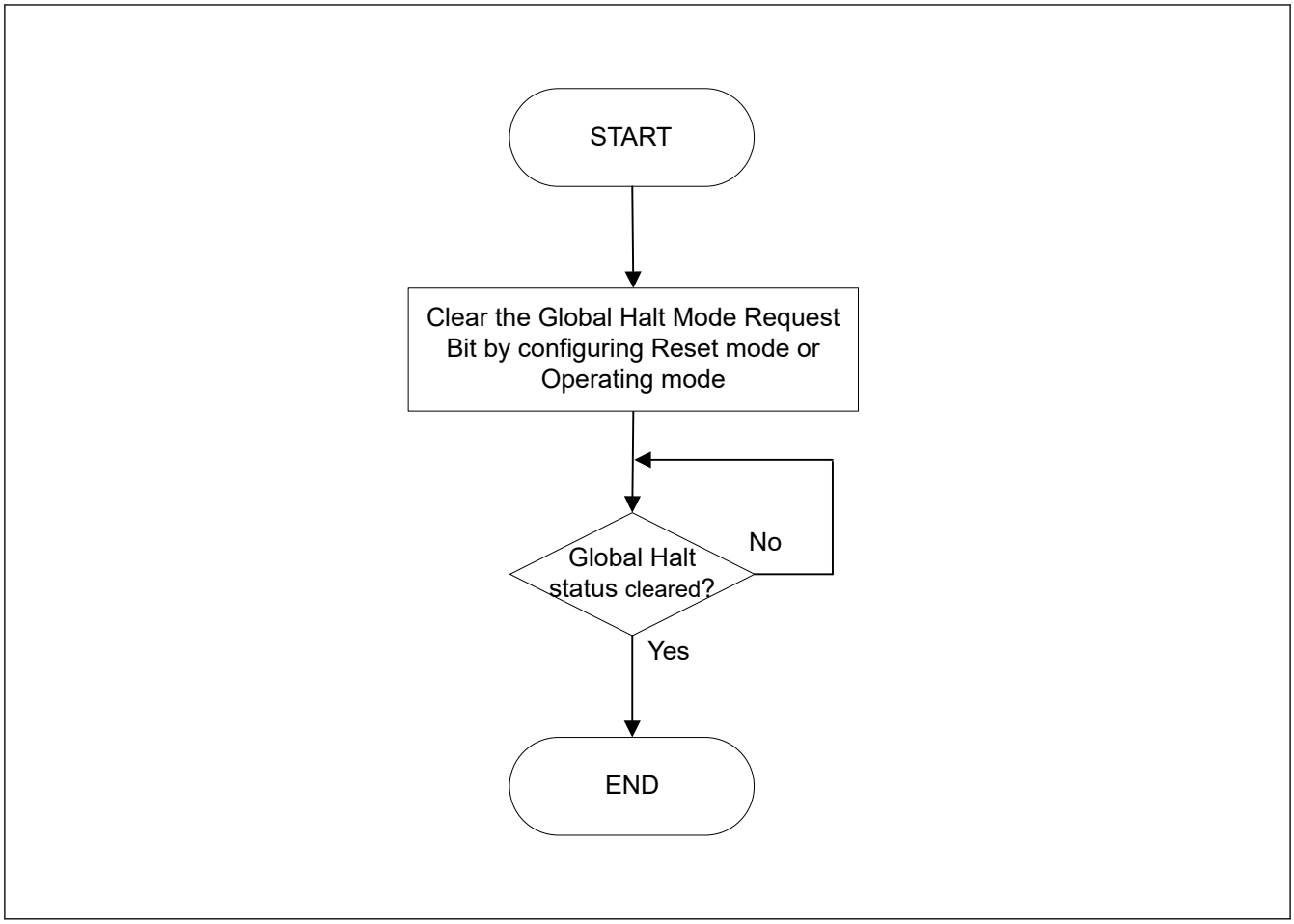


Figure 34.8 Procedure for exiting Global Halt mode

34.3.2.4 Global Operation Mode

The CANFD module enters this mode when the Global Mode Configuration bits are set to Global Operation mode.

The CANFD channel can only be set to Channel Operation mode and start CAN communication when CANFD is in Global Operation mode.

After setting the Global Mode Control bit CFDGCTR.GMDC to Global Operation mode, it is necessary to confirm that the Global Reset Mode Status bit CFDGSTS.GRSTSTS and the Global Halt Mode Status bit CFDGSTS.GHLTSTS in the Global Status Register have been cleared to indicate a successful transition to Global Operation mode before CFDGCTR.GMDC can be modified again.

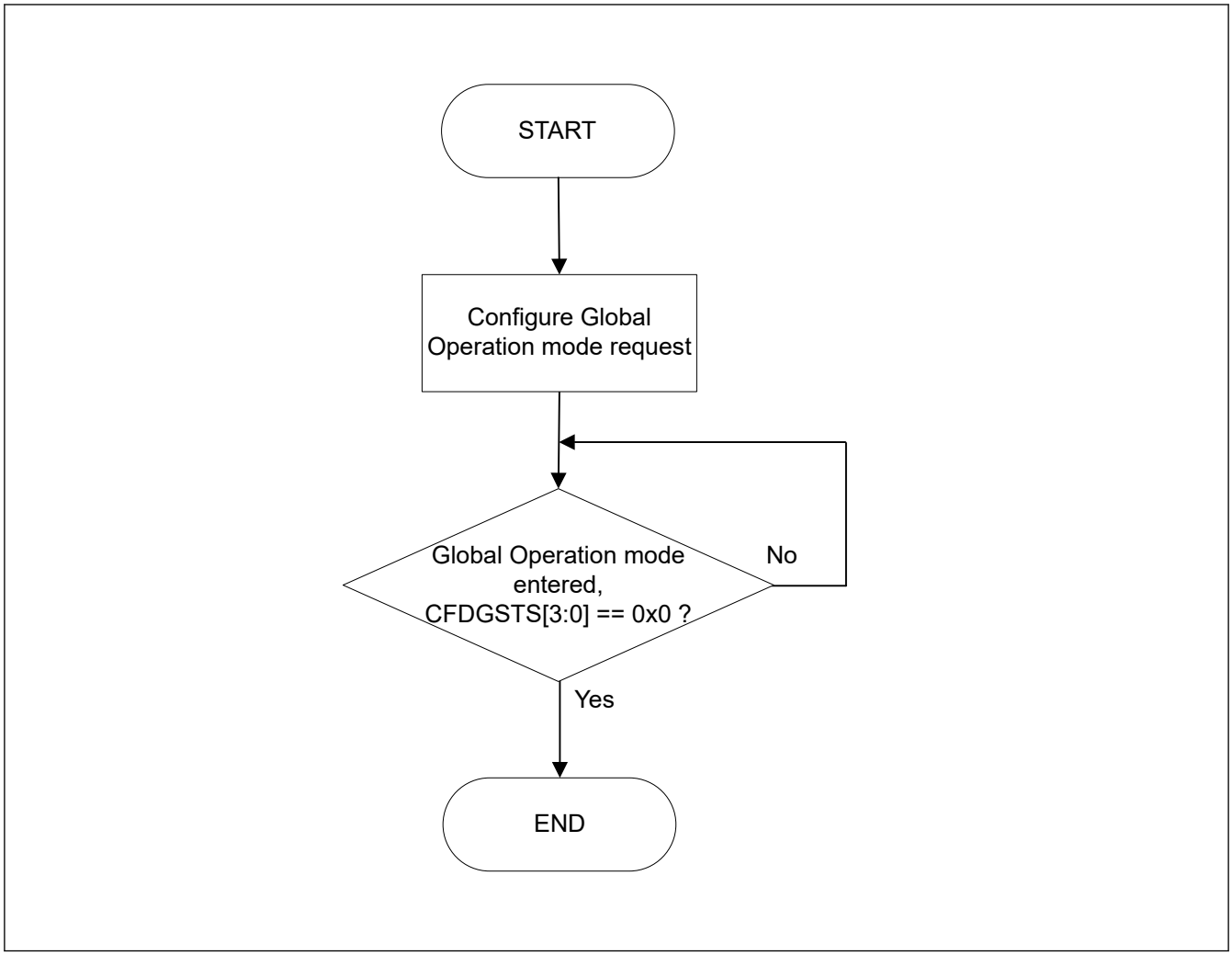


Figure 34.9 Procedure for entering Global Operation mode

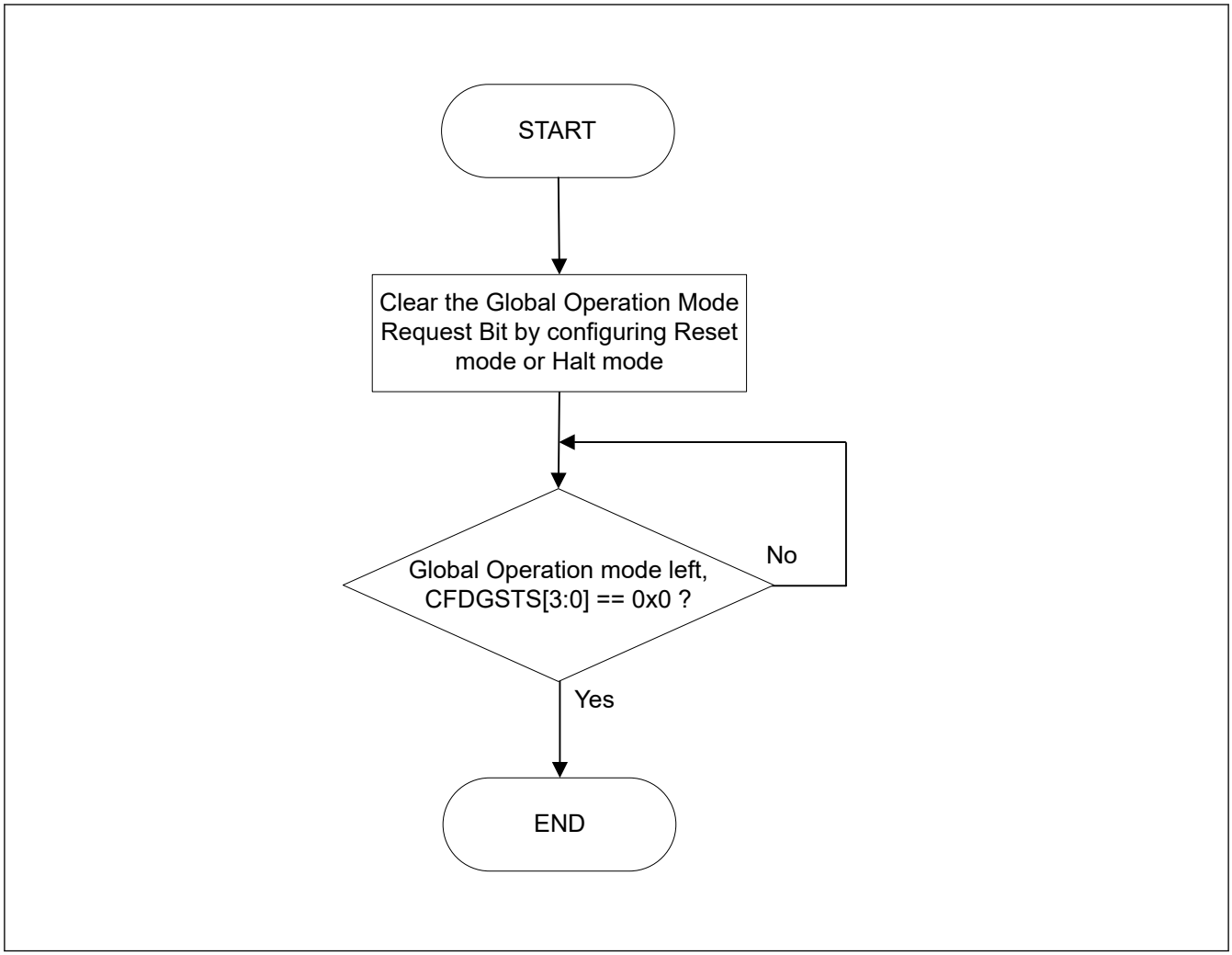


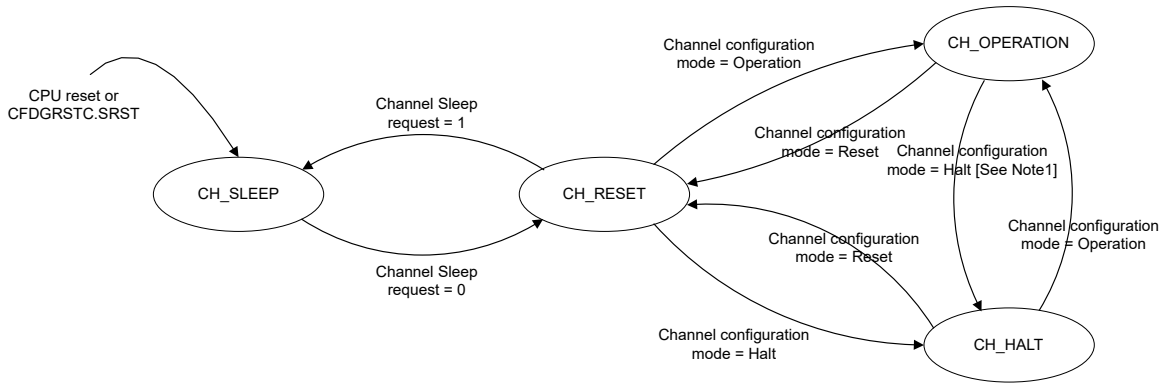
Figure 34.10 Procedure for exiting Global Operation mode

34.3.3 Channel Modes

A CAN channel can be in one of the following four channel modes:

- Reset
- Halt
- Operation
- Sleep.

Figure 34.11 shows the possible transitions between the channel modes.



Note 1. The transition timing depends on BOM setting in the Channel Configuration Register:
- when BOM = 01b, the state transition timing is immediate after entering bus-off state
- when BOM = 10b, the state transition timing is at the end of bus-off state
- when BOM = 11b, the state transition timing is matching with the setting of CHMDC to Halt mode.

Figure 34.11 Transition between CAN channel modes

34.3.3.1 CAN Channel Sleep Mode

After the release of a hardware reset or after setting and clearing the CFDGRSTC.SRST bit, a CAN channel of the CANFD module automatically enters Channel Sleep mode.

A CAN channel also enters Channel Sleep mode when the related Channel Sleep Mode Request bit is set while the CAN channel is in Channel Reset mode. Do not set this control bit in Channel Halt mode or Channel Operation mode.

Entering the CAN Channel Sleep mode instantly stops the clock supplied to the CAN channel unit and therefore reduces power consumption.

After setting the Channel Sleep Mode Request bit, it is necessary to confirm that the Channel Sleep mode status has been updated to indicate a successful transition to Channel Sleep mode before the Channel Sleep Mode Request bit can be cleared again.

During Channel Sleep mode, do not write to channel related registers. Read operation is still possible.

34.3.3.2 CAN Channel Reset Mode

A CANFD CAN channel enters this mode in the following ways:

- Channel Mode Control bit CFDC0CTR.CHMDC in the Channel Control Registers is configured for Channel Reset mode while the related CAN channel is in Channel Halt mode or Channel Operation mode
- Channel Sleep Mode Request bit is cleared while the related CAN channel is in Channel Sleep mode
- Global Mode Control bit CFDGCTR.GMDC is set to Global Reset mode and CAN channel is not in Channel Sleep mode or Channel Reset mode.

In Channel Reset mode, all CAN channel status and flag registers are initialized.

Additionally all channel related transmission control bits are cleared and the channel related TX Queue is disabled.

Configuration registers (except the Channel Test Mode registers) are not initialized in this mode and the CAN channel can be configured for communication.

See section 34.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Channel Reset mode is performed.

After setting the Channel Mode Control bit CFDC0CTR.CHMDC to Channel Reset mode, it is necessary to confirm that the Reset Mode Status bit CFDC0STS.CRSTSTS in the related Channel Status Registers has been updated to indicate a successful transition to Channel Reset mode before the related CFDC0CTR.CHMDC bit can be modified again.

See Table 34.14 for the behavior of transitioning to Channel Reset mode while CAN communication is ongoing.

34.3.3.3 CAN Channel Halt Mode

A CANFD CAN channel enters this mode in the following ways:

- Channel Mode Control bit CFDC0CTR.CHMDC in the Channel Control Registers is configured for Channel Halt mode while the related CAN channel is in Channel Reset mode or Channel Operation mode
- Global Mode Control bit CFDGCTR.GMDC is set to Global Halt mode and CAN channel is in Channel Operation mode.

In Channel Halt mode, all channel CAN communication is suspended but all status and flag registers remain unchanged during Channel Halt mode entry (except for the bus-off case where REC and TEC values are cleared for this channel).

In addition, the Channel Test Mode Configuration and Control registers are not initialized in this mode.

The Channel Halt mode should be used to configure channel test modes.

See section 34.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Channel Halt mode is performed.

After setting the Channel Mode Control bit CFDC0CTR.CHMDC to Channel Halt mode, it is necessary to confirm that the Halt Mode Status bit CFDC0STS.CHLTSTS in the related Channel Status Register has been updated to indicate a successful transition to Channel Halt mode before the related CFDC0CTR.CHMDC can be modified again.

See Table 34.14 for the transition behavior to Channel Halt mode while CAN communication is ongoing.

Table 34.14 Transition behavior in CAN Reset mode and Halt mode

Mode	State		
	Receiver	**Transmitter**	**Bus-Off**
CAN Channel Reset mode (CFDC0CTR.CHMDC = 01b)	The CAN channel enters Channel Reset mode without waiting for the completion of the ongoing reception.*1	The CAN channel enters Channel Reset mode without waiting for the completion of the ongoing transmission.*1	The CAN channel enters Channel Reset mode without waiting for the completion of the bus-off recovery.
CAN Channel Halt mode (CFDC0CTR.CHMDC = 10b)	CAN channel enters Channel Halt mode at the end of the ongoing reception or error.*2	CAN channel enters Channel Halt mode after completion of the ongoing transmission.	When CFDC0CTR.BOM is set to 00b, a Channel Halt mode request is accepted only after the completion of the full bus-off recovery sequence. When CFDC0CTR.BOM is set to 10b, the CAN channel transits automatically to Channel Halt mode after waiting for the completion of the bus-off recovery. When CFDC0CTR.BOM is set to 01b, the CAN channel transits automatically to Channel Halt mode without waiting for the completion of the bus-off recovery. When CFDC0CTR.BOM is set to 11b, the CAN channel enters Channel Halt mode as soon as Channel Halt mode is requested (without waiting for the completion of the bus-off recovery).

Note 1. If the entry to Channel Reset mode is required only at the end of an ongoing communication, then Channel Halt mode can be requested first to prevent interruption of CAN communication by direct transition to Channel Reset mode. After the CAN channel enters Channel Halt mode, the Channel Reset mode can be requested.

Note 2. If CAN communication is locked at dominant level after an error flag, software can detect this situation by monitoring the channel related BusLock flag and resolve lock condition by setting the CAN channel to Channel Reset mode.

34.3.3.4 CAN Channel Operation Mode

The Channel Operation mode is activated by setting the CFDC0CTR.CHMDC bits to 00b. If 11 consecutive recessive bits are detected after entering the CAN Operation mode, the CFDC0STS.COMSTS bit is set and the CAN channel:

- Enables the functions of the channel communication by allowing the channel to become an active node on the CAN network
- Releases the internal fault confinement logic including receive and transmit error counters

At this point, the CAN channel can start transmission and reception of CAN messages.

Within the CAN Channel Operation mode, the channel may be in four different sub-modes, depending on which type of communication functions are performed (see Figure 34.12):

- Channel idle: The CAN channel is neither receiving nor transmitting
- Channel receives: The channel is receiving a CAN message sent by another CAN node
- Channel transmits: The channel is transmitting a CAN message

  Note: The channel may receive its own message simultaneously when Self-test mode is enabled.
- Channel is in bus-off state: The CAN channel is cut-off from CAN bus communication.

After setting the Channel Mode Control bit CFDC0CTR.CHMDC to Channel Operation mode, it is necessary to confirm that the Channel Reset Mode Status bit CFDC0STS.CRSTSTS and the Channel Halt Mode Status bit CFDC0STS.CHLTSTS in the Channel Status Register have been updated to indicate a successful transition to Channel Operation mode before the related CFDC0CTR.CHMDC bit can be changed again.

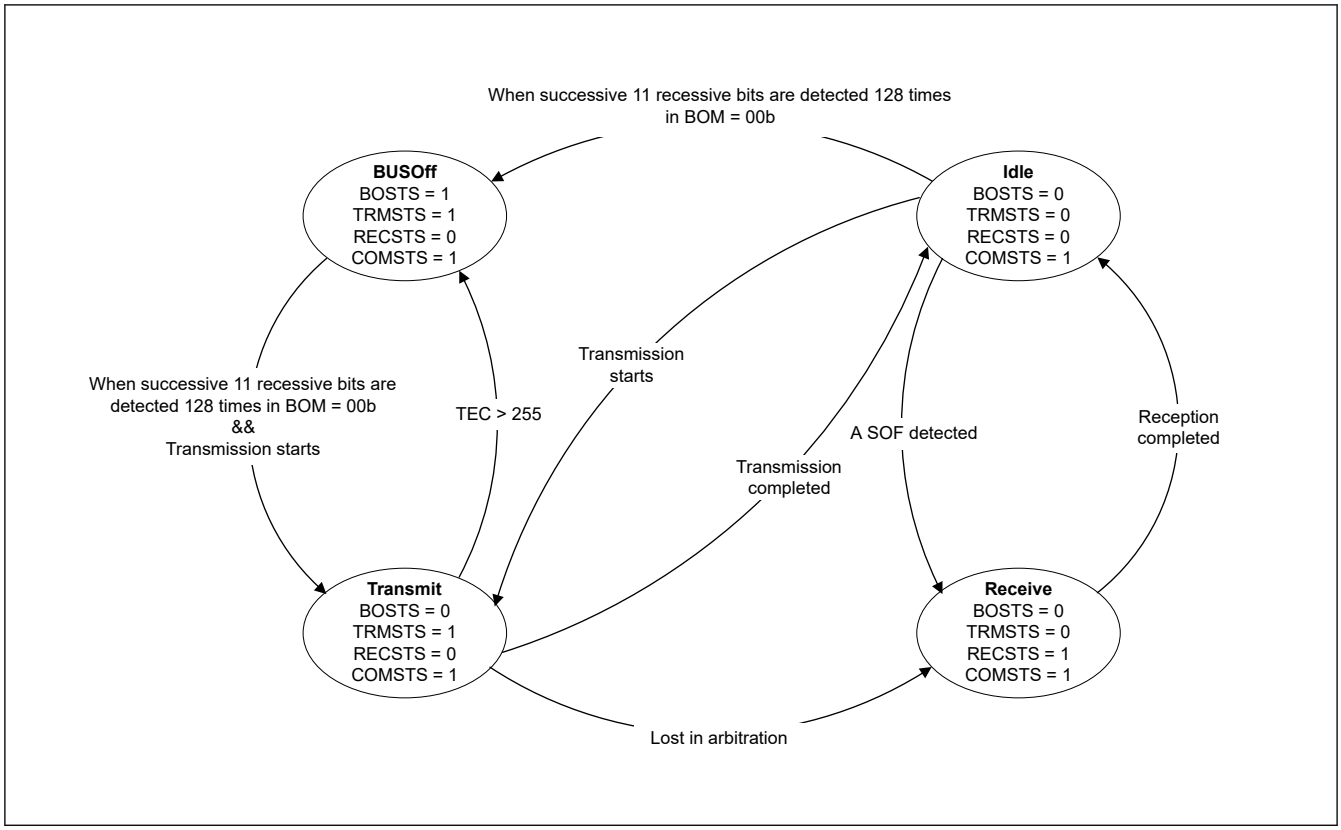


Figure 34.12 Sub-modes of CAN Channel Operation mode (only when BOM = 00b)

34.3.3.5 CAN Channel Bus-Off State

The CAN channel bus-off state is entered according to the fault confinement rules of the CAN specification. The following modes can be configured for returning to the CAN Channel Operation mode from the bus-off state:

- CFDC0CTR.BOM = 00b:
  Bus-Off recovery is compliant to ISO 11898-1, namely the CAN channel re-enters CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. TEC and REC counters are initialized to 0. The Bus-Off Recovery Flag CFDC0ERFL.BORF is set in this case.
- CFDC0CTR.BOM = 01b:

The CAN channel changes the value of the CFDC0CTR.CHMDC bits within the CAN Channel Control Register to 10b and switches immediately to Channel Halt mode automatically after entering bus-off state. TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDC0ERFL.BORF is not set in this case.

- CFDC0CTR.BOM = 10b:
  The CAN channel changes the value of the CFDC0CTR.CHMDC bits within the CAN Channel Control Register to 10b as soon as it reaches bus-off state and enters Channel Halt mode automatically after the CAN channel has completed the bus-off recovery sequence (after 11 consecutive recessive bits are detected 128 times). TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDC0ERFL.BORF is set in this case.
- CFDC0CTR.BOM = 11b:
  Bus-off recovery is initiated but CAN channel can immediately enter Channel Halt mode when still in bus-off state if a request is made to enter Channel Halt mode.
  TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDC0ERFL.BORF is not set.
  Without setting CFDC0CTR.CHMDC [1:0] = 10b and when 11 recessive bits is detected 128 times continuously, transition conditions become the same as CFDC0CTR.BOM = 00b.

Note: If the recovery from bus-off occurs normally in this mode (after waiting for 128 sequences of 11 consecutive recessive bits), and no halt request has been generated during this period, then the Bus-Off Recovery flag CFDC0ERFL.BORF is set.

When software writes to the CFDC0CTR.CHMDC bit at the same time as the CAN channel enters Halt mode (at the start of bus-off when CFDC0CTR.BOM = 01b, or at the end of bus-off when CFDC0CTR.BOM = 10b), the software request has the highest priority.

Note: In the above case, the automatic setting of the CFDC0CTR.CHMDC bit to Channel Halt mode request is performed when the CFDC0CTR.CHMDC bit value is previously 00b (Channel Operation mode).

Additionally, it is possible to force the CAN channel to recover from the bus-off state by setting CFDC0CTR.RTBO to 1. The error state changes from bus-off state to integrating state with a maximum delay of 1 CAN bit time, and the CAN communication becomes possible again after 11 consecutive recessive bits are detected. The Bus-Off Recovery Flag is not set in this case, and the TEC and REC counters are initialized to 0.

Before setting CFDC0CTR.RTBO to 1, all pending transmissions from the TX message buffers, TX Queues and/or Common FIFO in TX mode should be disabled.

The disable of the pending transmission message buffer, TX Queue or FIFO must be confirmed by the corresponding acknowledge flags.

For the TX message buffer, the acknowledge flags are the Transmission Result Flags (CFDTMSTSj.TMTRF). For the TX Queue, it is the TX Queue Empty flag (CFDTXQSTS.TXQEMP). For the FIFO, it is the FIFO Empty flag (CFDCFSTS.CFEMP).

The CFDC0CTR.RTBO bit should be used for bus-off recovery only when CFDC0CTR.BOM is set to 00b.

Setting this bit in any state other than bus-off has no effect and the bit is cleared immediately.

Table 34.15 shows the settings for the Bus-Off Entry flag CFDC0ERFL.BOEF and the Bus-Off Recovery flag CFDC0ERFL.BORF for the different configurations of CFDC0CTR.BOM.

Table 34.15 Behavior of Bus-off Entry and Recovery flags

BOM	BOEF bit set	BORF bit set
00b	Always (on entry to bus-off)	Always (on exit from bus-off)
00b CFDC0CTR.RTBO set to 1	Always (on entry to bus-off)	Only if normal bus-off recovery occurs before software sets CFDC0CTR.RTBO to 1'
01b	Always (on entry to bus-off)	Never
10b	Always (on entry to bus-off)	Always (on exit from bus-off)
11b	Always (on entry to bus-off)	Only if normal bus-off recovery occurs before software issues a Halt request

For an efficient software procedure, it is not necessary to wait for the bus-off recovery sequence to end.

It is possible to perform a transmission re-initialization during bus-off recovery. To do this, follow the recommended software flow in Figure 34.13.

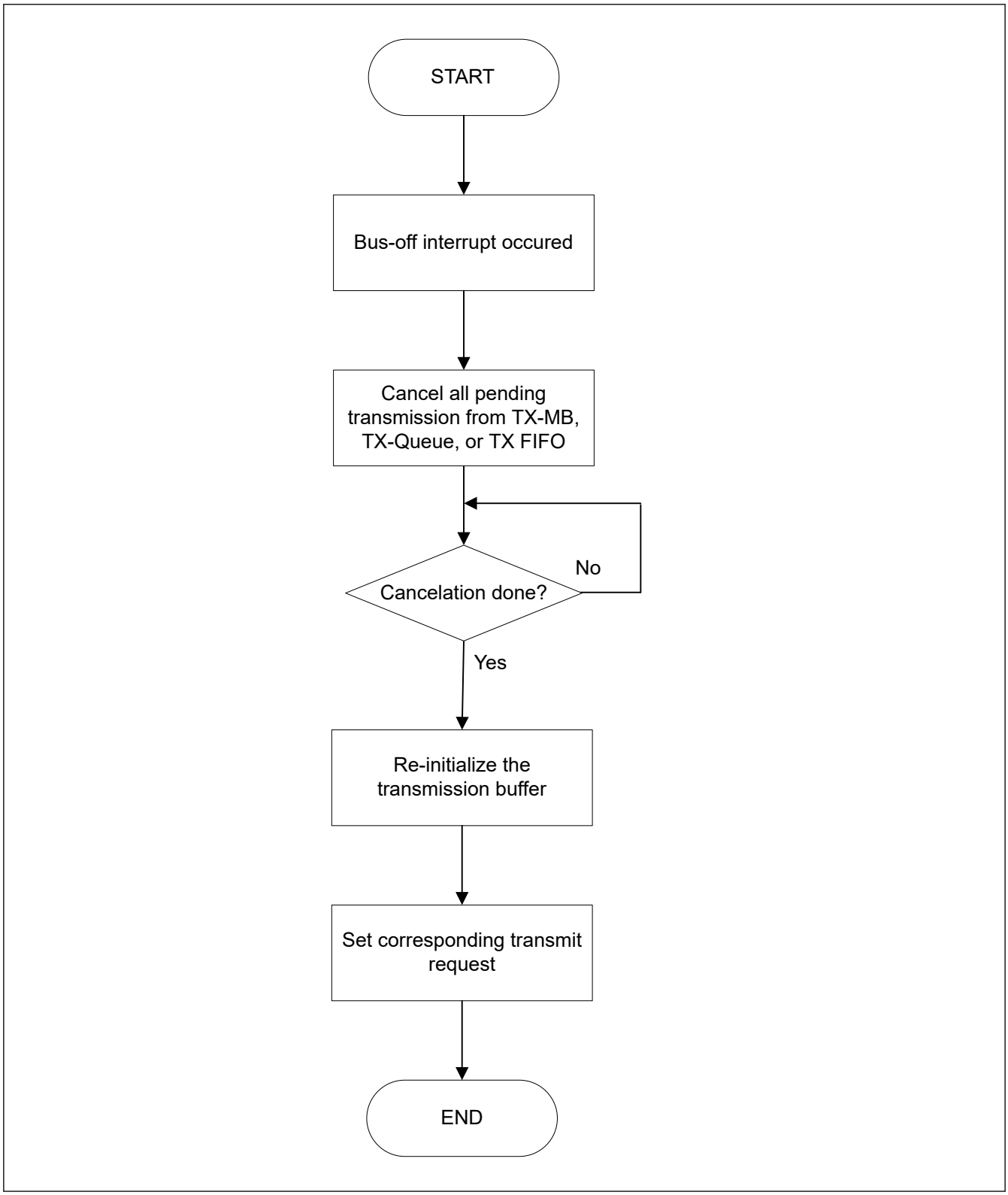


Figure 34.13 Transmission re-initialization during bus-off

34.3.4 Global Mode and Channel Mode Transition Interactions

The interaction between Global mode setting and Channel mode setting is as follows:

- Changing the Channel Mode Control bit CFDC0CTR.CHMDC in the Channel Control Registers does not affect the Global Mode Control bit CFDGCTR.GMDC.
- Changing the Global Mode Control bit CFDGCTR.GMDC affects the channel mode control as described in Table 34.16.

Table 34.16 Interaction between Global and Channel mode transition

Global mode change	Channel mode	Channel mode transition action
Sleep → Reset	Sleep	Channel remains in Sleep mode
Sleep → Halt	— (Global mode change not possible)	
Sleep → Operation	— (Global mode change not possible)	
Reset → Sleep	Sleep	Channel remains in Sleep mode
	Reset	Channel Sleep request bit is set automatically, channel enters Sleep Mode
Reset → Halt	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
Reset → Operation	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
Halt → Sleep	— (Global mode change not possible)	
Halt → Reset	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel mode control is set to Reset mode, channel enters Reset mode
Halt → Operation	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel remains in Halt mode
Operation → Sleep	— (Global mode change not possible)	
Operation → Reset	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel mode control is set to Reset mode, channel enters Reset mode
	Operation	Channel mode control is set to Reset mode, channel enters Reset mode
Operation → Halt	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel remains in Halt mode
	Operation	Channel mode control is set to Halt mode, channel enters Halt mode after communication finished

34.3.4.1 Timing of Global Mode Change

The transition time for the Global mode changes are shown in the following table.

Table 34.17 Maximum transition time for the global mode (1 of 2)

From	To	Maximum transition time
GL_SLEEP	GL_RESET	3 peripheral clock cycles*2
GL_RESET	GL_SLEEP	3 peripheral clock cycles
GL_RESET	GL_HALT	10 peripheral clock cycles
GL_RESET	GL_OPERATION	10 peripheral clock cycles

Table 34.17 Maximum transition time for the global mode (2 of 2)

From	To	Maximum transition time
GL_HALT	GL_RESET	2 CAN bit times
GL_HALT	GL_OPERATION	3 peripheral clock cycles
GL_OPERATION	GL_RESET	2 CAN bit times
GL_OPERATION	GL_HALT	3 CAN frames*1 *3

Note 1. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked RX lines or continued error conditions.
Note 2. Exit GL_SLEEP mode only when CFDGSTS.GRAMINIT is cleared.
Note 3. TQ, CAN frame and CAN bits are related to the individual channels. For the maximum transition time, the channel with the lowest baud rate must be used.

34.3.4.2 Timing of Channel Mode Change

The transition time for the Channel mode changes are shown in the following table.

Table 34.18 Maximum transition time for the channel mode

From	To	max. transition time
CH_SLEEP	CH_RESET	3 peripheral clock cycles
CH_RESET	CH_SLEEP	3 peripheral clock cycles
CH_RESET	CH_HALT	3 CAN bit times
CH_RESET	CH_OPERATION	4 CAN bit times
CH_HALT	CH_RESET	2 CAN bit times
CH_HALT	CH_OPERATION	4 CAN bit times*3
CH_OPERATION	CH_RESET	2 CAN bit times
CH_OPERATION	CH_HALT	2 CAN frames*1 *2

Note 1. The time specified for this transition does not include the case where channel enters bus-off state. For bus-off, the timing depends on the configuration of the CFDC0CTR.BOM[1:0] bits.
Note 2. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked RX lines or continued error conditions.
Note 3. In general, if the baudrate prescaler value CFDC0NCFG.NBRP is changed in CH_HALT mode, the transition time can deviate. The internal prescaler is a free running down counter that creates the TQ clock, and new BRP value is captured when the counter reaches the value 0.

34.4 Initialization

Before joining CAN communications, configure the following settings:

- Clock setting
- Bit timing setting (nominal and data rate)
- Baud Rate setting (nominal and data rate)
- CANFD setting
- Acceptance Filter setting (configuration of Global Acceptance Filter List)
- Reception, Transmission and GW-FIFO setting
- CAN Operation mode setting

34.4.1 Initialization of CAN Clock, Bit Timing and Baud Rate

34.4.1.1 Bit Timing Conditions

The following lines describe the composition of each segment and the restriction that apply to the segment setting.

1. Each segment setting
   SS = Fixed to 1 TQ

TSEG1 = See to (CFDC0NCFG) and (CFDC0DCFG)*1
TSEG2 = See to (CFDC0NCFG) and (CFDC0DCFG)*1
SJW = See to (CFDC0NCFG) and (CFDC0DCFG)*1
SS + TSEG1 + TSEG2 = 5 to 49 TQs for Data Bit Rate and 8 to 385 for Nominal Bit Rate

2. Restriction on TSEG1, TSEG2 and SJW
TSEG1(N) > TSEG2(N) ≥ SJW(N)
TSEG1(D) ≥ TSEG2(D) ≥ SJW(D)*1
When only classical frames are used, configure the bit fields TSEG1 and TSEG2 of CFDC0DCFG to valid values.

Note 1. This feature is not available in the classical CAN function.

Table 34.19 shows an example of how to set the bit timing to achieve the required Sample Point settings.

Table 34.19 Bit timing examples

1 bit	Set value (TQ)				Sample point*1(%)
	SS	TSEG1	TSEG2	SJW	
5TQ	1	2	2	1	60.00
8TQ	1	4	3	1	62.50
	1	5	2	1	75.00
10TQ	1	6	3	1	70.00
	1	7	2	1	80.00
12TQ	1	8	3	1	75.00
	1	9	2	1	83.33
15TQ	1	10	4	1	73.33
	1	11	3	1	80.00
16TQ	1	10	5	1	68.75
	1	11	4	1	75.00
20TQ	1	12	7	1	65.00
	1	13	6	1	70.00
24TQ	1	15	8	1	66.66
	1	16	7	1	70.83
50TQ	1	39	10	4	80.00

Note 1. Sample point (in case of 75%)

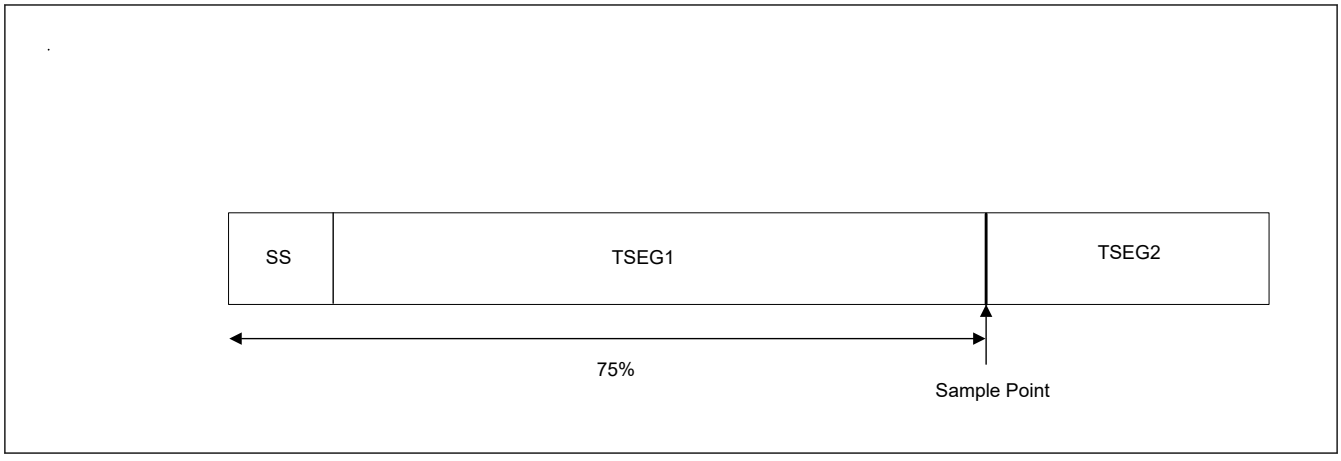


Figure 34.14 Sample point (in case of 75%)

34.4.1.2 CAN Bit Timing

In the CAN protocol, each bit in a communication frame is composed of three segments that can be configured individually for channel using the related CFDC0NCFG and CFDC0DCFG*1 registers.

Note 1. This register is not available in the classical CAN function.

Figure 34.15 shows the segment composition of a bit and the sample point in it.

Of these segments, the Time Segment 1 (TSEG1) and Time Segment 2 (TSEG2) are used to specify the position of the sample point, so that the timing at which each bit on the CAN bus is sampled can be altered by changing the values of these segments.

The minimum resolution for this timing is referred to as Time Quantum (TQ), which is determined by the clock frequency supplied to the CAN channel and the divide-by-N value of the baud rate prescaler (nominal and data rate).

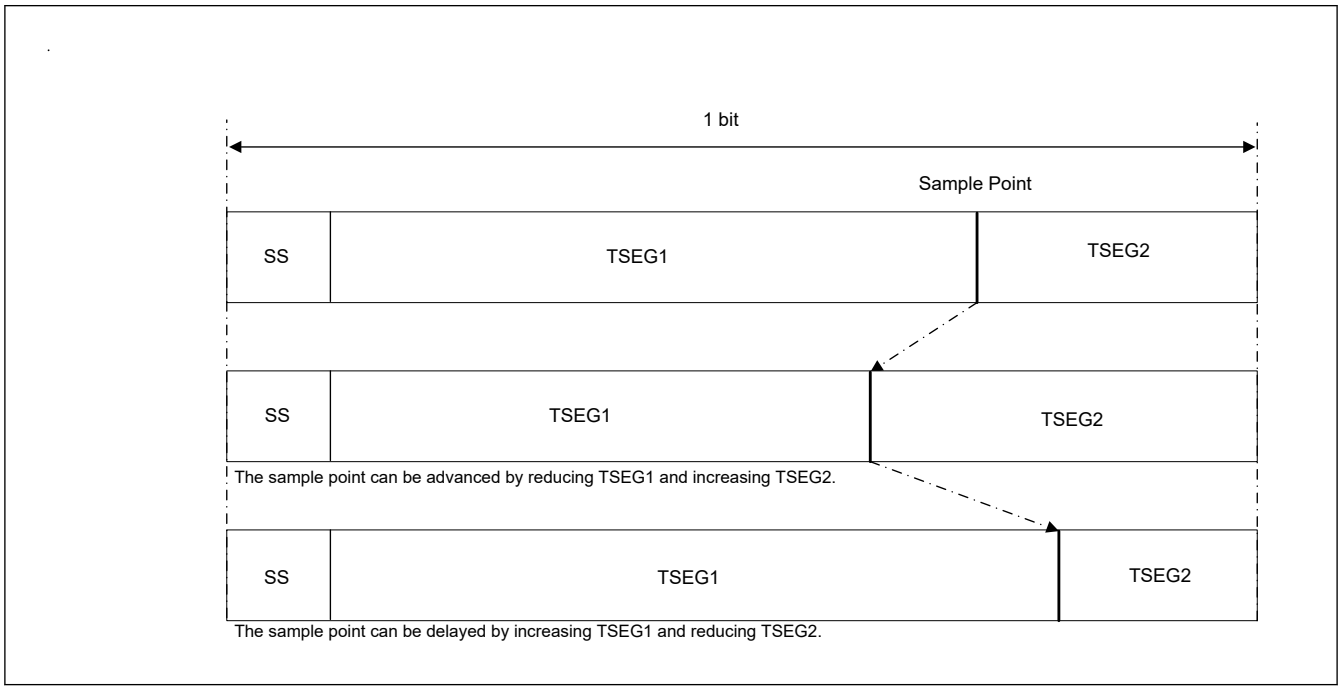


Figure 34.15 Segment composition of a bit and the sample point

1. SS: Synchronization Segment
   This segment is used to synchronize bits by monitoring a recessive-to-dominant edge during the interframe space. This comprises of intermission, suspend transmission, bus idle, during bus idle, and all nodes that can start transmission.
2. TSEG1: Time Segment 1
   This segment absorbs physical delays on the CAN network. A physical delay on the network is two times the total sum of a bus delay, input comparator delay, and output driver delay. It can be lengthened by SJW.
3. TSEG2: Time Segment 2
   This segment is used to correct a phase error by performing resynchronization. It can be shortened by SJW. While sending or receiving a message, communication frames between some nodes may get out of sync due to a drift in the oscillator frequency or a delay in the transmission path. This is referred to as a phase error.
4. SJW: Resynchronization Jump Width
   This is the maximum width by which bits that have become out of sync due to a phase error may be corrected.

Figure 34.15 shows only one symbolic sample point.

34.4.1.3 Baud Rate

Either the CANFD core clock (CANFDCLK) or the external oscillator clock (CANMCLK) can be selected globally as CAN communication clock.

The transfer speed is determined by the DLL clock, the divide-by-N value of the baud rate prescaler, and the number of TQs in one bit.

$$\text{baudrate} = \frac{\text{DLL\_Clock}}{(\text{number\_of\_time\_quanta\_per\_bit}) \times (\text{BRP} + 1)}$$

Figure 34.16 shows a block diagram of the circuit that generates the CAN channel communication clock and Table 34.20 shows a baud rate examples.

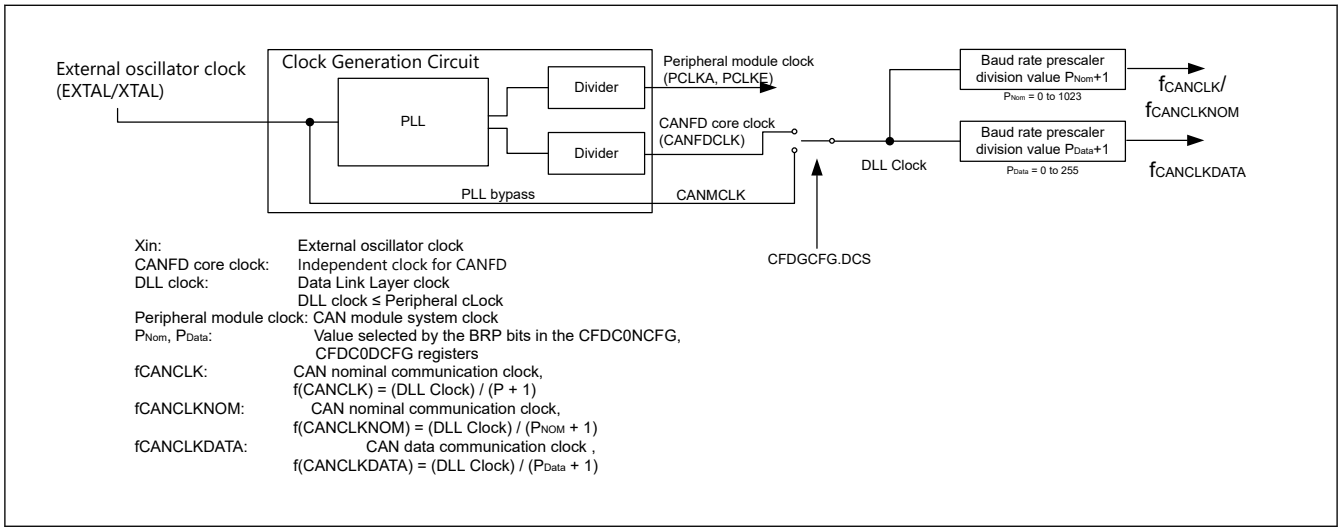


Figure 34.16 Block diagram of the circuit that generates the CAN channel communication clock

Table 34.20 Nominal baud rate calculation formula and example CAN communication configurations

Baud rate calculation formula	(DLL clock) (baud rate prescaler divide-by-N value*1) × (number of TQs in one bit)								
	80 MHz	40 MHz	32 MHz	30 MHz	24 MHz	20 MHz	16 MHz	10 MHz	8 MHz*2
1 Mbps	8TQ (10) 20TQ (4)	8TQ (5) 20TQ (2)	8TQ (4) 16TQ (2)	10TQ (3) 15TQ (2)	8TQ (3) 12TQ (2) 24TQ (1)	10TQ (2) 20TQ (1)	8TQ (2) 16TQ (1)	10TQ (1)	8TQ (1)
500 Kbps	8TQ (20) 20TQ (8)	8TQ (10) 20TQ (4)	8TQ (8) 16TQ (4)	10TQ (6) 15TQ (4) 20TQ (3)	8TQ (6) 12TQ (4) 24TQ (2)	10TQ (4) 20TQ (2)	8TQ (4) 16TQ (2)	10TQ (2) 20TQ (1)	8TQ (2) 16TQ (1)
250 Kbps	8TQ (40) 20TQ (16)	8TQ (20) 20TQ (8)	8TQ (16) 16TQ (8)	10TQ (12) 15TQ (8) 20TQ (6)	8TQ (12) 12TQ (8) 24TQ (4)	10TQ (8) 20TQ (4)	8TQ (8) 16TQ (4)	10TQ (4) 20TQ (2)	8TQ (4) 16TQ (2)
125 Kbps	8TQ (80) 20TQ (32)	8TQ (40) 20TQ (16)	8TQ (32) 16TQ (16)	10TQ (24) 15TQ (16) 20TQ (12)	8TQ (24) 12TQ (16) 24TQ (8)	10TQ (16) 20TQ (8)	8TQ (16) 16TQ (8)	10TQ (8) 20TQ (4)	8TQ (8) 16TQ (4)
83.3 Kbps	8TQ (120) 12TQ (80) 16TQ (60) 24TQ (40)	8TQ (60) 12TQ (40) 16TQ (30) 24TQ (20)	8TQ (48) 12TQ (32) 16TQ (24) 24TQ (16)	8TQ (45) 10TQ (36) 12TQ (30) 15TQ (24) 20TQ (18) 24TQ (15)	8TQ (36) 12TQ (24) 16TQ (18) 24TQ (12)	8TQ (30) 10TQ (24) 12TQ (20) 15TQ (16) 16TQ (15) 20TQ (12) 24TQ (10)	8TQ (24) 12TQ (16) 16TQ (12) 24TQ (8)	8TQ (15) 10TQ (12) 12TQ (10) 15TQ (8) 20TQ (6) 24TQ (5)	8TQ (12)
33.3 Kbps	8TQ (300) 12TQ (200) 16TQ (150) 20TQ (120) 24TQ (100)	8TQ (150) 12TQ (100) 16TQ (75) 20TQ (60) 24TQ (50)	8TQ (120) 10TQ (96) 12TQ (80) 15TQ (64) 16TQ (60) 20TQ (48) 24TQ (40)	10TQ (90) 12TQ (75) 15TQ (60) 20TQ (45)	8TQ (90) 10TQ (72) 12TQ (60) 15TQ (48) 16TQ (45) 20TQ (36) 24TQ (30)	8TQ (75) 10TQ (60) 12TQ (50) 15TQ (40) 20TQ (30) 24TQ (25)	8TQ (60) 10TQ (48) 12TQ (40) 15TQ (32) 16TQ (30) 20TQ (24) 24TQ (20)	10TQ (30) 12TQ (25) 15TQ (20) 20TQ (15)	8TQ (30)

Note: Shown in ( ) are the baud rate prescaler divide-by-N value.
Note 1. Baud rate prescaler divide-by-N value = P + 1 (P = 0 - 1023) P: value selected by the BRP bits in the Channel Configuration Registers.
Note 2. Minimum frequency to achieve maximum nominal baud rate of 1 Mbps.

Table 34.21 Baud rate calculation example for nominal and data bit rate CAN communication configurations

Baud rate calculation formula	(DLL clock) (baud rate prescaler divide-by-N value*1) × (number of TQs in one bit)		
	80 MHz	40 MHz	20 MHz
Nominal 1 Mbps Data 8 Mbps	80TQ (1)	40TQ (1)	20TQ (1)
	10TQ (1)	5TQ (1)	Not possible
Nominal 1 Mbps Data 5 Mbps	80TQ (1)	40TQ (1)	20TQ (1)
	16TQ (1)	8TQ (1)	Not possible
Nominal 500 Kbps Data 2 Mbps	160TQ (1)	80TQ (1)	40TQ (1)
	40TQ (1)	20TQ (1)	10TQ (1)

Note: Shown in ( ) are the baud rate prescaler divide-by-N values and this table is not available in the classical CAN function.
Note 1. Baud rate prescaler divide-by-N value = P + 1 (P = 0 - 1023) P: value selected by the BRP bits in the Channel Configuration Registers.

For optimum clock tolerance in networks using the FD frame format, the length of the time quantum should be the same in nominal bit time and in data bit time. This means CFDC0NCFG.NBRP = CFDC0DCFG.DBRP.

Additionally, if transceiver delay compensation is used, do not program the CFDC0DCFG.DBRP bit to be greater than 1, as 1 means divide by 2.

34.4.1.4 Setting of CAN Clock, Bit Timing and Baud Rate

Figure 34.17 shows the procedure for setting the CAN clock and the baud rate for each channel.

These settings should be performed during Channel Reset mode (Configuration mode) for the CAN channels.

Before going to channel communication state, the baud rate must be configured, otherwise the mode does not switch correctly.

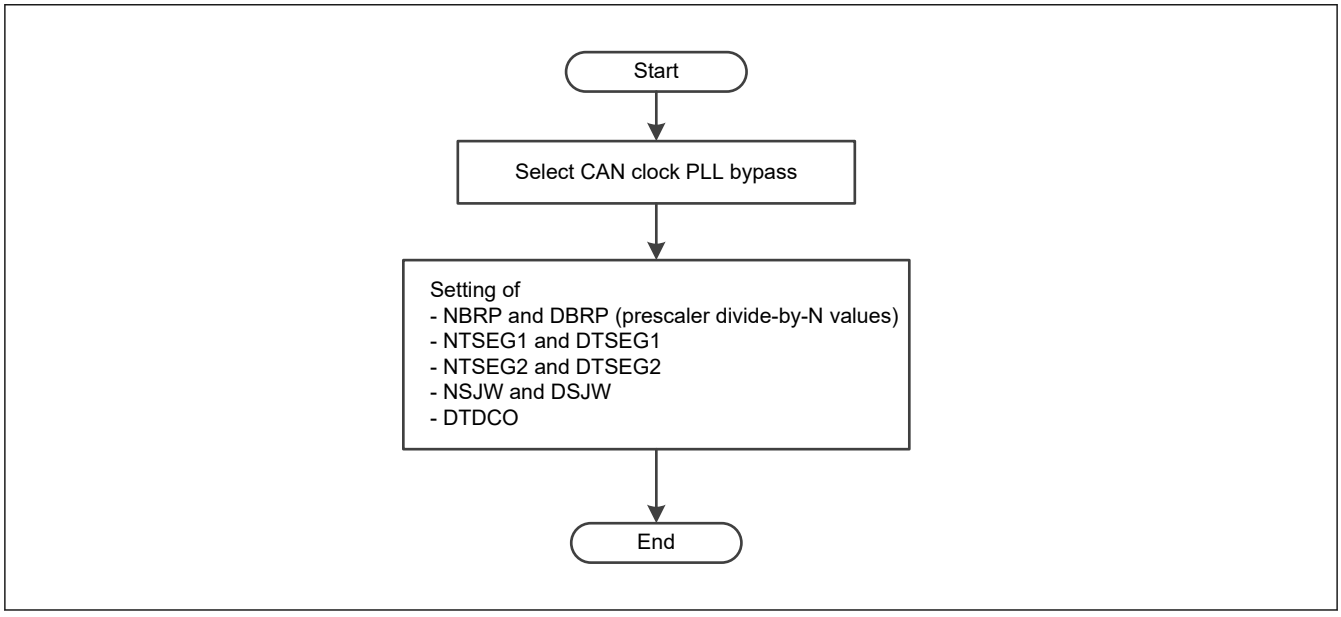


Figure 34.17 Procedure for setting the CAN bit timing and baud rate

34.4.1.5 Transmitter Delay Compensation

This chapter is not valid for classical CAN.

When a high baud rate is used such as 5 to 8 Mbps for the data phase, the transmitter delay can become greater than TSEG1. In this case, the transmitter always detects a bit-error in the data phase of the CANFD frame. The TDC compensates for the inability of the transmitter to receive its own transmitted bit at the sample point of that bit.

There is another symbolic sample point known as the Secondary Sample Point (SSP) that is used only during the data phase of CANFD frames. This is derived from the Transceiver Delay Compensation Result bit (CFDC0FDSTS.TDCR) as shown in Figure 34.18.

The resolution of the configuration, measured and offset values is based on the CAN channel DLL clock.

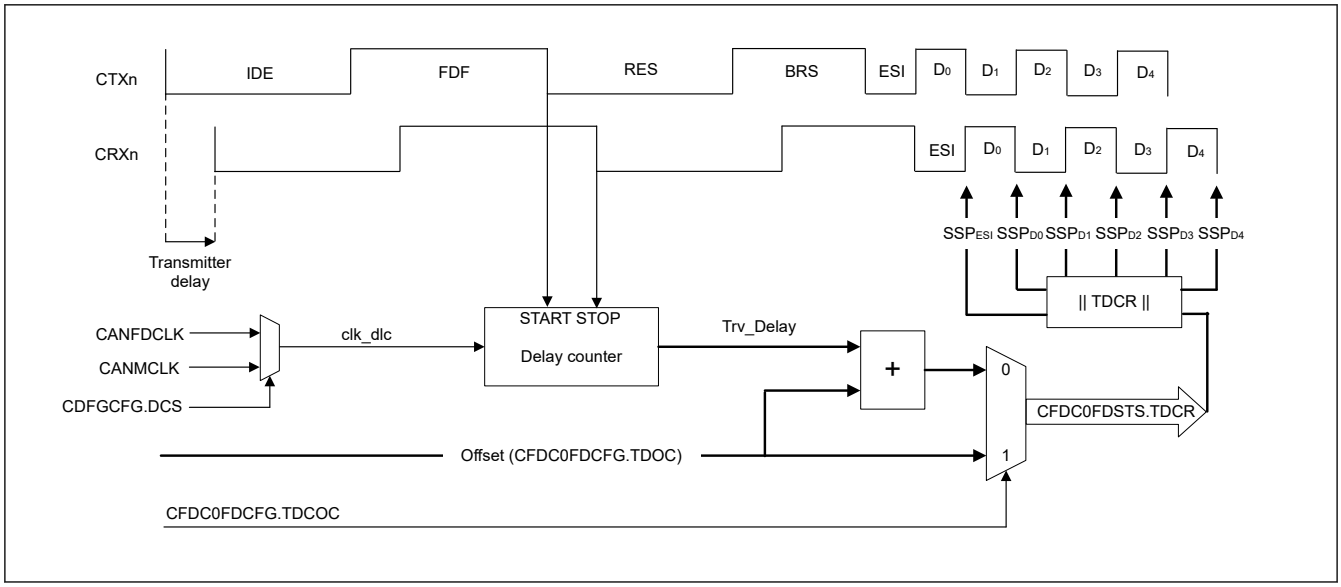


Figure 34.18 Transmitter delay compensation

The measured Trv_Delay is based on the number of clk_dlc clock cycles.The delay is counted up by one for each started clock until the dominant value is seen on CRXn. Figure 34.19 shows the measured result. Trv_Delay counted to maximum 127 with a clk_dlc clock.

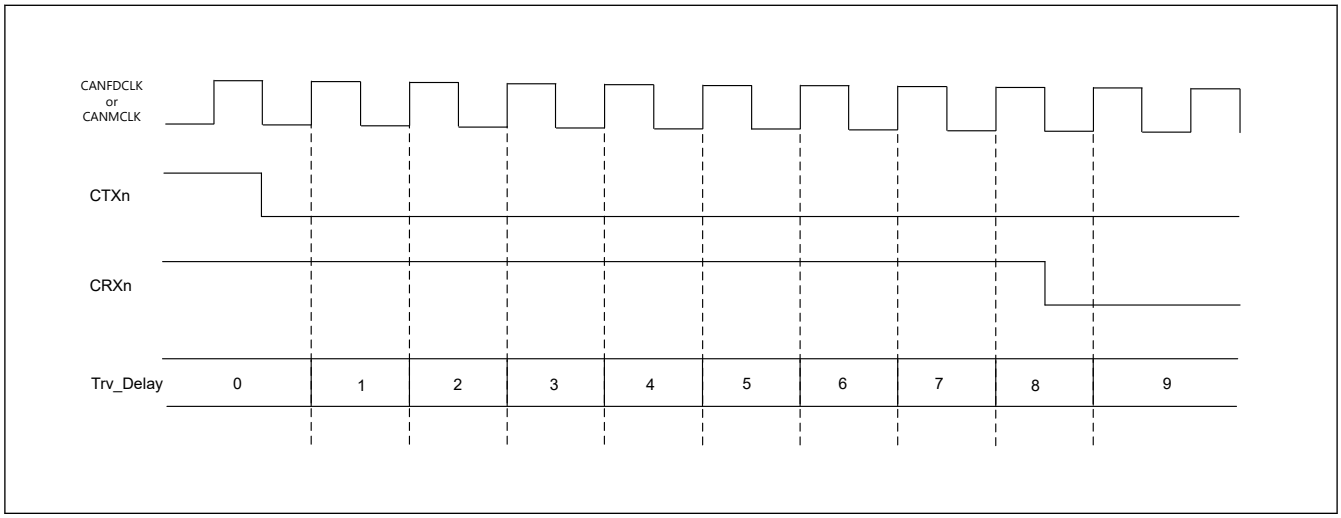


Figure 34.19 Trv_Delay measurement example

The SSP is calculated by taking the result from CFDC0FDSTS.TDCR and rounding the value down to the nearest integer number of data time quanta.

Figure 34.20 shows the positioning of the secondary sample point. When CFDC0FDCFG.TDCOC is equal to 0, the SSP is equal to the Trv_Delay (measured delay) + CFDC0FDCFG.TDCO, rounded down to the nearest integer number of time quanta. Usually, the TDCO value should have the size of (SyncSegmentdata + TSEG1data) to position the SSP to a theoretical location of the sample point.

If the CFDC0FDCFG.TDCOC is equal to 1, the SSP is defined by CFDC0FDCFG.TDCO. If CFDC0DCFG.DBRP is greater than 0, the value is also rounded down to the nearest integer number of time quanta.

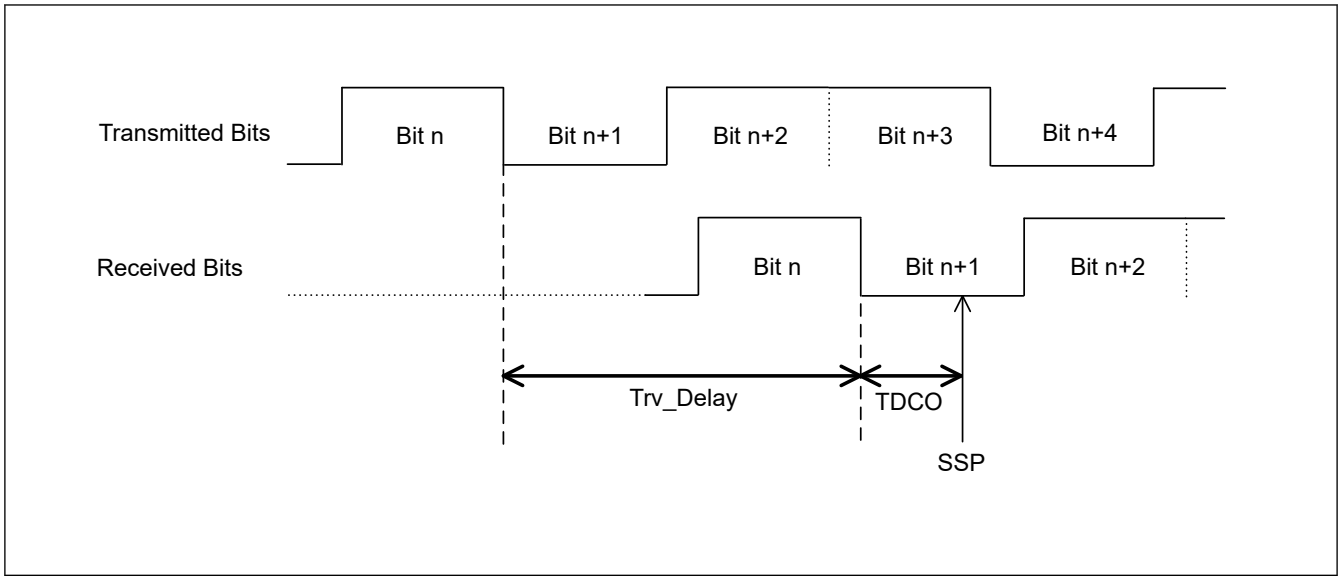


Figure 34.20 Position of the secondary sample point

The maximum delay (Trv_Delay + TDCO) which can be compensated by the CANFD module is (6 data bits - 2clk_dlc).

The ISO 11898-1 allows you to set different values for BRP_data and BRP_nom.

If different values are used for CFDC0NCFG.NBRP and CFDC0DCFG.DBRP, then two CAN nodes may be out of synchronization at the point when the bit rate changes from nominal bit rate to data bit rate after sample point of the BRS bit. This condition is shown in Figure 34.21.

The length of the time quantum should be the same in the nominal bit time and in the data bit time. This means CFDC0NCFG.NBRP = CFDC0DCFG.DBRP.

Different bit rates can be achieved by selecting different configuration values for the Time Segments. The nominal bit rate can be configured from 8 to 385 TQs and the data bit rate from 5 to 49 TQs.

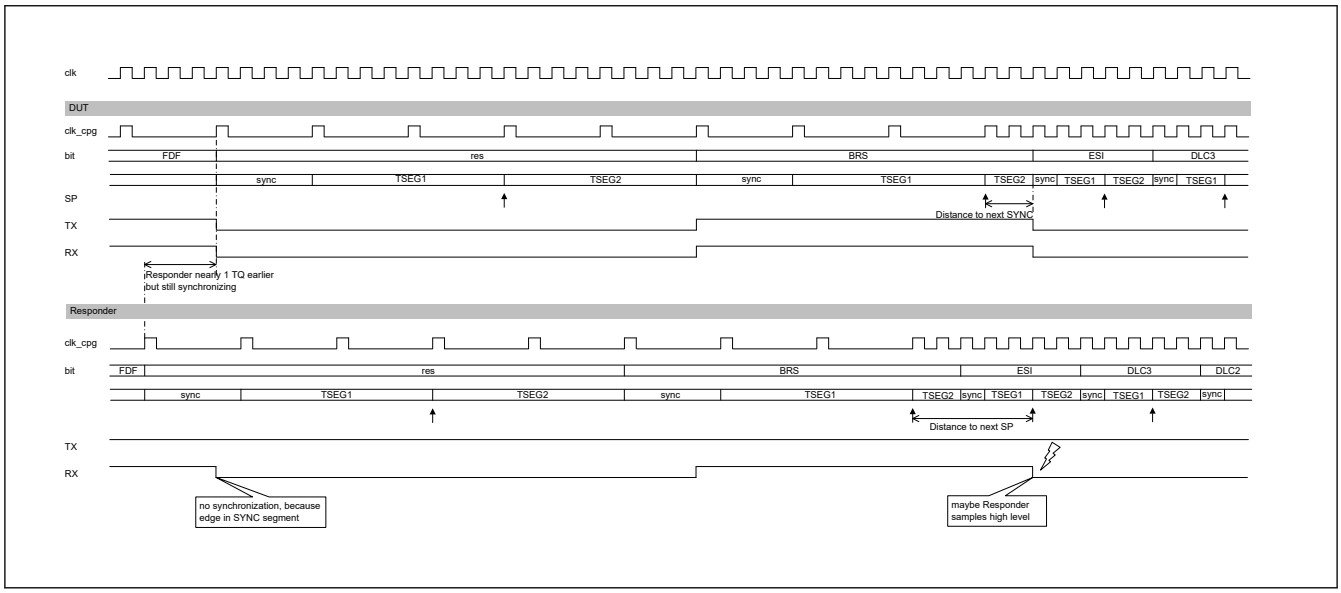


Figure 34.21 Loss of synchronization between two CAN nodes

The transmitter delay compensation measurement result is updated at the falling edge from FDF bit to RES bit when configured accordingly (CFDC0FDCFG.TDCE = 1, CFDC0FDCFG.TDCOC = 0).

Figure 34.22 shows the read flow to get the measured transmitter delay compensation result.

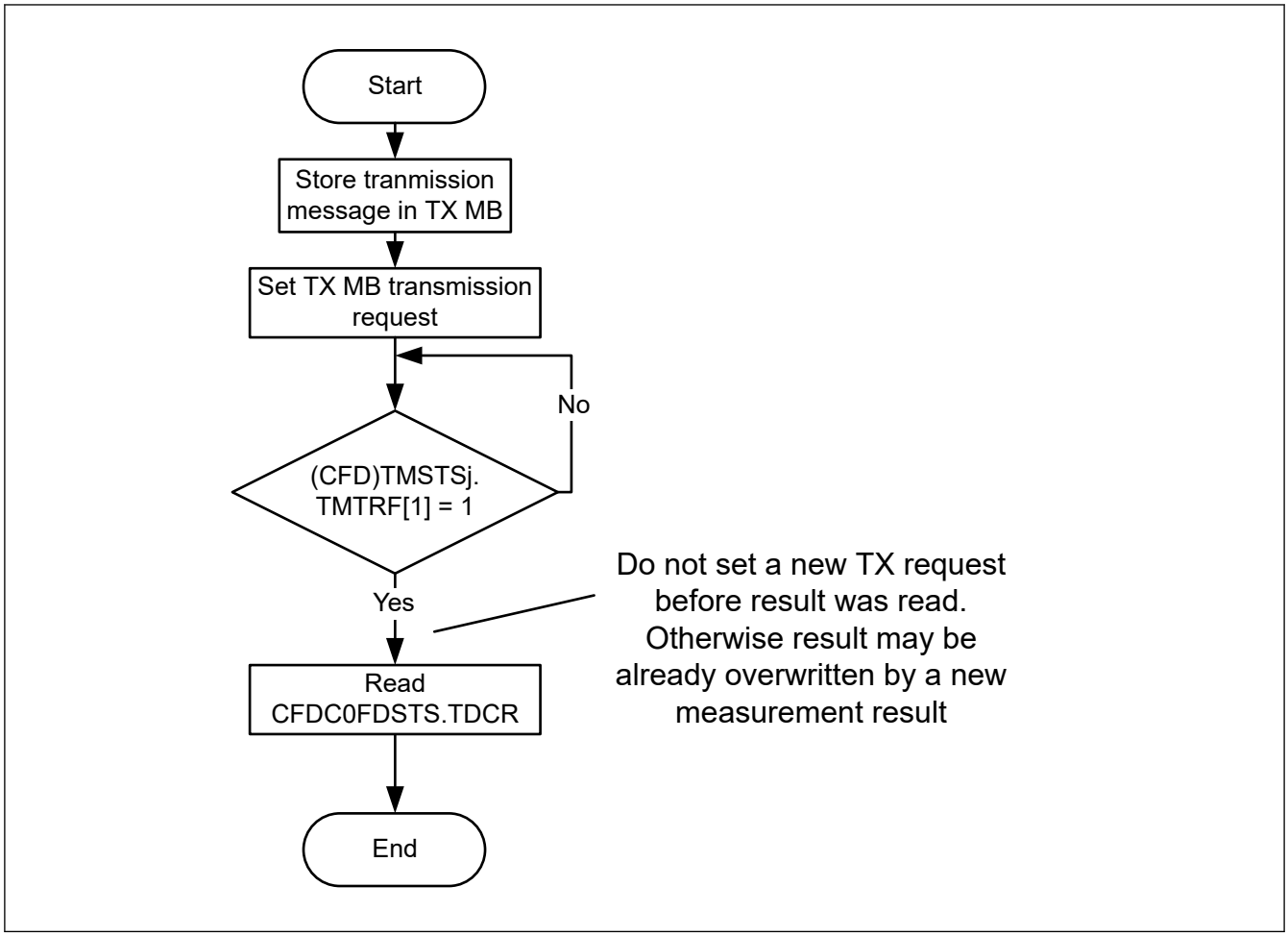


Figure 34.22 TDC result read flow

34.4.2 CAN Module Configuration after Hardware Reset

After a hardware reset (power on reset) or after setting and clearing a CFDGRSTC.SRST bit, the CANFD module enters Global Sleep mode automatically.

To enable configuration of the CANFD module, you must exit Sleep mode by clearing the Global Sleep Request bit CFDGCTR.GSLPR to 0.

After a hardware reset, the module starts RAM initialization, the CFDGSTS.GRAMINIT bit in the Global Status Register is set automatically to indicate that the CANFD logic is initializing the RAM.

After RAM initialization is complete, this bit is cleared automatically.

RAM initialization is necessary to avoid setting of false ECC error flag after HW reset the random data presented in the RAM.

Do not access registers of CANFD in either read or write until RAM initialization is complete and the CFDGSTS.GRAMINIT bit is cleared.

Before going to communication mode, the Global Acceptance Filter List and message FIFO buffers must be configured. In addition, CAN channel must be configured such as CAN bit timing. For this configuration, CAN channel must be released from Channel Sleep mode and must be configured for communication in Channel Reset mode (Configuration mode).

Figure 34.23 shows the configuration procedure. For details about each step, see section 34.5. Acceptance Filtering Function using Global Acceptance Filter List (AFL), section 34.6. FIFO Buffers and Normal Message Buffer Configuration, section 34.7. Interrupts and DMA and section 34.4.1.3. Baud Rate.

The CANFD module does not perform the RAM initialization sequence after executing a software reset by setting CFDGRSTC.SRST.

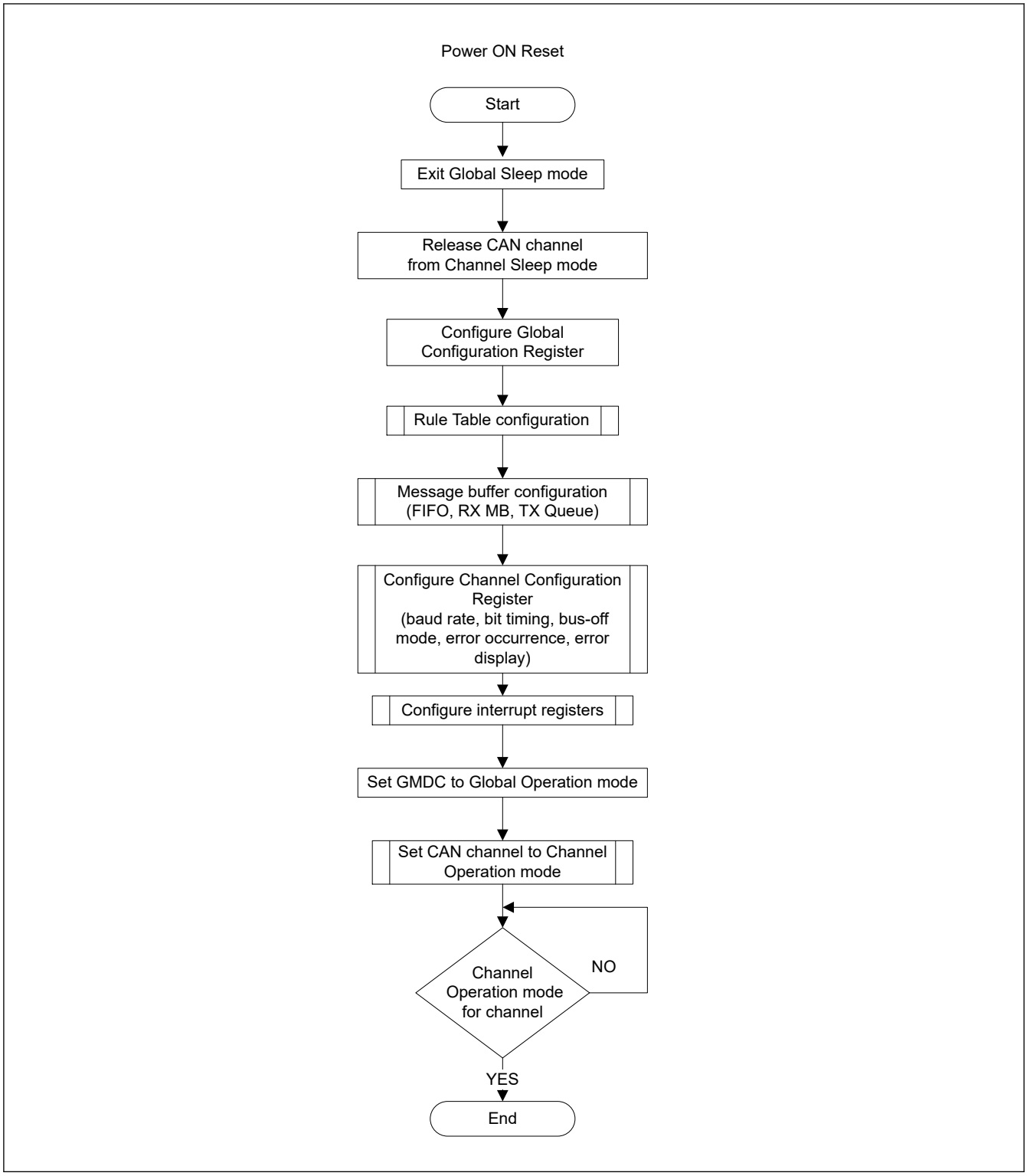


Figure 34.23 Configuration procedure after a hardware reset

34.5 Acceptance Filtering Function using Global Acceptance Filter List (AFL)

34.5.1 Overview

The CANFD module can handle message acceptance filtering with a global Acceptance Filter List (called AFL). Each element of the AFL defines a filter rule for messages received on a specific channel.

The following actions are performed based on the AFL entries:

- Acceptance filtering based on received CAN Identifier and masking
- DLC filtering based on received DLC value
- Message data payload according to the CFDGCFG.CMPOC bit*1
- Storage of accepted messages in the message buffer objects defined in the related AFL entry
- Attaching a 16-bit pointer to the stored messages defined in the related AFL entry, for example to support AUTOSAR applications
- Attaching a 2-bit information label to the stored messages defined in the related AFL entry

Note 1. This feature is not available in the classical CAN function.

The CANFD module allows a maximum of 16 AFL entries.

During acceptance filtering process, each AFL entry in a channel is checked against the received message by the acceptance filter unit. The check starts from the lowest AFL entry number for this channel.

AFL search stops when a match of the received identifier with a configured identifier/mask combination occurs or when the received identifier has been compared against all AFL entries defined for the related channel. If no match occurs, then the received message is rejected. No notification is given to the application in this case.

Additionally, an automatic DLC filtering is performed for each accepted message if DLC check is globally enabled. If the DLC value of the received message is equal to or higher than the configured DLC value in the matching AFL entry, the DLC check is passed.

If DLC replacement (CFDGCFG.DRE bit) is enabled, DLC value configured in the matching AFL entry is greater than 0x0 and DLC check passes, then the configured value of DLC in the matching AFL entry is stored in the destination RXMB or FIFO Buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional data bytes received on the CAN Bus are not stored in the destination RXMB or FIFO Buffer. These additional data bytes are stored as 0x00 in the destination RXMB or FIFO Buffer.

If DLC replacement is enabled and DLC value of matching AFL entry is 0x0, then the received value of DLC is stored in the destination RX MB or FIFO Buffer.

If DLC replacement (the CFDGCFG.DRE bit) is disabled and DLC check passes, then the received value of DLC on the CAN bus is stored in the destination RXMB or FIFO buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional data bytes received from the CAN bus are also stored in the destination RXMB or FIFO buffer.

If DLC value of the received message is less than the configured DLC value in the matching AFL entry, then DLC check fails. In this case, the received message is rejected and is not stored in any RXMB or FIFO buffer.

Additionally, DLC check failure is flagged by the DLC Error Flag in the Global Error Flag Register. If configured, an error interrupt is also generated. The DLC replacement configuration has no impact if the DLC check fails.

If a message has passed both acceptance filtering and DLC filtering, it is stored in a single reception message buffer and/or in FIFO buffers configured for reception function.

This message storage target information is also defined in the same AFL entry. Do not set a target at the AFL entry which is not configured.

Each accepted received message can be stored into a maximum of 2 different target destinations (single reception message buffer and/or FIFO buffers).

The programming of more than 2 target destinations is not allowed. If more destinations are programmed, then the internal timing might lead to a race condition that prevents the storage of received messages in the message RAM.. Correct configuration of the numbers of target destination is the responsibility of the application.

Additional protection mechanism is made for the case when a received message contains more data payload Bytes than possible to store in the target destination (CFDRMNB.RMPLS, CFDRFCCa.RFPLS or CFDCFCC.CFPLS).

If CFDGCFG.CMPOC = 0, the message is completely rejected and is stored in the target destination. When CFDGCFG.CMPOC = 0 and RX or Common FIFO full including the received message contains more data payload bytes than possible to store in the target destination (CFDRMNB.RMPLS, CFDRFCCa.RFPLS or CFDCFCC.CFPLS), the corresponding CFDFMSTS.RFxMLT or CFDFMSTS.CFxMLT bit is not set to 1, respectively.

When CFDGCFG.CMPOC = 1, the received data bytes greater than CFDRMNB.RMPLS is rejected. When CFDGCFG.CMPOC = 1 and RX or Common FIFO full including the received message contains more data payload bytes than possible to store in the target destination (CFDRMNB.RMPLS, CFDRFCCa.RFPLS or CFDCFCC.CFPLS), the corresponding CFDFMSTS.RFxMLT or CFDFMSTS.CFxMLT bit is set to 1, respectively.

Depending on the CFDGCFG.DRE bit, the original received DLC or the DLC value configured at the AFL entry is stored.

Regardless of the CFDGCFG.CMPOC bit setting, CFDGERFL.CMPOF is set to 1 if a payload overflow condition is detected.

The DLC filtering is performed before the payload overflow function. So for one reception frame, only one flag can be set at the same time with CFDGERFL.DEF or CFDGERFL.CMPOF*1.

Note 1. This bit is not available in the classical CAN function.

34.5.2 Allocation of AFL Entries

The number of AFL entries per channel can be configured using the dedicated field in the related Global Acceptance Filter Configuration Registers (see Figure 34.24).

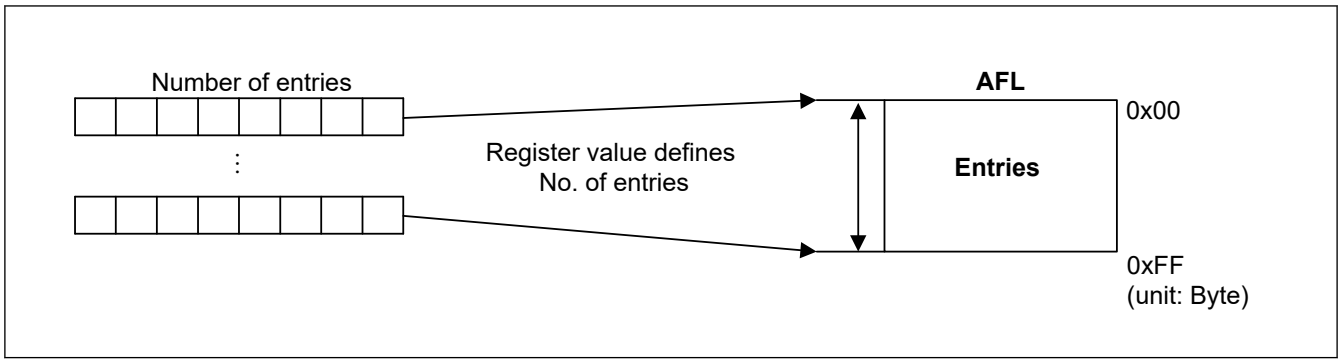


Figure 34.24 Configuration of AFL for each channel

The minimum number of entries for one channel is 0 (no entries defined for the channel) and the maximum number of entries is 16.

All entries are unique for a channel and overlapping or sharing of entries is not supported. Correct configuration of the AFL is the responsibility of the application.

The CANFD module does not flag errors related to the configuration of the AFL.

34.5.3 AFL Entry Description

Each AFL entry consists of 16 bytes. The fields in all entries are identical.

Each entry contains the following information for acceptance filtering and DLC filtering:

- Identifier (11 bits for Standard Frame format, 29 bits for Extended Frame format):
  Acceptance filter unit checks the identifier field of the received message against the identifier field of each AFL entry (full 29 bits masking of identifier bits is possible, see information that follows).
- IDE bit:
  Acceptance filter unit checks the IDE bit of the received message against this bit and selects the relevant part of the identifier field for acceptance filtering (masking of IDE bit is possible, see the information that follows).
- RTR bit:
  Acceptance filter unit only accepts data frames (RTR = 0) or remote frames (RTR = 1) according to the setting of this bit (masking of RTR bit is possible, see the information that follows).
- Loopback Configuration bit:
  This bit can enable or disable the AFL entry depending on the Loopback Configuration or Mirror mode condition.
- Mask for Identifier bits (29 bits):

Each bit in the identifier mask can mask the corresponding identifier bit in the AFL entry during acceptance filtering, see Figure 34.25.

- Mask for IDE bit:
  If this Mask bit masks the IDE bit of the AFL entry in both Standard Identifier and Extended Identifier format, messages can be accepted by this AFL entry. The identifier of the received message is compared against the Standard Identifier part of the AFL entry for Standard Identifier format messages and against the Extended Identifier part of the AFL entry for Extended Identifier format messages.

- Mask for RTR bit:
  If this Mask bit masks the RTR bit of the AFL entry in both frame formats, data frame and remote frame formats are accepted by this AFL entry.

- Pointer information (16 bits):
  This 16-bit pointer is attached to a received message accepted by the related AFL entry. The pointer is added during message storage in the message buffer area and can be used by application as support function. The pointer information can be used for example to support PDU identifier allocation for the received message in AUTOSAR systems.

- Information label (2 bits):
  This 2-bit label is attached to a received message accepted by the related AFL entry. The label is added during message storage in the message buffer area and can be used by application as support function.

- DLC value for automatic DLC filtering:
  If the DLC value of the received message is equal or higher than the configured DLC value, the DLC check is passed.

If the DLC value in this AFL entry is configured to 0, DLC filtering is effectively disabled for this entry (all accepted messages pass DLC filtering).

Each AFL entry contains the following information for the handling of received messages:

- Message buffer number of one single reception message buffer as target for received message storage
- Single reception message buffer enable bit to configure the single reception message buffer number to be valid or invalid, as target for received message storage
- FIFO direction pointer - each bit of the FIFO direction pointer configures a dedicated FIFO as possible target for a received message

There is no hardware protection against such storage of message. Therefore, the FIFO direction pointer must be configured carefully.

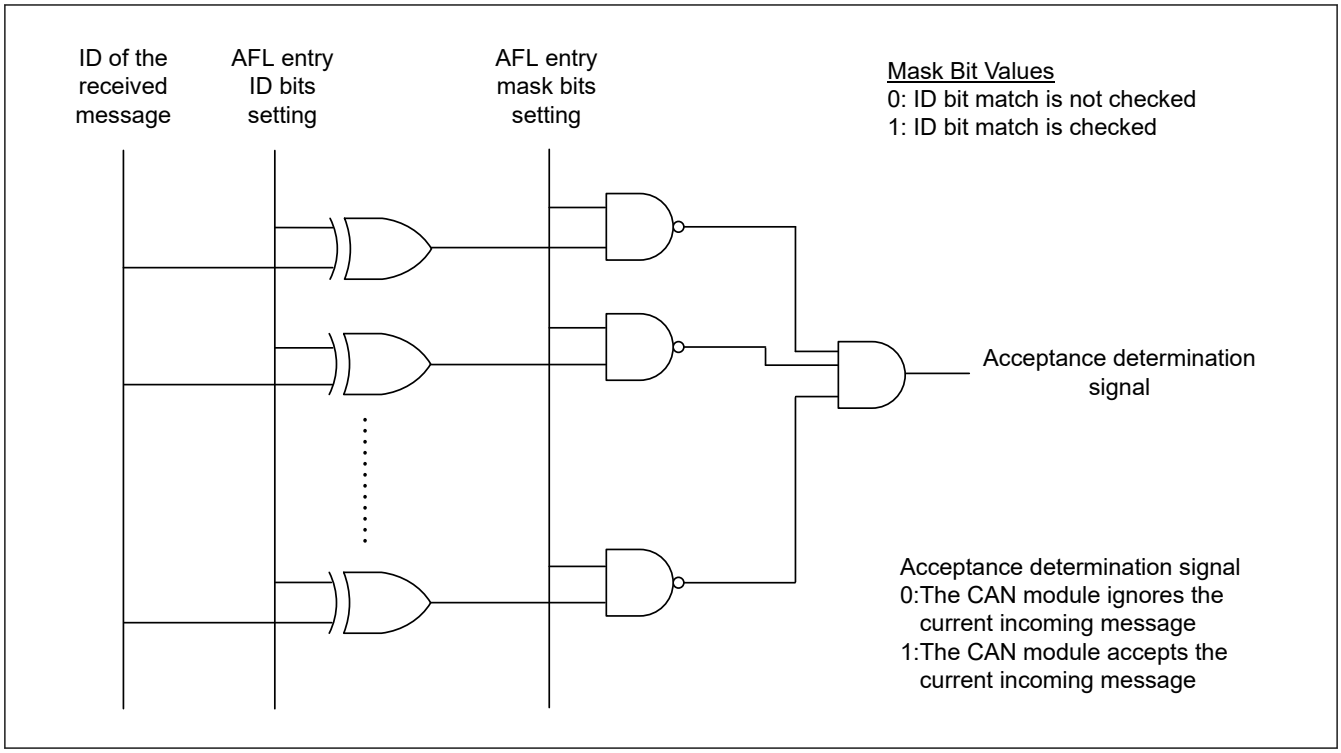


Figure 34.25 Acceptance function

34.5.4 Entering Entries in the AFL

Application software can enter one full entry into the AFL using the following registers:

- Global AFL ID Entry Register: Part 1 of the AFL entry
- Global AFL Mask Entry Register: Part 2 of the AFL entry
- Global AFL Pointer 0 Entry Register: Part 3 of the AFL entry
- Global AFL Pointer 1 Entry Register: Part 4 of the AFL entry.

16 sets of these registers form a group of AFL entries. The AFL should only be configured in CH_RESET or CH_HALT mode.

Follow the configuration shown in Figure 34.26 to program the AFL.

After entering all entries in Configuration mode, locking of the AFL access should be performed to protect unwanted write access to the AFL.

Write protection is active during all Global modes (GL_RESET, GL_HALT, and GL_OPERATION) if the lock bit is set.

Read access to AFL is still possible during all Global modes even when AFL data access is disabled (consistency check of AFL contents is possible during run time).

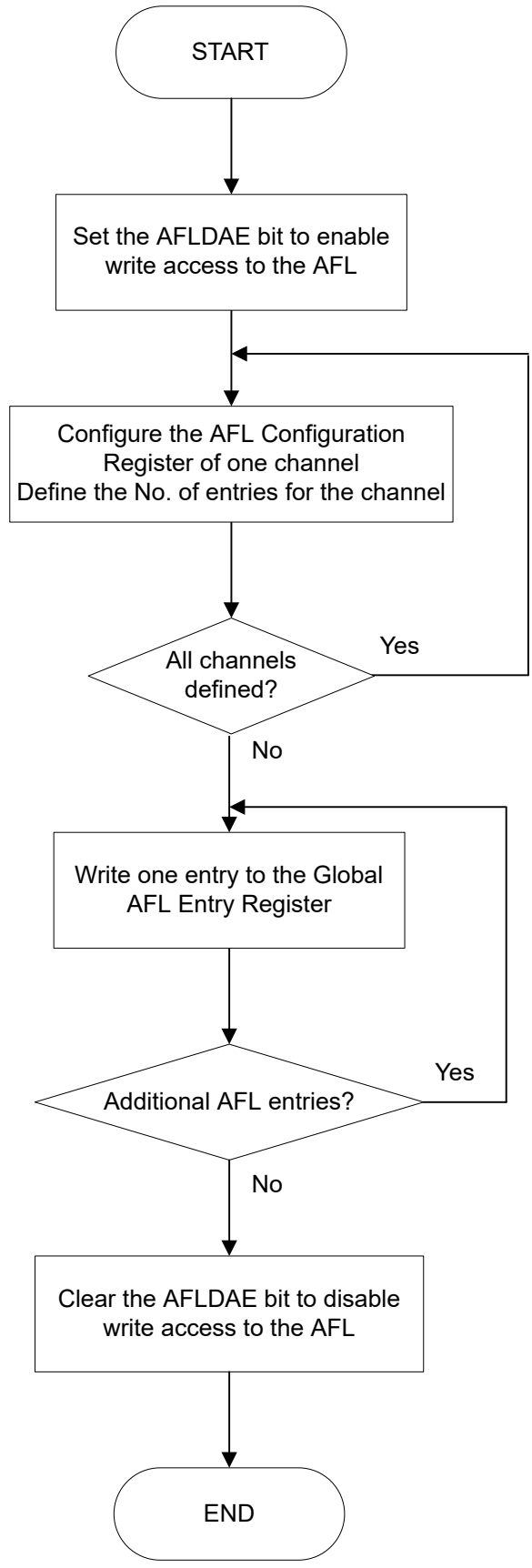


Figure 34.26 AFL configuration flow

34.5.5 Loopback Modes

If the Loopback Configuration bit is set, the AFL entry is only valid in Loopback test mode (Self-test mode 0 or Self-test mode 1) or in mirror mode when receiving messages that were transmitted by the respective CAN channel itself.

The AFL entry is not valid for received messages in loopback mode transmitted by other CAN nodes on the bus. The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID respectively.

If the Loopback Configuration bit is 0, the AFL entry is only valid for:

- Received messages transmitted by other CAN nodes on the bus in normal (non-loopback mode) and mirror modes
- Received messages transmitted by other CAN nodes or the CAN channel itself in Loopback test mode.

The mirror mode can be enabled with the CFDGCFG.MME bit in the Global Configuration Register. If CFDGCFG.MME bit is set, then a successfully transmitted message can be stored back in an RX message buffer or FIFO buffer if a matching entry is configured in the AFL for that channel.

The Loopback Configuration bit in the matching AFL entry must be set to store this frame.

If Mirror mode and Loopback test mode are configured at the same time, the Loopback test mode behavior applies.

Table 34.22 shows the behavior of the acceptance filter unit depending on the setting of the related input signals.

Table 34.22 Behavior of acceptance filter based on the loopback configuration setting in AFL entry

Mirror Mode Enable (MME Configuration bit)	Loopback in test mode (Self-test mode 0 or Self-test mode 1)	Channel mode	Loopback Configuration bit in AFL entry	AFL entry
0	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Invalid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid
1	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Valid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid

Note: The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID, respectively.

34.5.6 IDE Masking

When the GAFLIDEM bit is 0 in an AFL entry, the IDE bit configured in the AFL entry is not used for ID matching. In this case, the use of ID[10:0] or ID[28:0] matching is based on the received IDE bit.

Consider the following example:

- The ID and Mask fields of an AFL entry x is configured as follows:
  - CFDGAFLID [x] = 0xC0553A20 → IDE = 1, RTR = 1, LLB = 0, ID[10:0] = 0x220 / ID[28:0] = 0x00553A20
  - CFDGAFLMr = 0x0000FFFF → IDEM = 0. RTRM = 0, IDM[10:0] = 0x7FF / IDM[28:0] = 0x0000FFFF

- The comparison result for the four different received IDs with AFL entry x is described as follows:
  - If a frame with IDE = 0 and ID = 0x220 is received, this is considered as a match
  - If a frame with IDE = 0 and ID = 0x320 is received, this is not a match
  - If a frame with IDE = 1 and ID = 0x1FFF3A20 is received, this is considered as a match
  - If a frame with IDE = 1 and ID = 0x08803220 is received, this is not a match.

34.5.7 Updating AFL Entry during Communication

You can update the AFL entry without disabling all CAN communications. Choose the entry number to be updated by setting the AFL entry number, and ignore the enable bit.

This entry number is ignored from the AFL matching while the entry is being updated.

Figure 34.27 shows the update flow for an AFL entry.

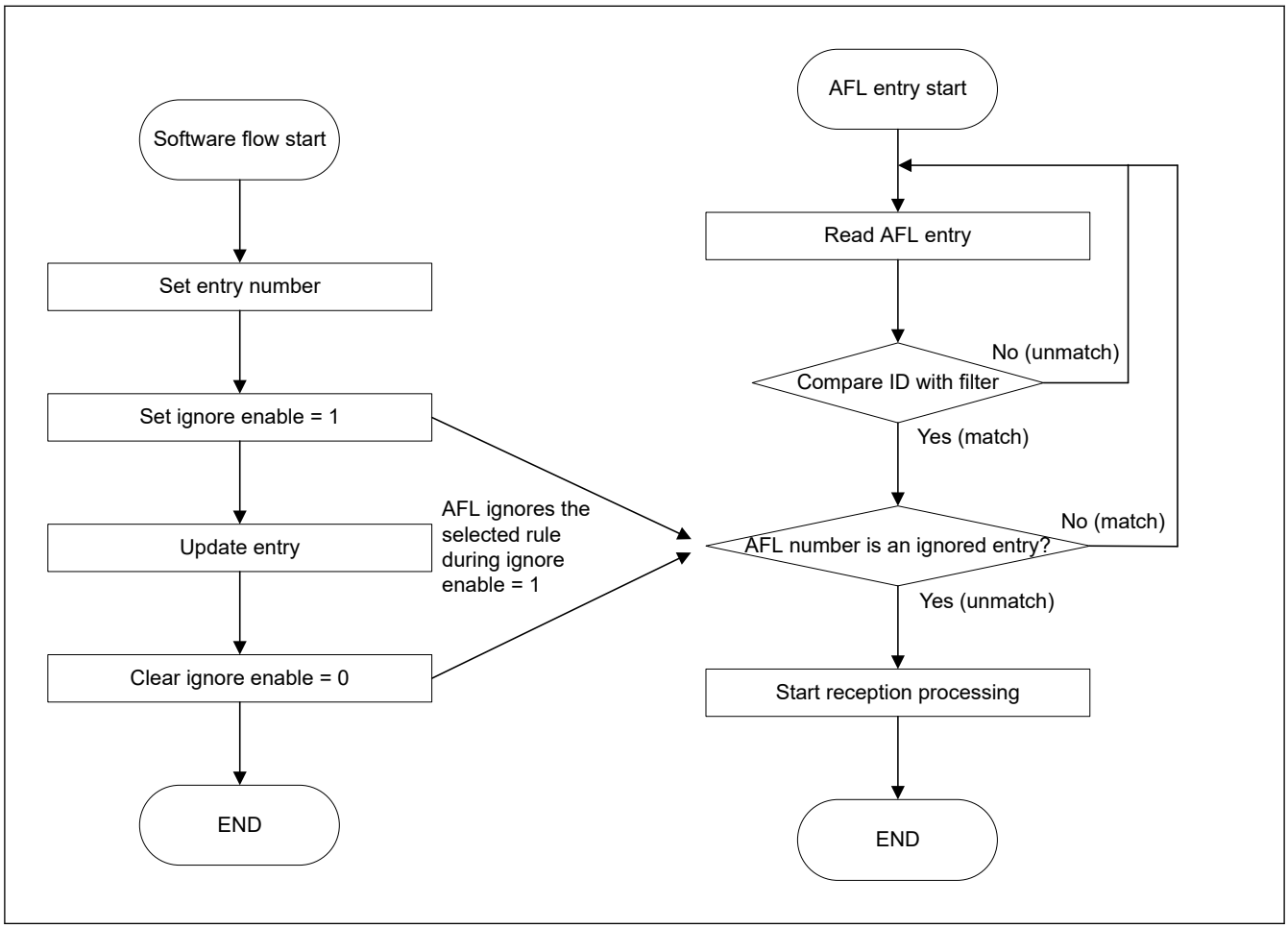


Figure 34.27 Update flow for an AFL entry

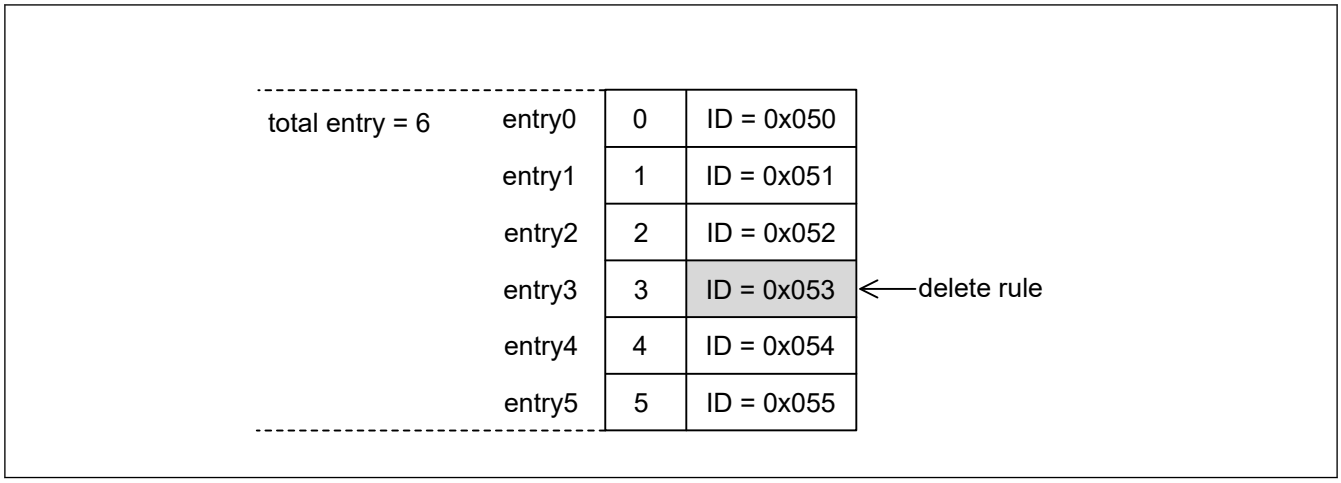
The method to update an AFL entry is as follows:

1. Set the entry number to CFDGAFLIGNENT register.
2. Set the value 0xC401 (key code and enable bit) to CFDGAFLIGNCTR register.
3. CFDGAFLECTR.AFLDAE is set to 1.
4. Set the new rule to CFDGAFLIDr, CFDGAFLMr, CFDGAFLP0r, CFDGAFLP1r registers.
5. CFDGAFLECTR.AFLDAE is cleared to 0.
6. Set the value 0xC400 (key code and clear enable bit) to CFDGAFLIGNCTR register.

Note: This entry number is ignored during the periods from (2) to (5).

(1) Example 1: Deleting an entry

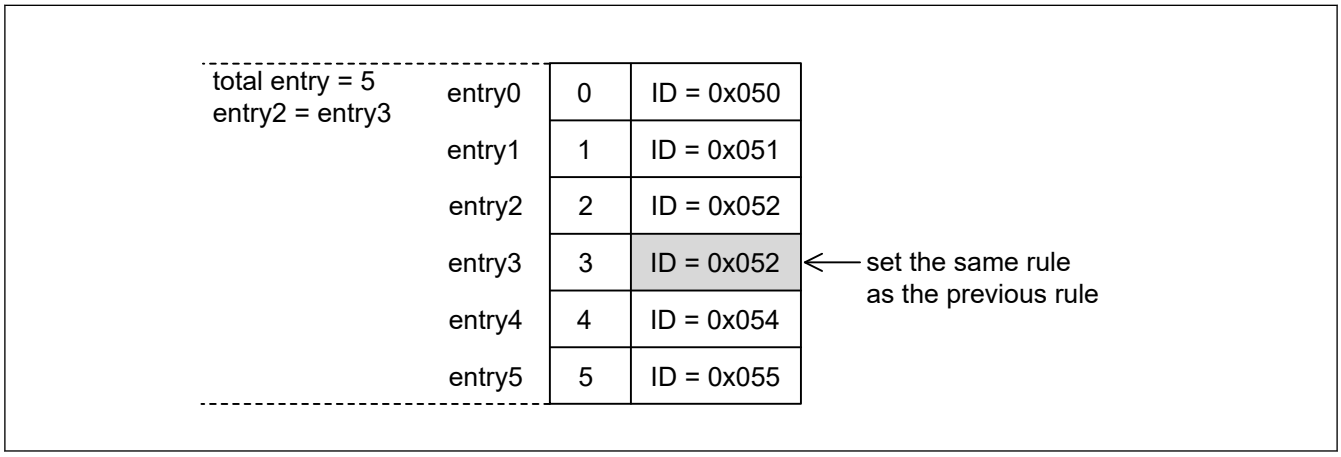
Deleting entry3 when the total number of entries is 6 channel.



How to delete an entry

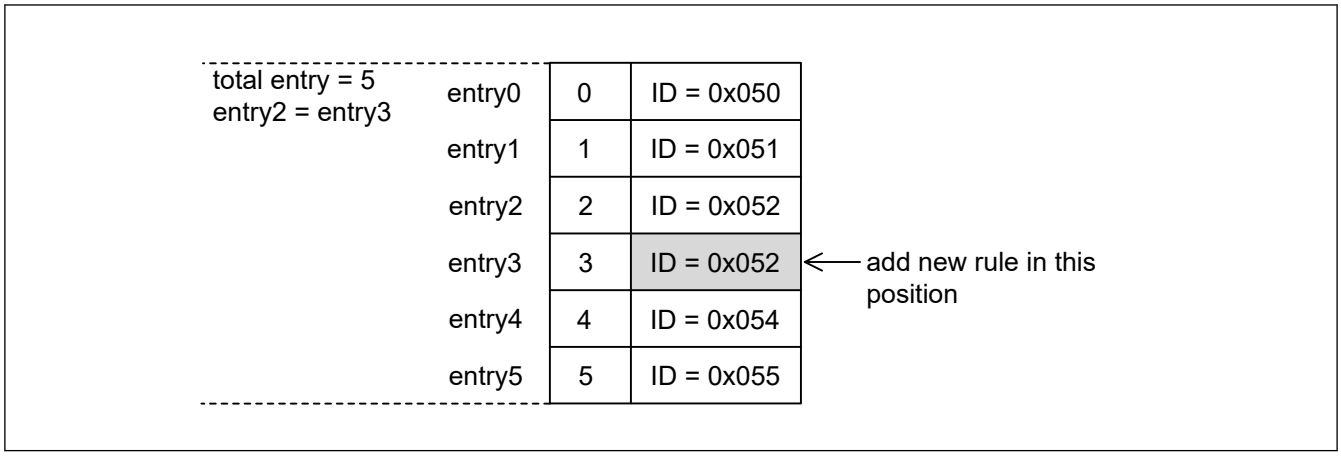
1. Set 0x00000003 to CFDGAFLIGNENT register.
2. Set 0x0000C401 to CFDGAFLIGNCTR register.
3. Set 0x00000100 to CFDGAFLECTR register.
4. Set the same rule as the previous rule by accessing CFDGAFLIDr, CFDGAFLMr, CFDGAFLP0r, CFDGAFLP1r (r = 3, this is entry3).
5. Set 0x00000000 to CFDGAFLECTR register.
6. Set 0x0000C400 to CFDGAFLIGNCTR register.

Entry3 is now deleted.



(2) Example 2: Adding an entry

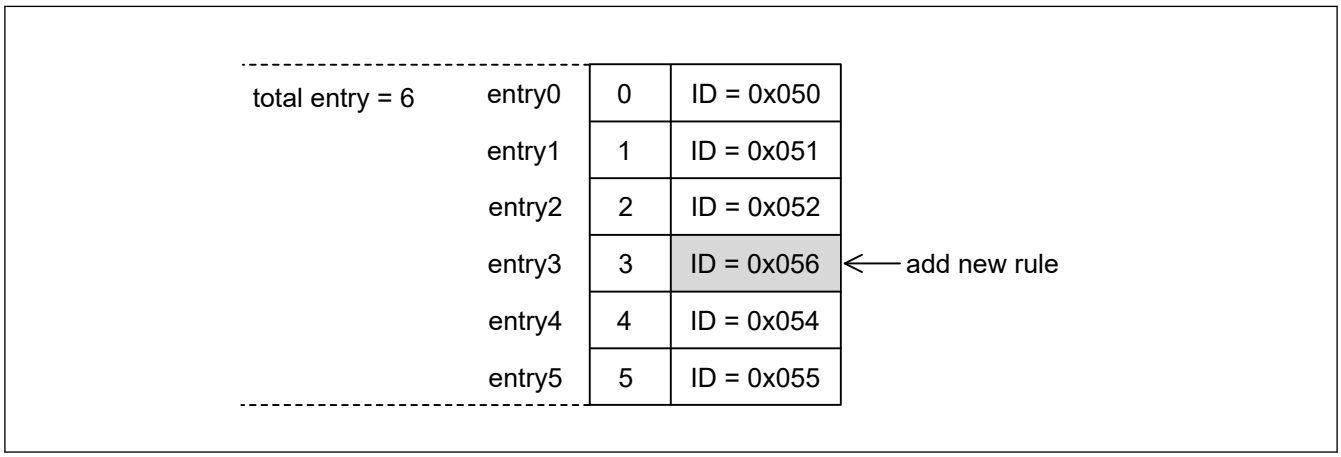
Adding a new entry to entry3 when the total number of entries is 6.



How to add an entry

1. Set 0x00000003 to CFDGAFLIGNENT register.
2. Set 0x0000C401 to CFDGAFLIGNCTR register.
3. Set 0x00000100 to CFDGAFLECTR register.
4. Set the new rule by accessing CFDGAFLIDr, CFDGAFLMr, CFDGAFLP0r, CFDGAFLP1r (r = 3, this is entry3).
5. Set 0x00000000 to CFDGAFLECTR register.
6. Set 0x0000C400 to CFDGAFLIGNCTR register.

The new entry is now added.



The AFL filter can be used to set CFDGAFLCFG, and addition/deletion of an entry is possible. Therefore, it is necessary to set the maximum number to be used to CFDGAFLCFG.

34.6 FIFO Buffers and Normal Message Buffer Configuration

This section describes the process for configuring the number of RX message buffers, the FIFO buffers, and the flat TX message buffers in the CANFD module. The message buffers are mapped as shown in Figure 34.28.

The RX message buffers can be accessed with the RX Message Buffer Registers.

The RX FIFO buffers and the common FIFO buffers configured in RX mode or TX mode can only be accessed with the FIFO Access Registers.

If the common FIFO is configured in TX mode, you can only write data into the FIFO buffer using the FIFO Access registers.

If the common FIFO is configured in RX mode, you can only read data from the FIFO Access Registers.

The TX message buffers can be accessed with the TX Message Buffer Registers.

If unused message buffer locations are read, the message buffer locations are read as unknown values.

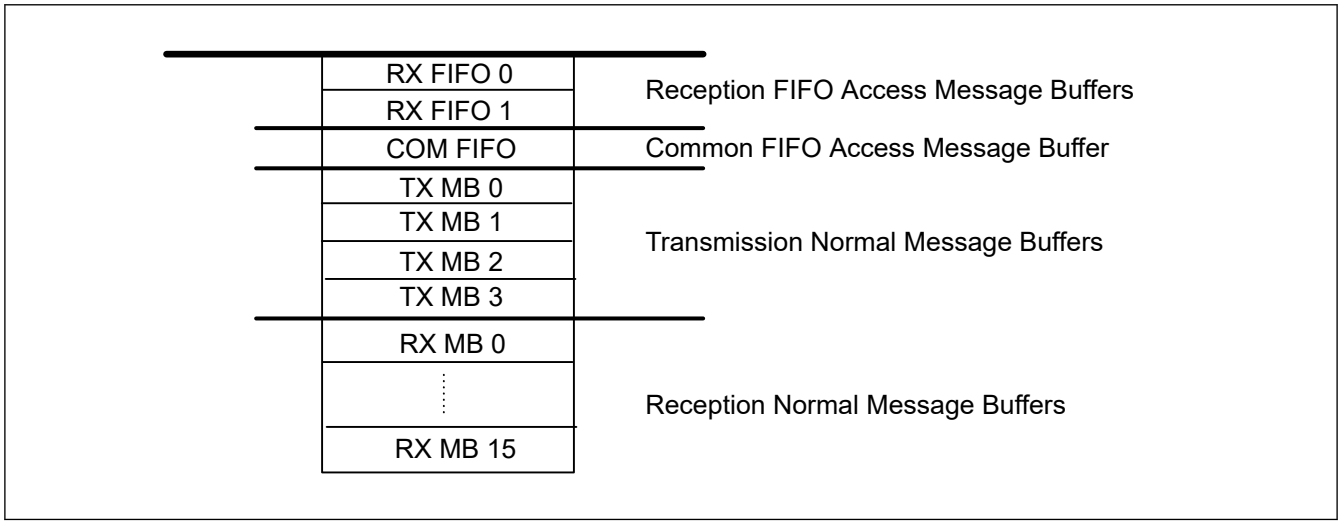


Figure 34.28 Message buffer configuration

34.6.1 Normal RX Message Buffers

In CANFD module, the frames received can be stored in normal RX message buffers based on the configuration of the AFL entries.

Additionally, the number of normal RX message buffers required in the system can be chosen up to a fixed maximum limit.

34.6.1.1 Normal RX Message Buffer Configuration

In CANFD module, the number of normal RX message buffers can be configured by writing to the RX Message Buffer Number Register.

The limiting values for the configuration of number of message buffers are:

- Minimum value = 0x00 (no normal RX MB)
- Maximum value = 0x10

Do not use values outside these limits.

The AFL entries for routing the received messages to normal RX message buffers must be configured to match the requirements of the system.

The AFL entries must also be configured properly, and an AFL entry for normal RX message buffers should not exceed the number of message buffers configured in the RX Message Buffer Number Register.

Note: There is no internal check procedure provided in CANFD module against wrong configuration of the AFL.

The data field size of the RX message buffer can be configured with the CFDRMNB.RMPLS bit. The default size is 8 bytes and the maximum data payload size is 64 bytes.

When the receiving frame exceeds the data field size, then the acceptance depends on the configuration of CFDGCFG.CMPOC (message rejecting or data payload cut).

Note: RMPLS and CMPOC bit is not available in the classical CAN function, so, these feature is not valid for classical CAN.

34.6.2 FIFO Buffers

The CANFD module provides a fixed number of FIFO buffers to support storage of frames for reception and transmission functions .

The number of reception-only FIFO buffers is fixed to 2. However, common FIFO buffer channel can be configured to store messages for transmission or reception function.

These FIFO buffers can be enabled or disabled, and the following parameters can be configured to match the system requirements:

- Size
- Interrupt structure
- Message lost mechanism
- Message overwrite mechanism of the FIFO buffers
- Location of the TX FIFO.

When the receiving frame exceeds the data field size, the acceptance depends on the configuration of the CFDGCFG.CMPOC bit (message rejecting or data payload cut).

34.6.2.1 FIFO Buffers Configuration

In CANFD module, the FIFO buffers can be configured to match the system requirements.

The total number of FIFO buffers = 2 RX FIFO buffers + 1 common FIFO buffer = 3 FIFO buffers.

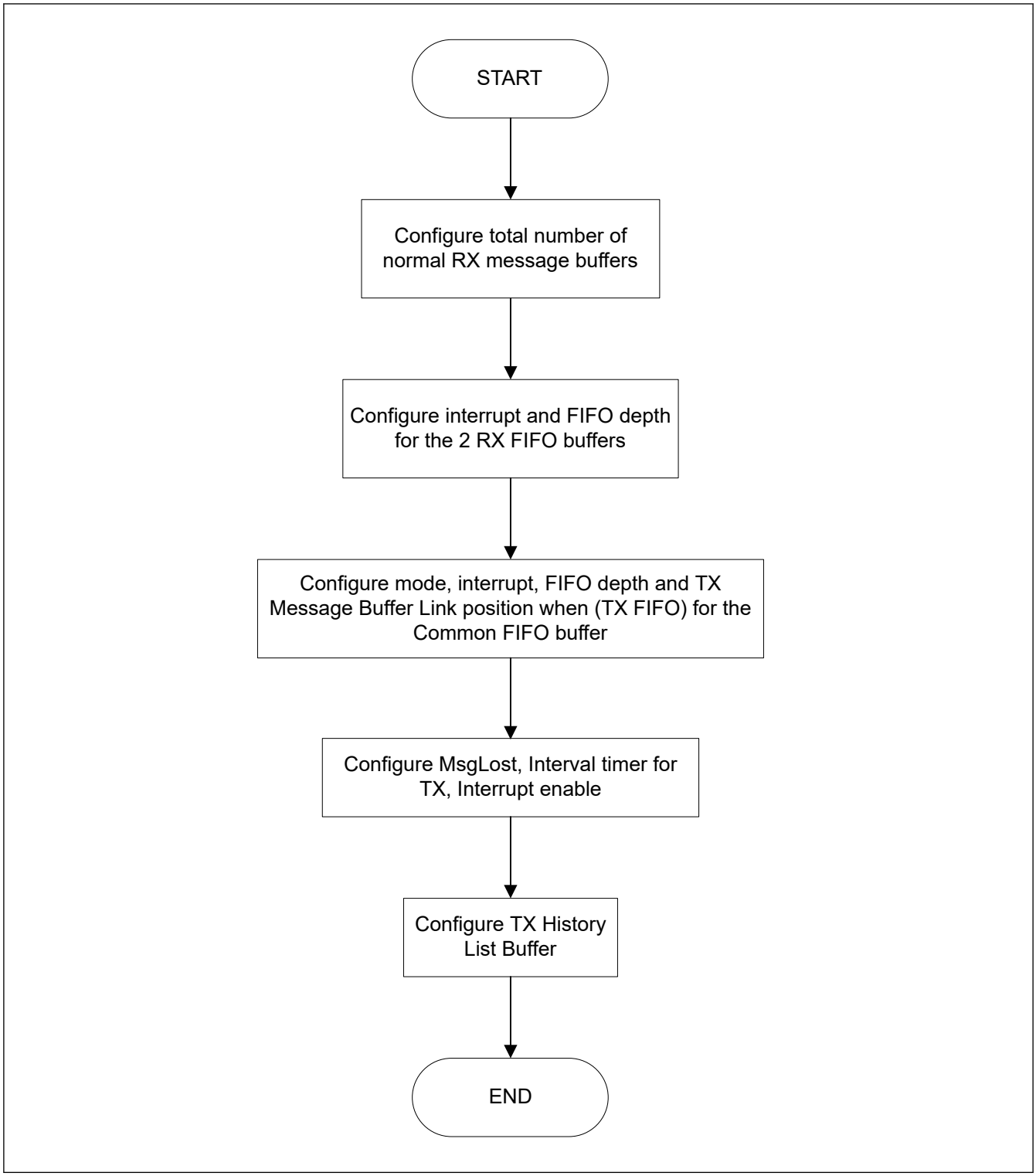


Figure 34.29 FIFO buffer configuration flow in CANFD module

As shown in Figure 34.29, the various FIFO buffers can be configured by writing to the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers.

For the 2 RX FIFO buffers, the following parameters can be configured:

- Interrupts
- FIFO depth
- FIFO payload data size.

For the common FIFO buffer, the following parameters can be configured:

- Mode
- Interrupts FIFO depth
- FIFO payload data size
- FIFO TX link position.

(1) FIFO mode configuration of Common FIFO buffer

The mode of the common FIFO buffer can be configured by writing to the CFDCFCC.CFM[1:0] bits in the Common FIFO Configuration/Control Register. The possible modes of configuration for Common FIFO buffer are:

- 0b RX mode (default mode after hardware reset)
- 1b TX mode

Messages can only be read from the RX FIFO buffers and the Common FIFO buffer configured in RX mode. Messages are stored by the CAN module in these FIFO buffers based on the AFL entries.

Messages can be read and written into the Common FIFO buffer configured in TX mode. These messages are transmitted on the appropriate CAN channel.

The pointers can only be incremented when a new message is stored in the FIFO buffer and decremented when a message is transmitted on the corresponding CAN channel by the CANFD module.

After a hardware reset, the Common FIFO buffer is configured in RX mode by default. Only enable the FIFO buffers after configuring the Common FIFO buffer in the required modes.

(2) FIFO TX message buffer link configuration

When the common FIFO is configured as TX FIFO, the FIFO buffer must be linked to a normal TX message buffer to participate in the transmission scan.

Do not write data into a TX message buffer that is linked to a Common FIFO buffer. Also, the TX message buffer linked to a Common FIFO buffer should not be a part of the TX Queue.

The TX message buffer link of each Common FIFO buffer can be configured by writing to the CFDCFCC.CFTML[1:0] bits in the Common FIFO Configuration/Control Registers. Available options for TX message buffer link configuration are:

- 0x00: TX Message Buffer 0
- 0x01: TX Message Buffer 1
- 0x10: TX Message Buffer 2
- 0x11: TX Message Buffer 3

(3) FIFO depth configuration

The depth of each FIFO buffer can be configured by writing to the CFDRFCCa.RFDC[2:0] bits and CFDCFCC.CFDC[2:0] bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers. The 6 available options for depth configuration are:

- 0x000: 0 Message (FIFO buffer cannot be enabled)
- 0x001: 4 Messages
- 0x010: 8 Messages
- 0x011: 16 Messages
- 0x100: 32 Messages
- 0x101: 48 Messages

The RAM allocation for RX message buffers along with FIFO buffers is limited to 16 messages with 64 data bytes (76 bytes including ID and PTR). Configuration of the RX message buffers, along with FIFO buffers, that exceeds this maximum limit should not be done.

CANFD module logic does not check the validity of the configuration.

Note: If the FIFO depth of a common FIFO is 4 messages or more (CFDCFCC.CFDC[2:0] > 000b), then the Common FIFO TX message buffer link is valid when the FIFO is disabled or enabled.

If FIFO depth is 0 messages, then the Common FIFO TX message buffer link is not valid when the FIFO is disabled or enabled.

(4) FIFO payload size configuration

The data size of each FIFO buffer can be configured by writing to the CFDRFCCa.RFPLS[2:0] bits and CFDCFCC.CFPLS[2:0] bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers. The eight available options for depth configuration are:

- 000b: 8 bytes
- 001b: 12 bytes
- 010b: 16 bytes
- 011b: 20 bytes
- 100b: 24 bytes
- 101b: 32 bytes
- 110b: 48 bytes
- 111b: 64 bytes

The RAM allocation for RX message buffers along with FIFO buffers is limited to 16 messages with 64 data bytes (76 bytes including ID and PTR). Configuration of the RX message buffers, along with FIFO buffers, that exceeds this maximum limit should not be done.

CANFD module logic does not check the validity of the configuration.

Note: This feature is not available in the classical CAN function.

(5) FIFO interrupt configuration

The Interrupt generation conditions for the FIFO buffers can be configured by writing to the CFDRFCCa.RFIM and CFDCFCC.CFIM bit in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers. The two available options are:

- 0:
  - RX FIFO mode: Interrupt generated when the Common FIFO counter reaches CFDRFCCa.RFIGCV/ CFDCFCC.CFIGCV value
  - TX FIFO mode: Interrupt generated when the Common FIFO transmits the last message successfully
- 1:
  - RX FIFO mode: Interrupt generated at the end of storage of every received message
  - TX FIFO mode: Interrupt generated for every successfully transmitted message

If the Interrupt Mode bit is 0 for a RX FIFO, then interrupt is generated based on the configuration of the CFDRFCCa.RFIGCV[2:0] bits.

Similarly, if the Interrupt Mode bit is 0 for a Common FIFO configured in RX mode, then interrupt is generated based on the configuration of CFDCFCC.CFIGCV[2:0] bits.

The eight available options for configuring the FIFO counter value for generation of an interrupt are:

- 000b: Interrupt generated when FIFO is 1/8th Full
- 001b: Interrupt generated when FIFO is 1/4th Full
- 010b: Interrupt generated when FIFO is 3/8th Full
- 011b: Interrupt generated when FIFO is 1/2 Full
- 100b: Interrupt generated when FIFO is 5/8th Full
- 101b: Interrupt generated when FIFO is 3/4th Full

- 110b: Interrupt generated when FIFO is 7/8th Full
- 111b: Interrupt generated when FIFO is Full.

In this case, an interrupt is generated when the message count matches the configured value.

However, there are some limitations on the configuration of the CFDRFCCa.RFIGCV[2:0] and CFDCFCC.CFIGCV[2:0] bits depending on the FDC[2:0] bits (FIFO Depth Configuration), see Table 34.23.

Table 34.23 FIFO interrupt generation counter and FIFO depth configuration

RFDC[2:0] (CFDC[2:0])	RFIGCV[2:0] (CFIGCV[2:0])							
	111b	110b	101b	100b	011b	010b	001b	000b
000b	Don't care (FIFO cannot be enabled)							
001b	Allowed	Not allowed	Allowed	Not allowed	Allowed	Not allowed	Allowed	Not allowed
010b	Allowed							
011b	Allowed							
100b	Allowed							
101b	Allowed							
110b	Allowed							
111b	Allowed							

34.6.2.2 FIFO Buffers Control

The FIFO interrupt must be enabled by setting any one of the following bits in the RX FIFO Configuration/Control Registers:

- CFDRFCCa.RFIE

In addition, the FIFO interrupt must be enabled by setting any one of the following bits in the Common FIFO Configuration/Control Register:

- CFDCFCC.CFRXIE
- CFDCFCC.CFTXIE

After configuration is complete, each FIFO can be enabled by setting the CFDRFCCa.RFE and CFDCFCC.CFE bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Register to allow transmission and reception of messages.

34.7 Interrupts and DMA

34.7.1 Interrupts

The CANFD module generates several interrupts. The interrupt output, which is connected to the Interrupt Controller Unit (ICU), can be controlled by the corresponding interrupt enable bit.

The status flag is set independent from this enable bit.

The channel transmission interrupt has an additional status flag register. The status bits are set when the corresponding interrupt enables are set.

The status flag register supports the identification of the interrupt source for the channel transmission, as this interrupt is driven by several trigger sources.

The interrupts in the CANFD module can be classified into two groups, global interrupts and channel interrupts:

- Global interrupts:
  The CANFD module can generate 3 global interrupts:
  - Global interrupt for successful reception into the 2 RX FIFO buffers
  - Global error interrupt.
  - Global Interrupt for successful reception into the 16 RX message buffers

- Channel interrupts:
  Channel of the CANFD module can generate 3 channel interrupts:
  1. Channel transmission
     - Transmission completion from channel
     - Transmission abort from channel
     - Transmission from TX Queue for a channel
     - Channel THL interrupt
     - Successful transmission from a Common FIFO in TX mode for a channel.
  2. Channel error interrupt
  3. Successful reception in a Common FIFO in RX mode for a channel.

The interrupts are cleared when the corresponding flag bits are cleared or the Interrupt enable bits are cleared.

Table 34.24 gives an overview of interrupt sources for the different interrupt outputs. The interrupt outputs are active-high.

Table 34.24 Interrupt source overview

Parameter	Interrupt	Name	Interrupt source	Interrupt clearing
Global Interrupts	Successful reception into at least one RX FIFO	CAN_RXF	Interrupt flag of corresponding RX FIFO for which interrupt is enabled	Clear the interrupt flag of corresponding RX FIFO buffer for which interrupt is enabled
	Global Error	CAN_GLERR	Any of the following: • DLC Error flag • Message Lost Status bit • TX History Entry Lost Status bit • CANFD Message Payload overflow flag	Clear all of : • DLC Error flag • Message Lost flags in all of the FIFO Status Registers • TX History List Entry Lost flag • CANFD Message Payload overflow flag
	Successful reception into at least one RXMB	CANn_RXMB (n = 0, 1)	Interrupt flag of corresponding RXMB for which interrupt is enabled	Clear the interrupt flag of corresponding RXMB buffer for which interrupt is enabled
Channel Transmission Interrupts	Channel successful transmission	CANn_TX (n = 0, 1)	Any channel related TXMB Successful flag when interrupt is enabled*1	Clear all channel related TXMB Result status bits for which the interrupt is enabled
	Channel Abort		Any channel related TXMB Abort flag when interrupt is enabled*1	Clear all channel related TXMB Result Status bits for which the interrupt is enabled globally
	Channel transmission from TX Queue		Related channel TX Queue Interrupt flag	Clear related channel TX Queue Interrupt flag
	Channel THL Interrupt		Channel THL Interrupt status flag	Clear the relevant THL Interrupt status flag
	Channel COM FIFO TX Interrupt		Interrupt Flag for Common FIFOs in TX mode belonging to the related channel	Clear the interrupt flags of Common FIFOs in TX mode belonging to the related channel
Channel Error Interrupt	Channel Error	CANn_CHERR (n = 0, 1)	Any channel related error flag in the Channel Error Flag Register for which interrupt is enabled in the Channel Error Interrupt Enable Register	Clear all channel related error flags in the Channel Error Flag Register for which interrupt is enabled in the Channel Error Interrupt Enable Register
Channel COM RX FIFO Interrupt	Channel COM FIFO RX Interrupt	CANn_COMFRX (n = 0, 1)	Interrupt flag for Common FIFOs in RX mode belonging to the related channel	Clear the interrupt flags of Common FIFOs in RX mode belonging to the related channel

Note 1. These interrupts are only set for TX Message Buffers that do not belong to anenabled TX Queue andare not pointing to a common FIFO.
Separate interrupts are provided for common FIFO buffers and TX Queue.

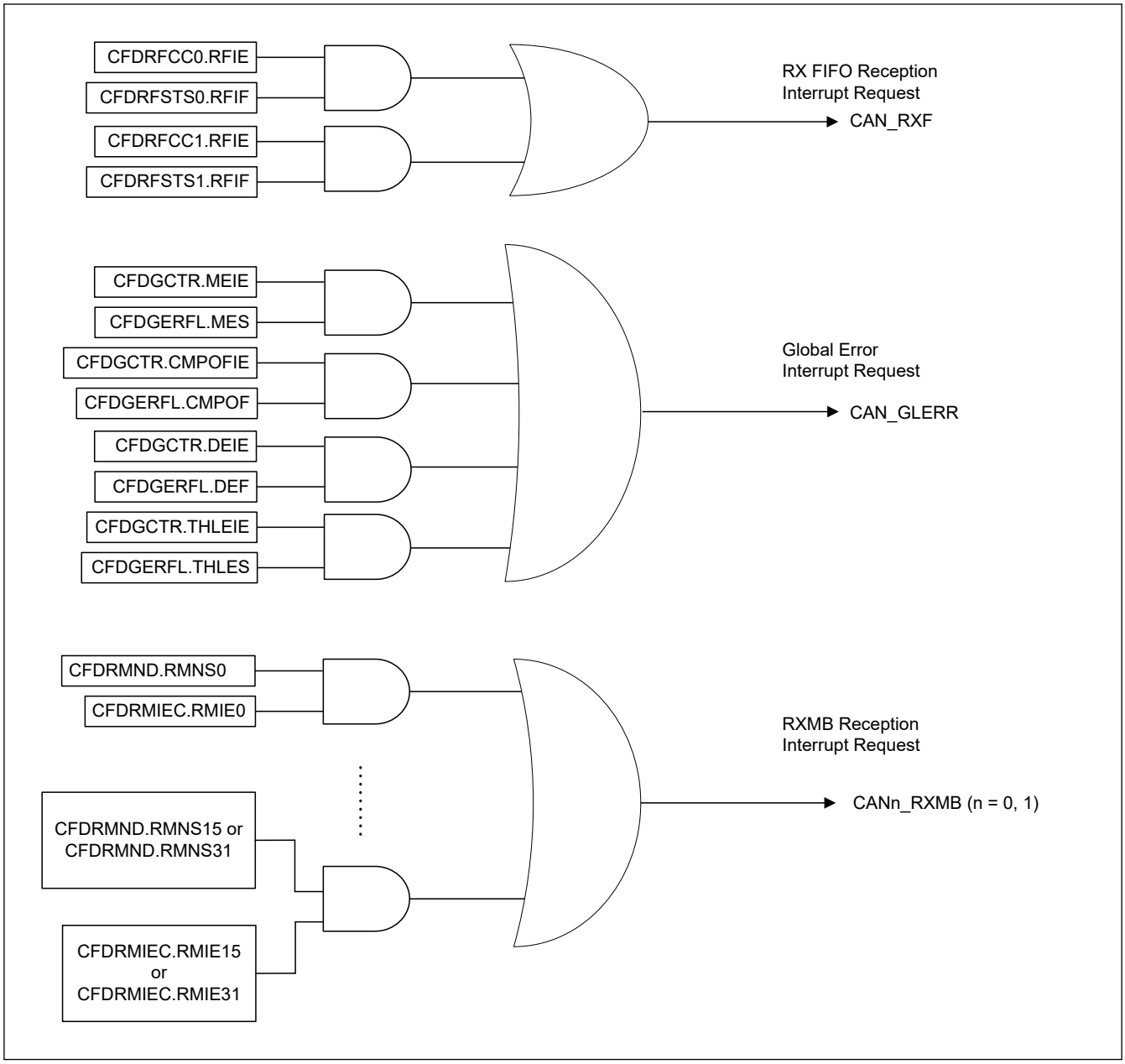


Figure 34.30 Global interrupt block diagram

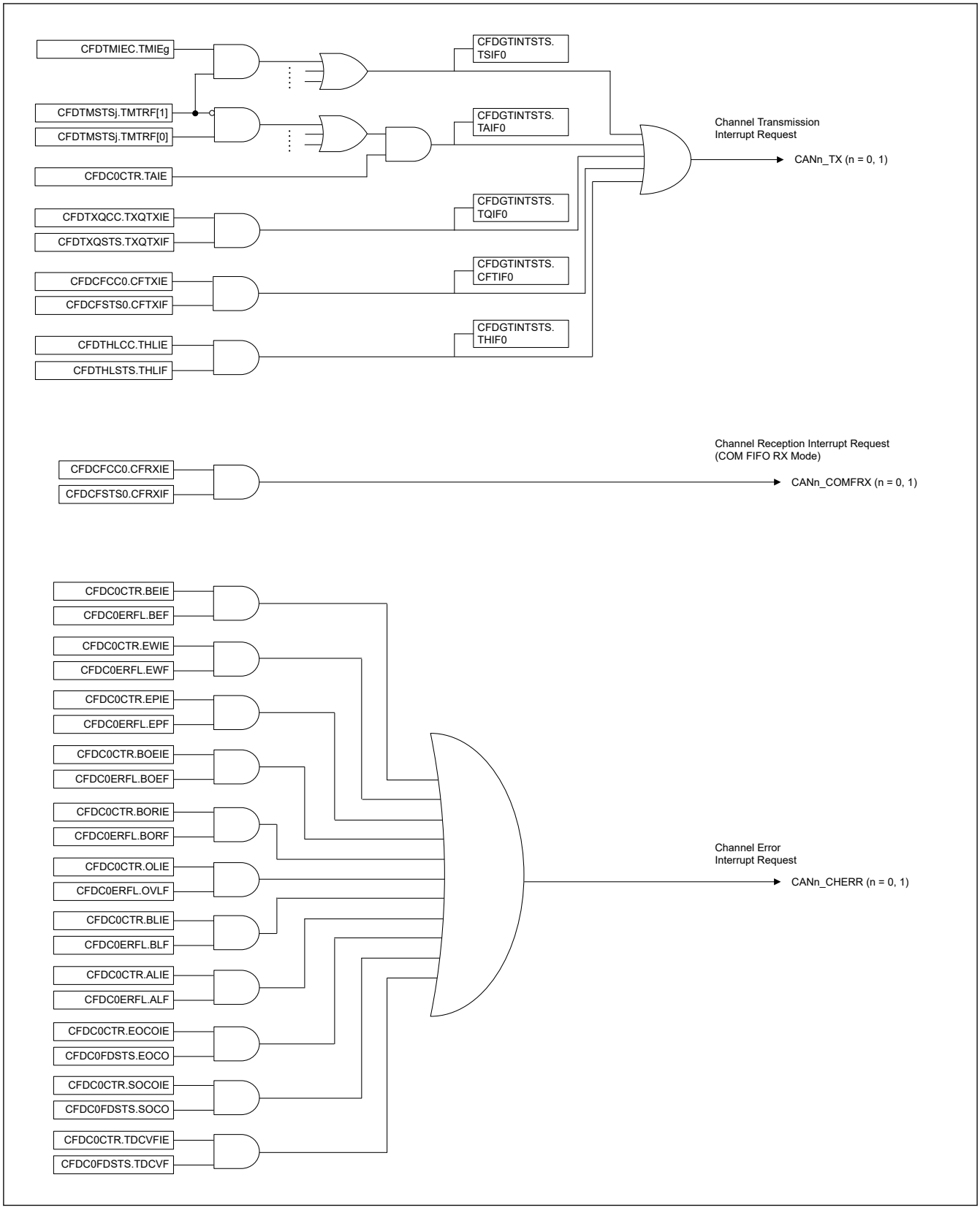


Figure 34.31 Channel interrupt block diagram

34.7.2 DMA Transfer

The CANFD module has message buffers that can be associated with a DMA channel:

- Reception DMA

– 2 RX FIFO message buffers
– Common FIFO Message Buffer

Figure 34.32 shows the potential DMA channels.

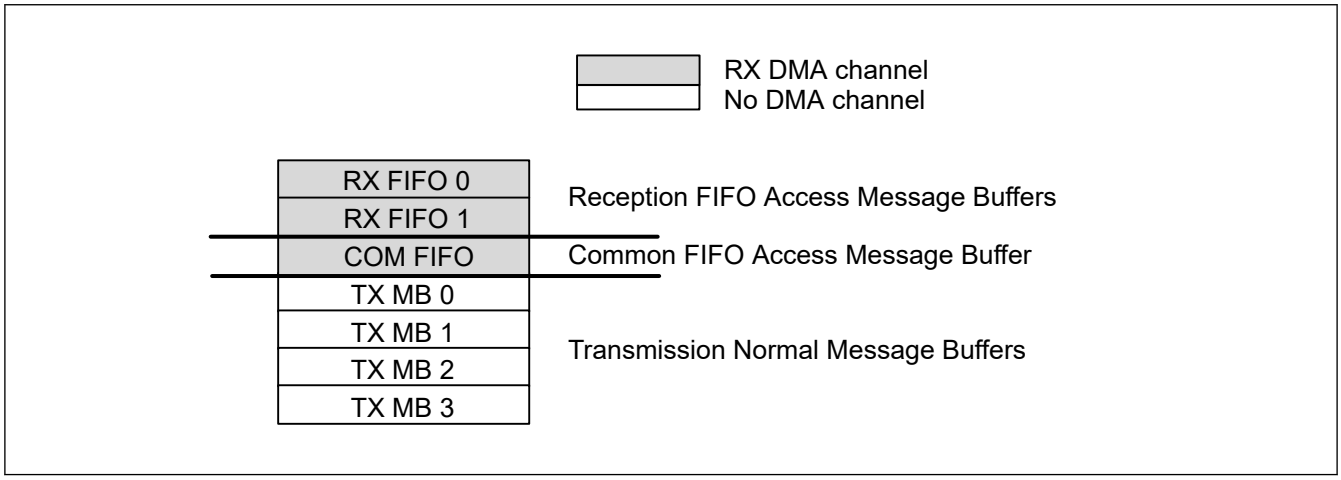


Figure 34.32 Message buffer connectable to a DMA channel

A DMA channel transfer request is generated for each FIFO entry to the DMAC when the related CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE is set to 1 and the belonging FIFO is not empty.

Reception FIFO Interrupt should be disabled for this particular FIFO (CFDRFCCa.RFIE or CFDCFCC.CFRXIE)

Use the regular start address for the DMA access window address. See Figure 34.33.

Table 34.25 DMA channel access window address

b = Message buffer component index	Message Buffer Component	Register	P	Regular Start Address
b = [0…1]	RFMBCPb[0]	CFDRFIDb	x	0x0520 + b × 0x004C
		CFDRFPTRb	x	0x0524 + b × 0x004C
		CFDRFFDSTSb	x	0x0528 + b × 0x004C
		CFDRFDFbp	[0…15]	0x052C + p × 0x0004 + b × 0x004C
—	CFMBCP0[0]	CFDCFID	x	0x05B8
		CFDCFPTR	x	0x05BC
		CFDCFFDCSTS	x	0x05C0
		CFDCFDFp	[0…15]	0x05C4 + p × 0x0004

DMA FIFO pointer decrement is done automatically by reading the last configured data payload byte (CFDRFCCa.RFPLS or CFDCFCC.CFPLS).

Note: The DMA must read the exact length of the configured data payload size (CFDRFCCa.RFPLS or CFDCFCC.CFPLS).

Note: This feature is not available for classical CAN function because CFDRFCCa.RFPLS and CFDCFCC.CFPLS are not in classical CAN.

Do not write to the FIFO control registers when DMA is enabled. The DMA enable of the particular DMA FIFO (CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE) can be set at any time. Figure 34.33 shows a configuration flow for an initial setup.

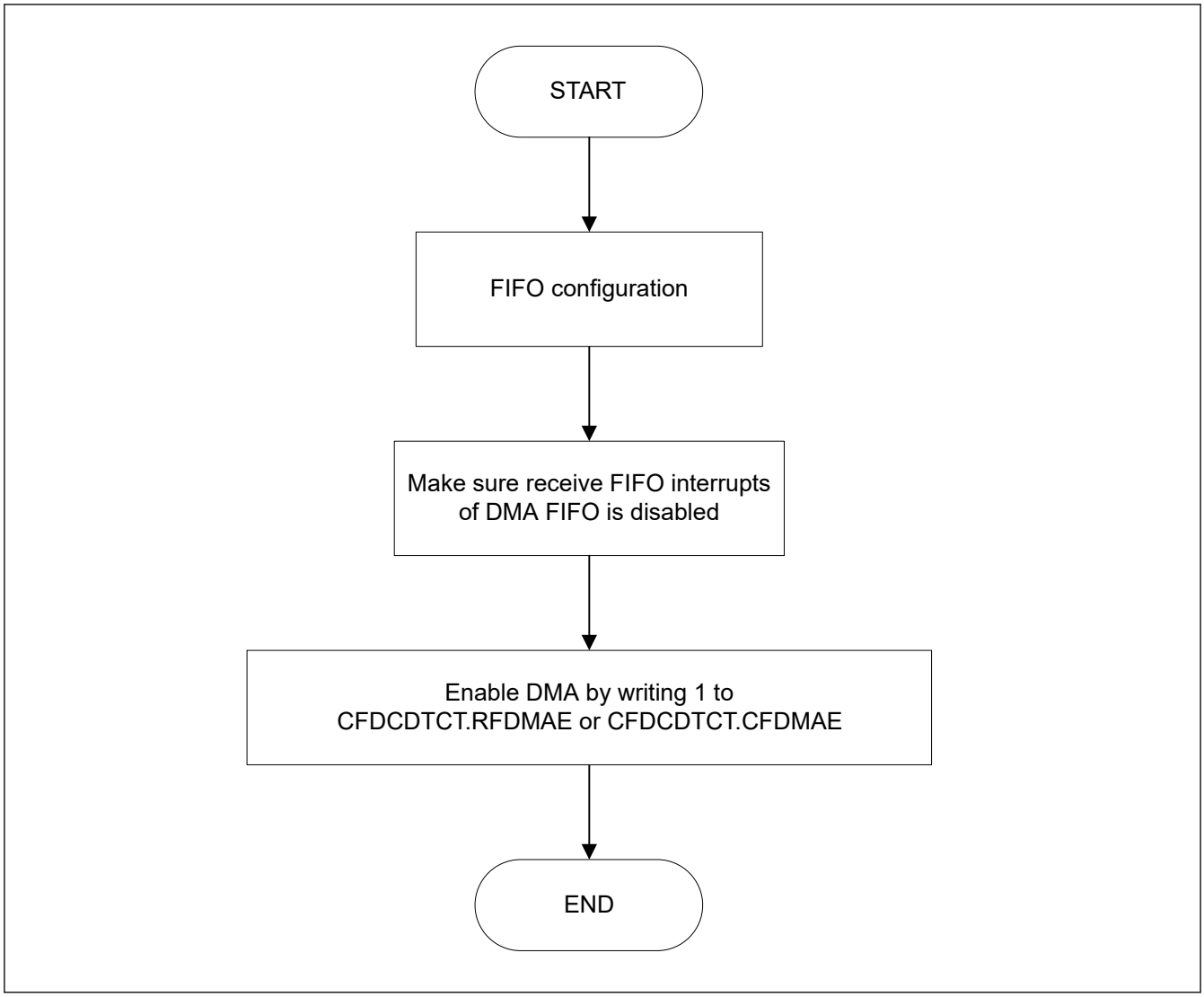


Figure 34.33 DMA enable flow

To disable a DMA transfer request, you must disable the particular DMA enable bit (CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE). If the disable is made during an ongoing transfer, then the transfer must be completed first before further action can be taken. The transfer status can be identified by the CFDCDTSTS.RFDMASTS or CFDCDTSTS.CFDMASTS bit. See Figure 34.34 for the DMA disable flow. When the DMA is disabled, consideration should be made for the remaining or new incoming messages to this particular reception FIFO.

When the FIFO is not disabled, reception to the FIFO continues.

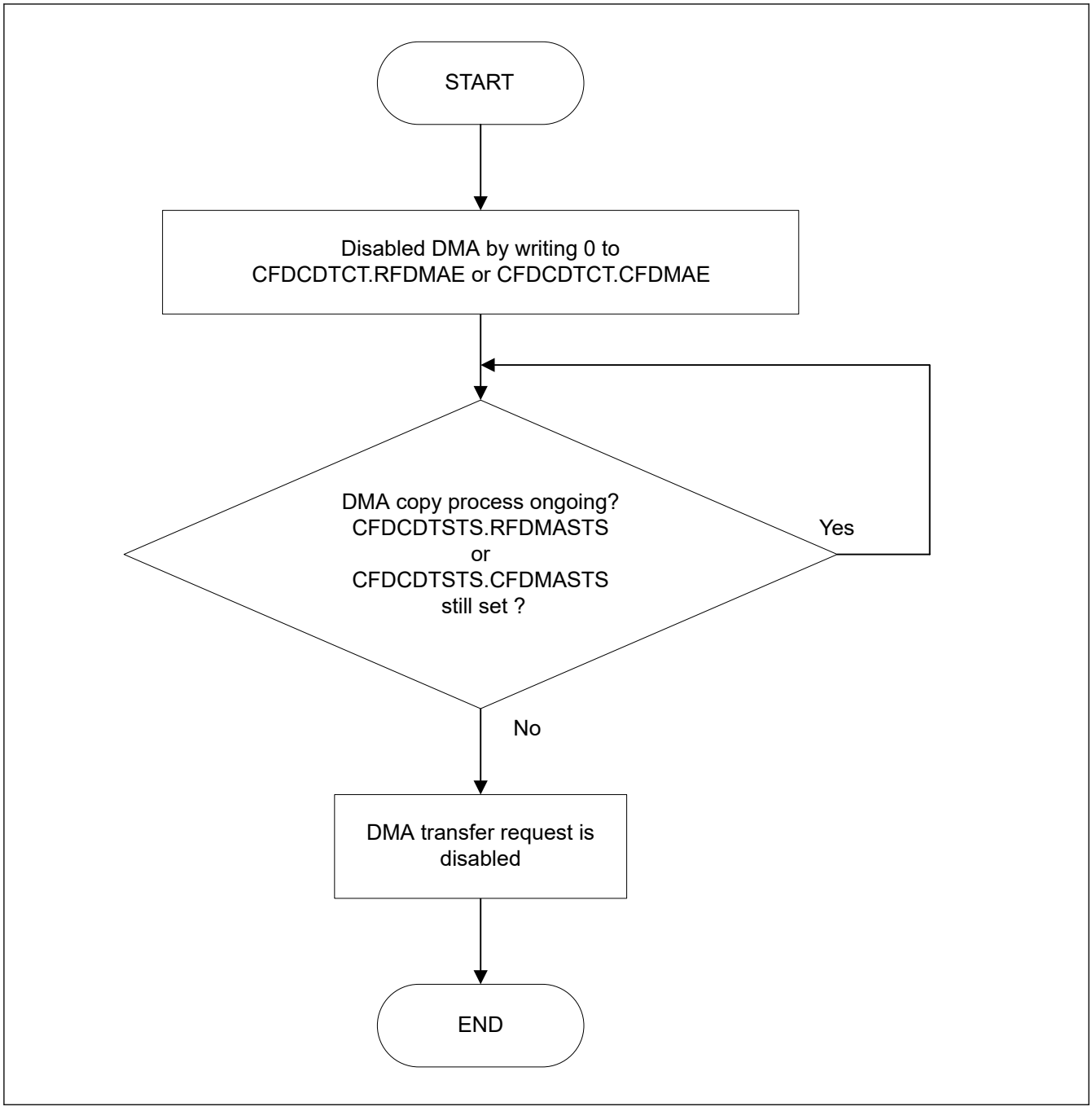


Figure 34.34 DMA disable flow

34.8 Reception and Transmission

34.8.1 Reception

In the CANFD module, CAN messages received on any of the channels are stored in RX message buffers, RX FIFO buffers, or Common FIFO buffers configured in RX mode depending on the Acceptance Filter List entries.

- Up to 16 RX message buffers can be configured
- 2 RX FIFO buffers available
- 1 Common FIFO Buffer can be configured in RX mode

34.8.1.1 Message Storage in RX Message Buffers

When a message is successfully received and stored in a RX message buffer, the corresponding New Data flag is set in the RX Message Buffer New Data Register.

The CAN message can be read from the corresponding RX message buffer.

If a new message is stored into a RX message buffer before the previous message in this message buffer can be read, then the original message is overwritten. There is no mechanism for preventing a new message from overwriting the current message in the RX message buffer. If such a loss of messages is not acceptable, then RX FIFO should be used for storing related messages.

Note: Users should do the same processing as the existing software flow also when using interrupt. (see Figure 34.36)

Note: Unused data bytes are filled with 0x00 depending on the DLC value.

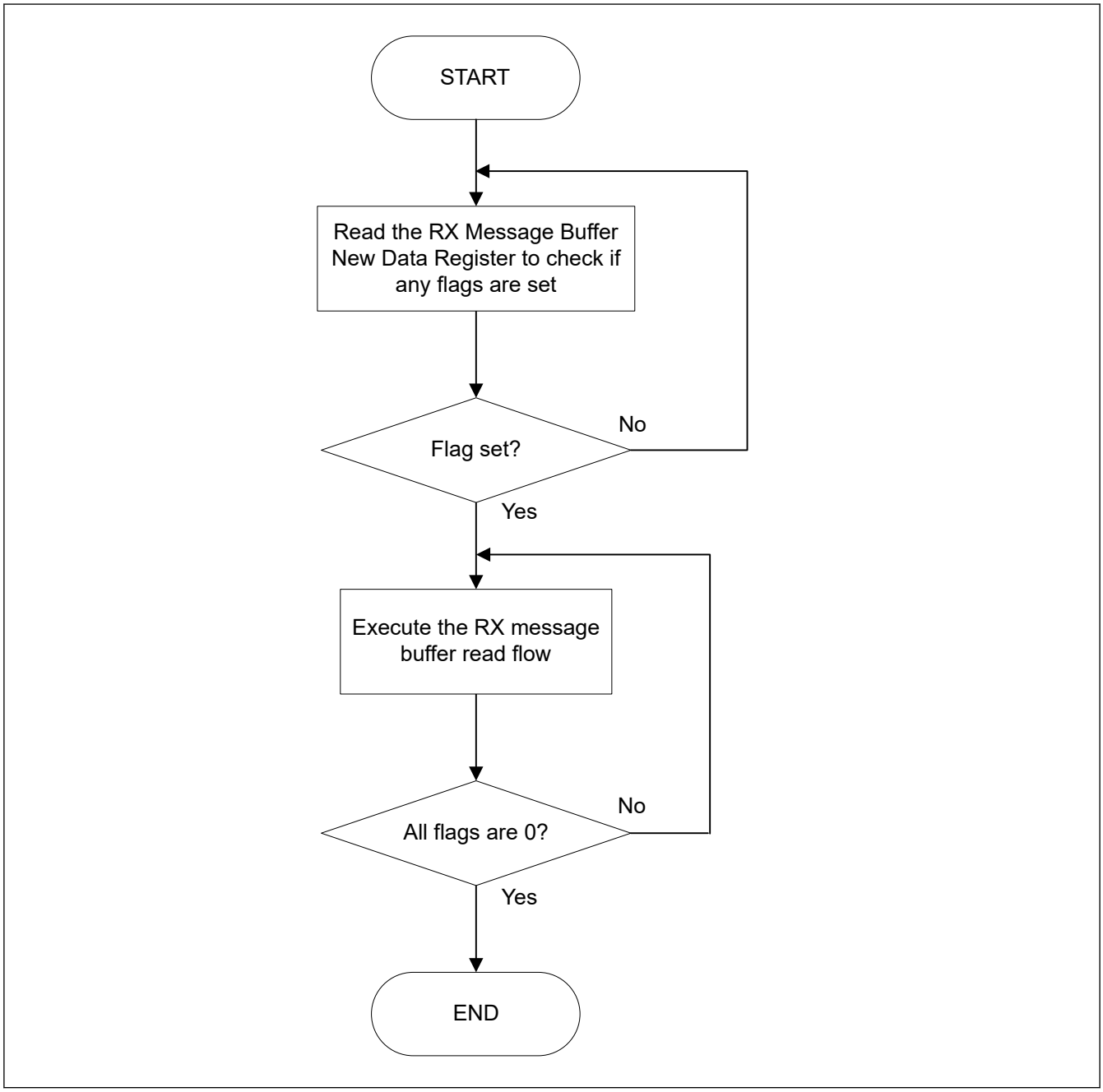


Figure 34.35 Access flow of RX message buffer (Polling)

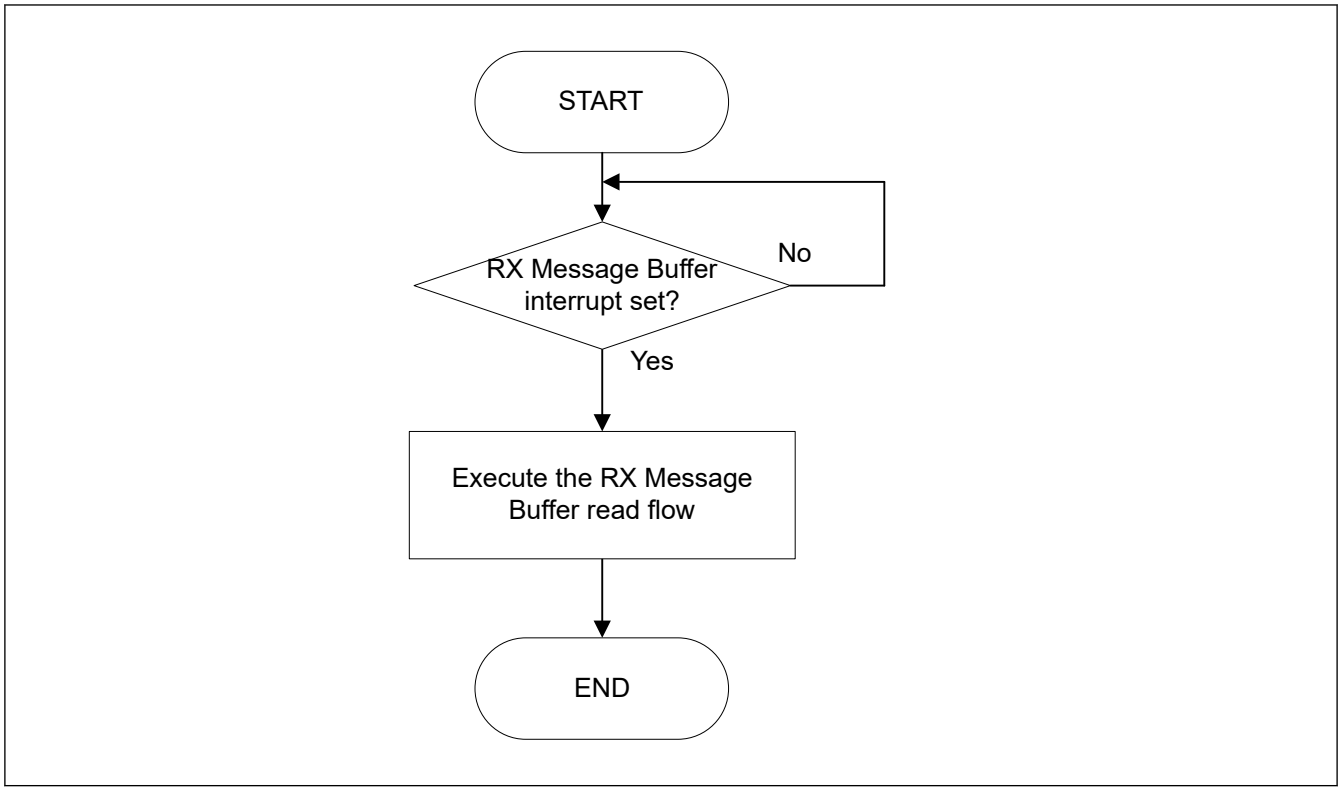


Figure 34.36 RX Message Buffer Message Access Flow (interrupt)

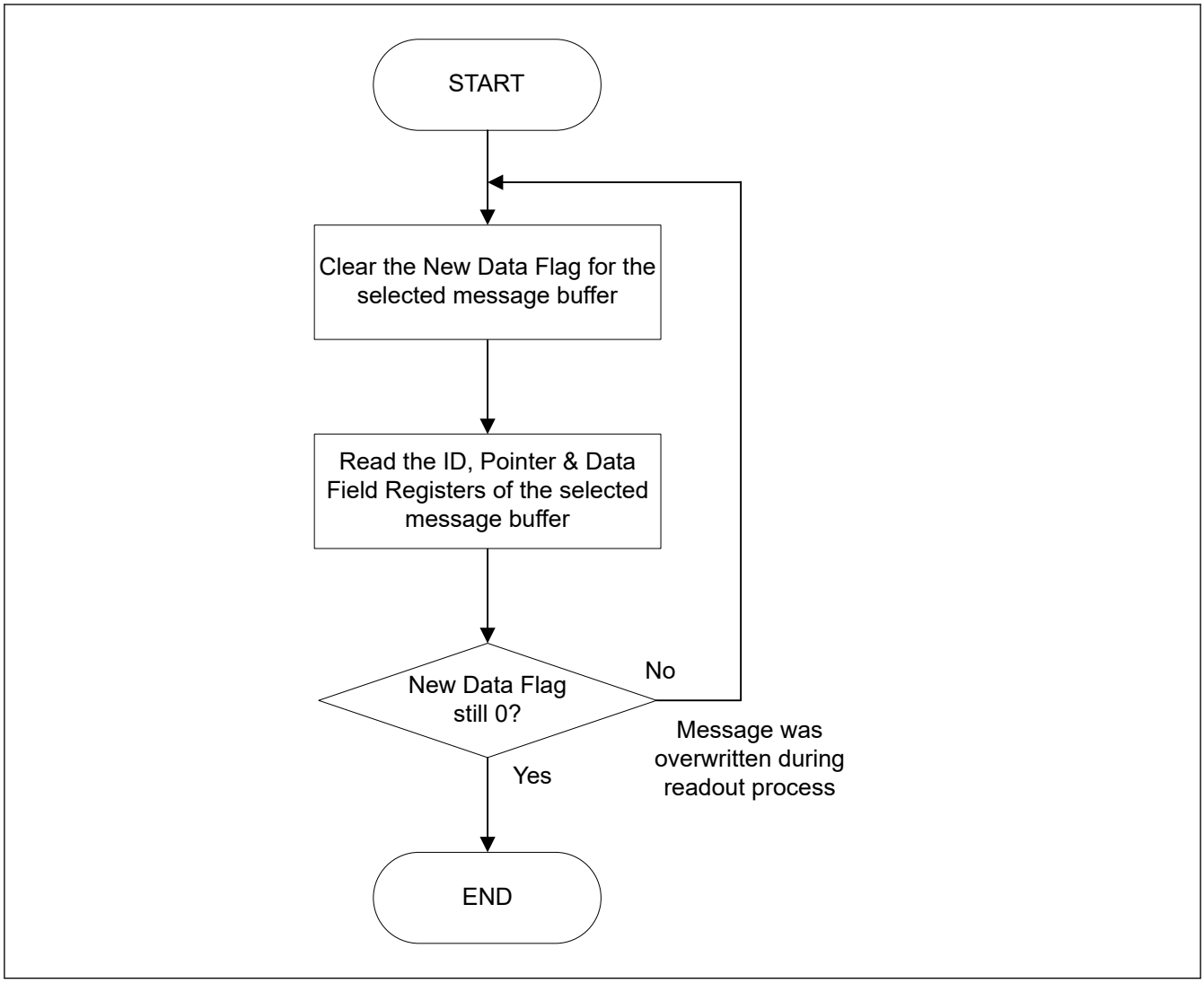


Figure 34.37 Read flow of RX message buffer

34.8.1.2 Message Storage in FIFO Buffers

The AFL entries for routing the received messages to RX FIFO buffers or Common FIFO buffer configured in RX mode should be configured based on system requirements.

The CFDGAFLP1r.GAFLFDP[8,1:0] field in the matching AFL entry selects the FIFO buffers to which the related reception message is stored.

When the received message is stored in one or more RX FIFO buffers or Common FIFO buffer configured in RX mode, the message counter value is incremented in the corresponding RX FIFO Status Registers or Common FIFO Status Register.

Depending on the configuration of the FIFO buffers, an interrupt might also be generated.

The message can be read from the corresponding FIFO Access registers.

Note: Because many messages can be stored in the FIFO buffers, reading more than one message may be required to read the latest message stored in a FIFO buffer.

If the message count value matches the FIFO depth, the FIFO Full flag is set.

When the value 0xFF is written to the corresponding FIFO Pointer Control Register, the message count is decremented by 1.

Only write 0xFF to the FIFO Pointer Control register after reading the complete message from the FIFO Access registers of the corresponding FIFO.

When all the messages stored in the FIFO are read, the FIFO Empty flag is set.

If a new message is stored into the FIFO when the FIFO message count matches the FIFO depth (FIFO full condition), the FIFO Message Lost flag is set and the new message is lost (no overwrite of already stored messages takes place).

An appropriate value can be configured as warning level to generate an interrupt before the FIFO full condition occurs to avoid loss of a message due to an overrun condition.

Note: The message lost can be set only in RX mode by CAN, and the flag is not set when the CPU is overloading the FIFO buffers.

The RX FIFO buffers and the Common FIFO buffers configured in RX mode can be disabled at any time by clearing the CFDRFCCa.RFE or CFDCFCC.CFE bit in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers.

When the CFDRFCCa.RFE or CFDCFCC.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Therefore, all messages in the FIFO buffers are lost and no further messages can be stored into the FIFO.

When the RX FIFO buffers or Common FIFO buffer configured in RX mode is assigned as a DMA channel, software should not access the FIFO Access Register of this FIFO buffer or write 0xFF to the FIFO Pointer Control Register (CFDCFPCTR.CFPC or CFDRFPCTRa.RFPC). This can lead to unintended FIFO message decrement. The DMA channel controls the FIFO decrement automatically.

Note: If the interrupt flag is set for a FIFO buffer and then the FIFO is disabled, the interrupt flag is not cleared automatically. The interrupt flag should be cleared before disabling the FIFO.

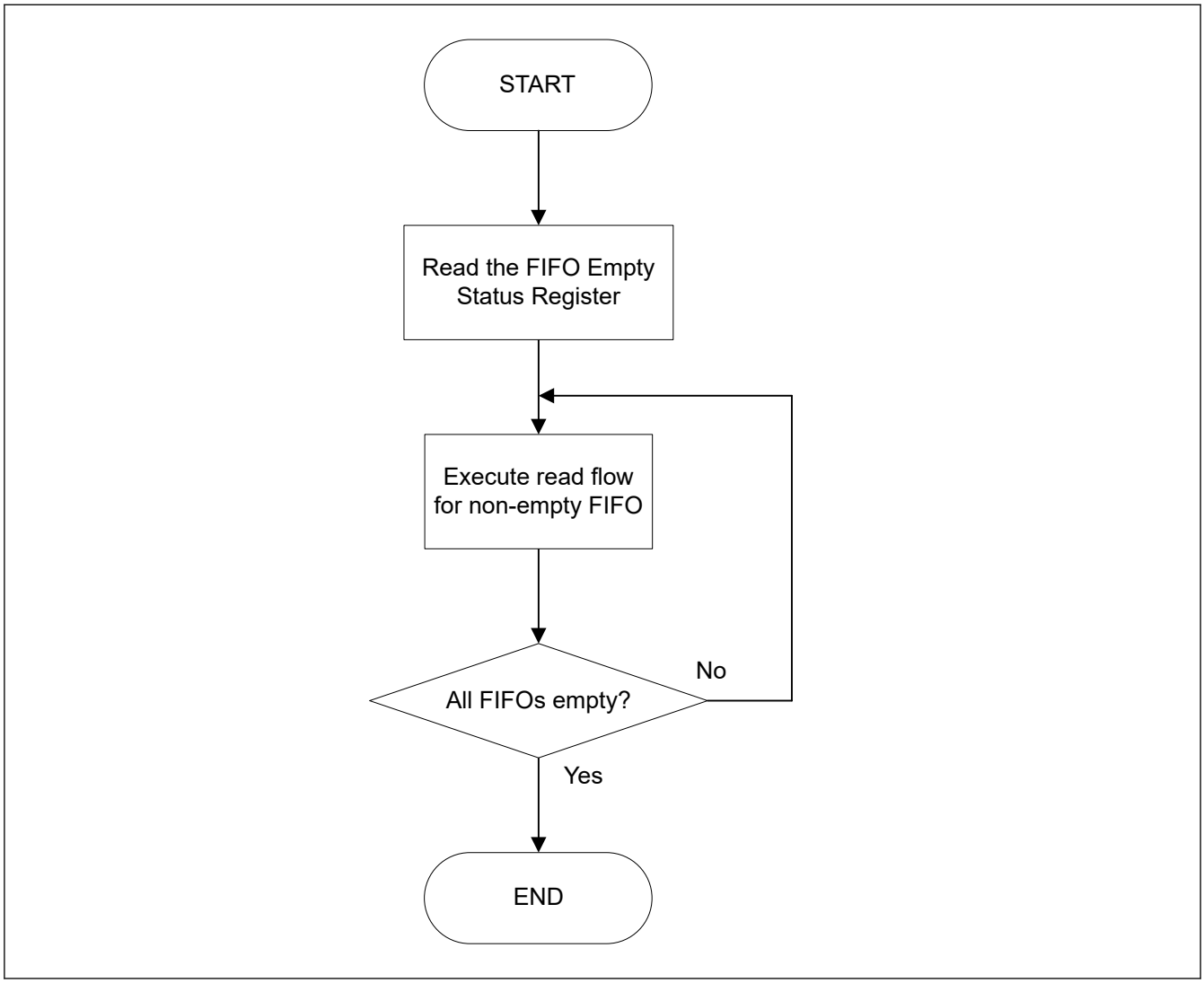


Figure 34.38 Access flow of FIFO buffer message (example for polling case)

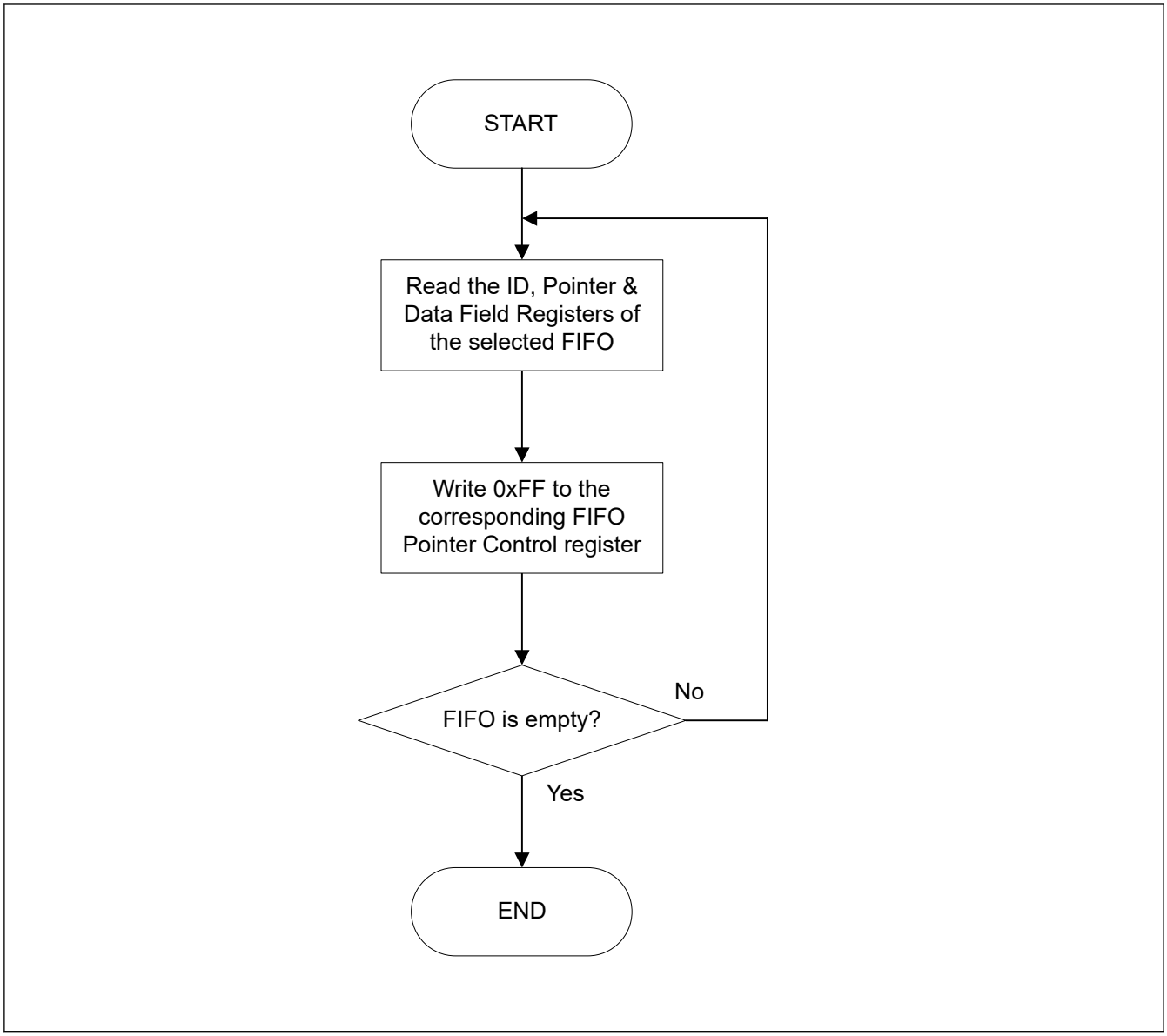


Figure 34.39 Read flow of RX FIFO buffer message (example for polling case)

Note: When the next frame is received before clearing the completion interrupt flag of reception, the completion interrupt of reception is not set again.

Even when an interruption flag is cleared after the completion processing of reception, the already received interrupt flag is not set.

It is necessary to perform the completion processing of reception even before the next completion of frame reception, and to clear an interruption flag.

When processing does not meet the condition, after checking that receiving data is empty, interrupt flag is cleared and it checks that receiving data is empty again.

34.8.1.3 Timestamp

The timestamp counter is a free-running counter that can be used to check reception time of an incoming message or transmission time of successful transmitted messages. The Timestamp counter value is captured based on the CFDGFDCFG.TSCCFG[1:0] configuration (at the sample point of start of frame, point in time when the frame is valid, or for CANFD frames also at the sample point of the RES bit). For reception, it is stored together with the message ID and data into the target RX message buffer or RX FIFO.

For transmit message, the timestamp counter value is stored as part of the TX History List entry.

The counter can be clocked with the peripheral clock or with the CAN channel bit timing clock. The counter source clock can be configured with the CFDGCFG.TSSS bit of the Global Configuration Register. If this bit is 0, the peripheral clock is used. If the bit is 1, the selected CAN channel bit time clock is used.

The channel selection is performed with the CFDGCFG.TSBTCS bit of the Global Configuration Register.

Care must be taken when using selected CAN channel bit time clock as the clock source. When entering Channel Halt mode or Channel Reset node, for this channel, the timestamp counter is stopped. For other CAN channels, the timestamp counter value is not updated.

If peripheral clock is selected as the timestamp counter clock source, Channel modes do not affect the timestamp counter function.

The source clock for the timestamp counter can be divided by a factor defined by the CFDGCFG.TSP bits (timestamp prescaler) in the Global Configuration Register.

The timestamp counter can be reset to 0x0000 with the CFDGCTR.TSRST bit (timestamp reset).

34.8.2 Transmission

There are several possible transmission configurations:

- Normal transmission
- FIFO transmission
- TX Queue transmission
  A fixed number of transmission message buffers (4 TX message buffers) are dedicated. These message buffers are only used for transmission and cannot be configured for reception.
  Additionally transmission from TX Queue or Common FIFO in TX mode can be configured in the following way (see Figure 34.40):
  – TX Queue: Up to four transmission message buffers can be grouped to form a TX Queue with a common access window.
    Upper transmission message buffers are used to form the TXQ.
    TXQ has an access window.
    • TXQ is transmission Message Buffer 0.
- Common FIFO (TX mode): Common FIFO in TX mode is linked to a dedicated channel.
  Channel has a fixed number of one Common FIFO assigned to it. Within the channel, a Common FIFO configured in TX mode, can be freely linked (assigned) between 0 and 3 transmission message buffers (only one FIFO to one transmission message buffer).
  The Common FIFO buffer then replaces the transmission message buffer linked to it.
  Transmission Control and Status registers of these transmission message buffers should not be used.

See Figure 34.28 for information about Common FIFO buffer assignment to related channel.

Note: Common FIFO buffer should not be linked to TX message buffers that are already part of a TX Queue.

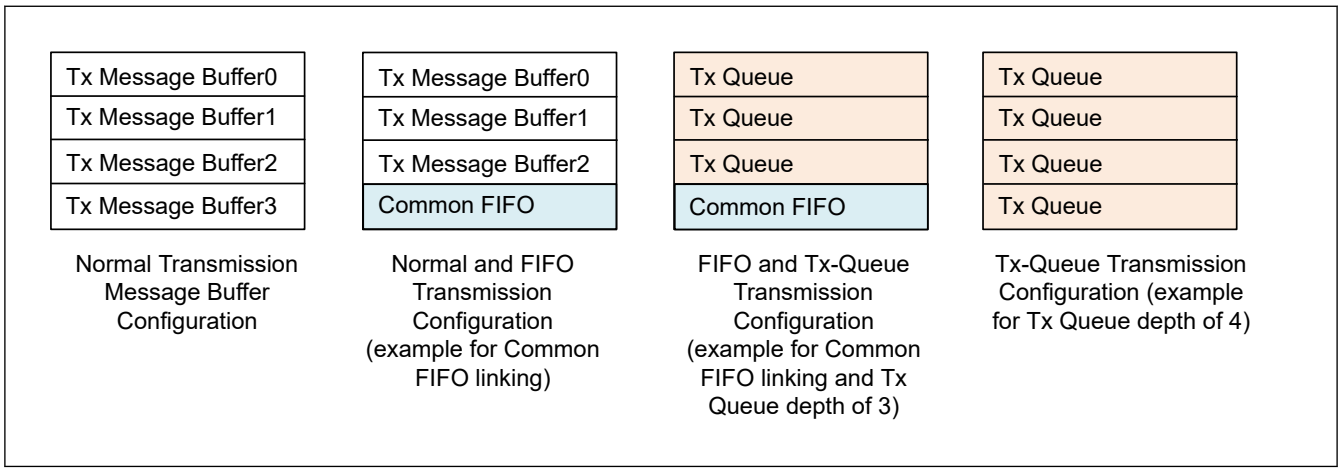


Figure 34.40 Configuration of channel transmission message buffer

34.8.2.1 Transmission Priority

If two or more transmission message buffers of a channel are configured for transmission, then the transmission priority in the CANFD module can be selected from the following two modes:

- CAN ID priority
- Message buffer number priority.

The transmission priority mode is common for all message buffers. It can be configured with the CFDGCFG.TPRI bit in the Global Configuration Register.

For message buffer number priority transmission, the smallest message buffer number with transmission request has the highest priority for transmission. This also includes the TX message buffers linked to the Common FIFO buffer configured in TX mode.

However, message buffer number priority should not be used if TX Queue is enabled.

For CAN ID priority transmission, ID priority complies with the CAN bus arbitration rule (as specified in ISO 11898-1 specification). All TX message buffers can enter the ID priority comparison for message buffers configured for transmission. This also includes the TX message buffers linked to the Common FIFO buffer configured in TX mode and includes the TX Queue message buffers.

If the ID of two or more message buffers is the same, then the smaller message buffer number has higher priority for transmission.

Note: For Common FIFO buffer configured in TX mode, only the message currently being pointed to by the FIFO read pointer can be included in the transmission arbitration.

If the message is being transmitted from the FIFO, then the next pending message within the same FIFO is considered in the transmission arbitration.

In contrast to this, all transmission message buffers of a TX Queue participate in internal transmission arbitration.

Figure 34.41 shows the transmission configuration flow.

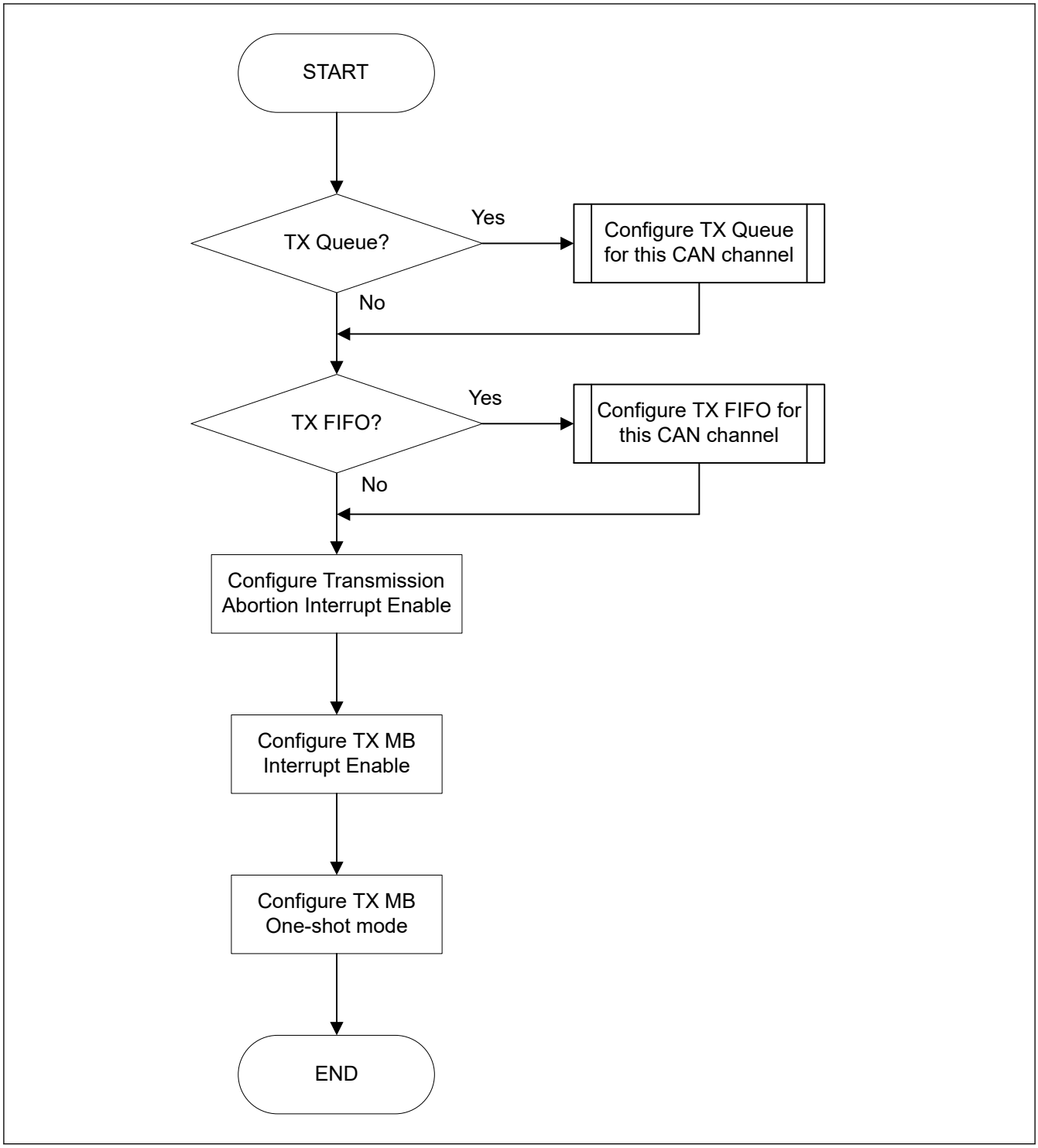


Figure 34.41 Flow for transmission configuration

34.8.2.2 Normal Transmission

Each transmission message buffer has two modes of message transmission:

1. Regular transmission mode
   If the message buffer is placed in regular transmission mode, the data frame or remote frame set in that message buffer can be transmitted.
   Completion of regular transmission can be checked through the related TX Message Buffer Transmission Result flag bits (CFDTMSTSj.TMTRF) in the TX Message Buffer Status Registers. These bits are set to 10b or 11b when the regular transmission is successful.

When arbitration is lost or an error occurs, message transmission is further attempted if no transmission abort request is set for this transmission message buffer.
New internal transmission arbitration for this channel is performed for all message buffers with transmission request.

2. One-shot transmission mode

When the CFDTMCi.TMOM bit of the TX Message Buffer Control Registers is set for a transmission message buffer, the message buffer is placed in One-shot transmission mode and attempts to transmit a message only once.

Completion of One shot transmission can be checked through the related TX Message Buffer Transmission Result Flag bits (CFDTMSTSj.TMTRF) in the TX Message Buffer Status Registers. The CFDTMSTSj.TMTRF bits are set to 10b or 11b when One-shot transmission is successful.

The CFDTMSTSj.TMTRF bits are set to 01b when arbitration is lost or an error occurs during transmission of the related message buffer.

Additional message transmission is not attempted in this case.

The regular transmission request procedure after a configuration is shown in Figure 34.42.

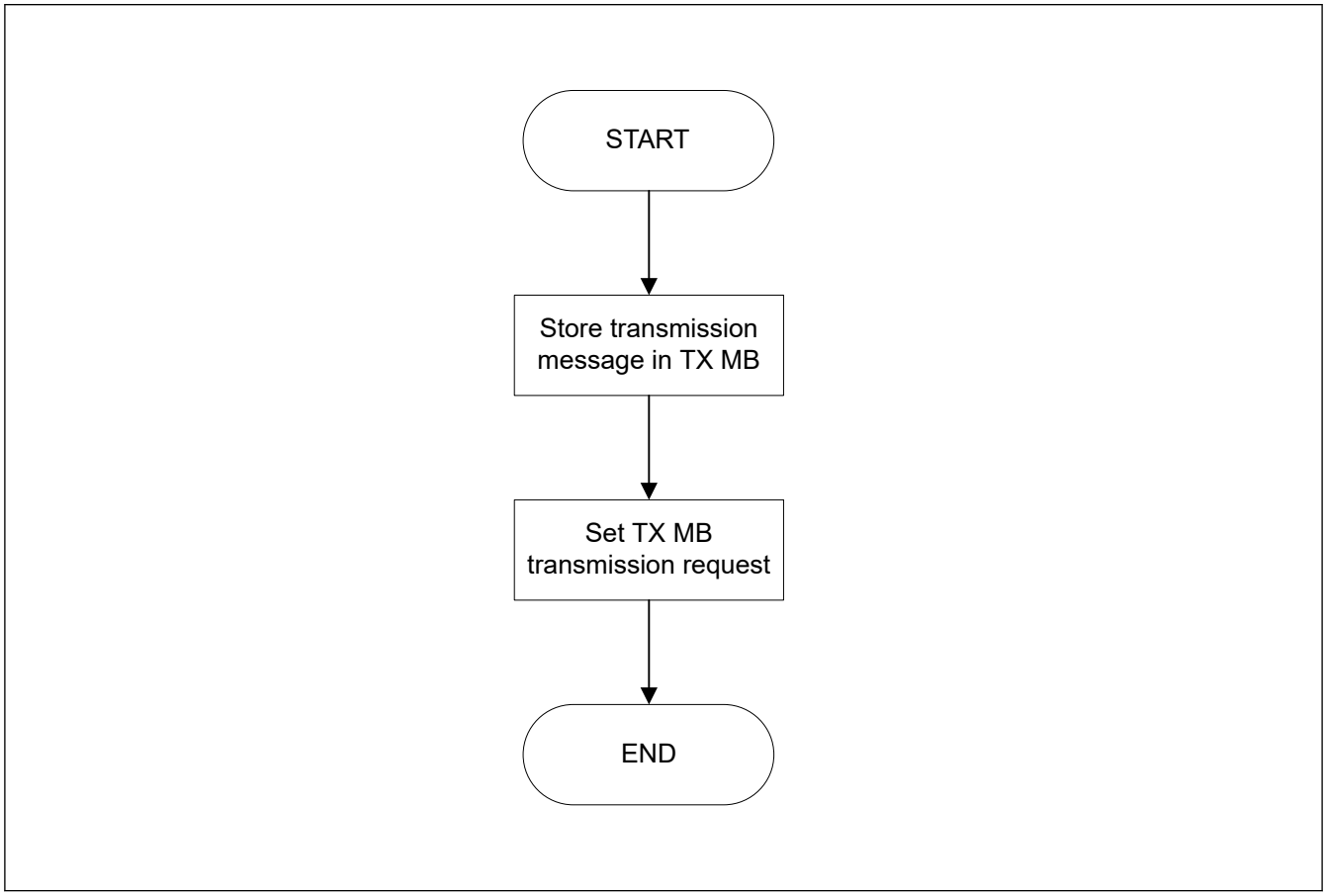


Figure 34.42 Transmission request procedure using normal TX Message Buffer mode

(1) Setting for TX Message Buffer Control Register

Table 34.26 shows configuration of a normal CAN transmission mode.

Table 34.26 Configuration of CAN transmission mode (1 of 2)

Transmission request CFDTMCi.TMTR	Transmission abort request CFDTMCi.TMTAR	One-shot enable CFDTMCi.TMOM	Communication activity
0	0	0	Message buffer disabled
0	0	1	Message buffer disabled

Table 34.26 Configuration of CAN transmission mode (2 of 2)

Transmission request CFDTMCi.TMTR	Transmission abort request CFDTMCi.TMTAR	One-shot enable CFDTMCi.TMOM	Communication activity
1	0	0	Configured as a transmission message buffer for a data frame or a remote frame
1	0	1	Configured as a one-shot transmission message buffer for a data frame or a remote frame
1	1	0	Transmission abort requested
1	1	1	One-shot transmission abort requested

The configuration bits can be configured in the TX Message Buffer Control Registers.

Figure 34.43 shows timings for successful transmission for two message buffers.

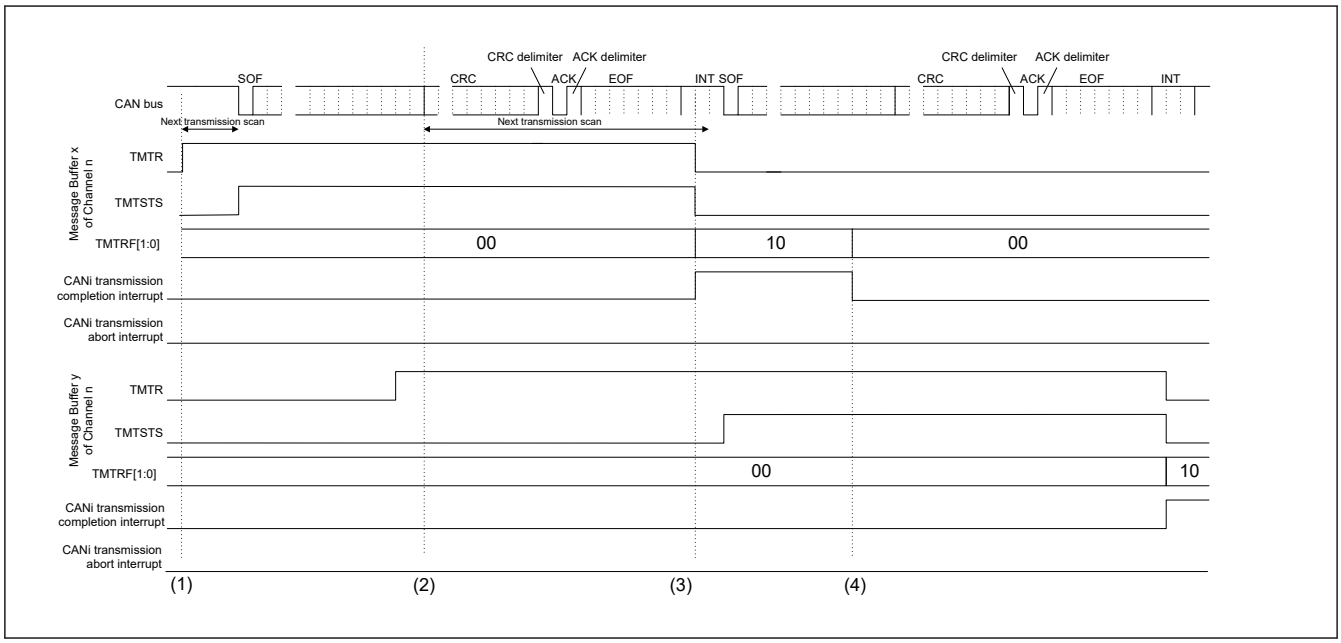


Figure 34.43 Timing of request and flag bits for successful transmission

1. If the CFDTMCi.TMTR bit in the TX Message Buffer Control Registers is set in the bus idle state, the message buffer scanning procedure determines the highest priority message buffer for transmission.
   When the transmission message buffer is determined, the CFDTMSTSj.TMTSTS bit in the related TX Message Buffer Status Registers is set (transmitting/transmitter), and CAN channel starts the transmission *1.
2. At the first bit of CRC, the transmission scanning procedure starts for the next possible transmission when pending transmission requests exist.
3. If the message has been successfully transmitted, the CFDTMSTSj.TMTRF[1:0] bits in the corresponding TX Message Buffer Status Registers are set to 10b and CFDTMSTSj.TMTSTS and the CFDTMCi.TMTR bits are cleared.
   When the TMIE bit in the TX Message Buffer Interrupt Enable Configuration Registers is set (interrupt enabled), the CAN successful transmission interrupt request is generated.
   To clear the related interrupt line, clear the CFDTMSTSj.TMTRF flag bits.
4. Before starting the next transmission, clear the CFDTMSTSj.TMTRF bits. Load the next message in the transmission message buffer and set the CFDTMCi.TMTR bit again.
   CFDTMCi.TMTR bit cannot be set again before CFDTMSTSj.TMTRF[1:0] bits are cleared.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the CFDTMSTSj.TMTSTS bit is cleared.

The transmission scanning procedure is performed again to search for the highest priority transmission message buffer from the beginning of the first CRC bit.

If an error occurs either during transmission or following the loss of arbitration, then during the error frame, the transmission scanning procedure is performed again to search for the highest priority transmission message buffer.

Note: The setting point of CFDTMSTSj.TMTSTS is not always fixed at the start of the SOF. It may be delayed up to the start of the standard ID due to the synchronization logic implemented for the PLL bypass.

Figure 34.44 shows timings for transmission abort for two message buffers.

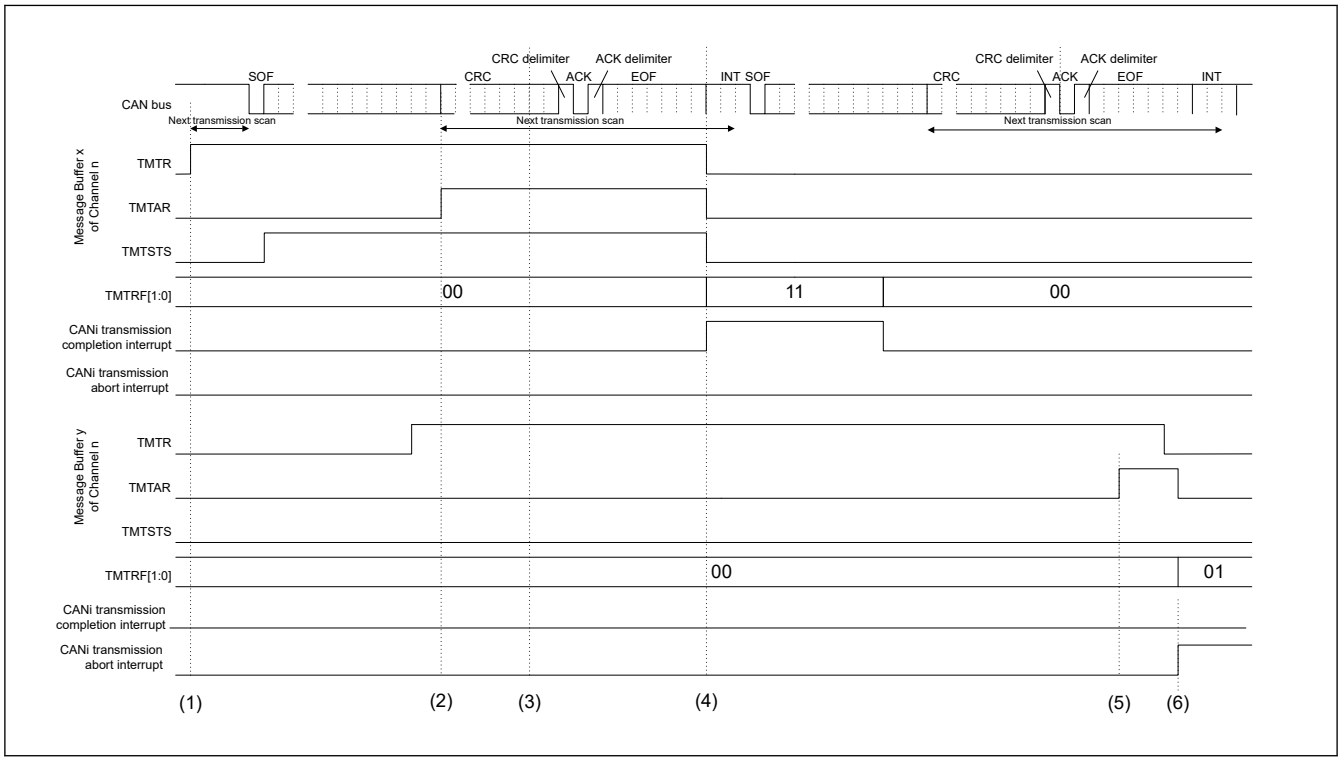


Figure 34.44 Timing of request and flag bits for transmission abort

1. If the CFDTMCi.TMTR bit in the TX Message Buffer Control Registers is set in the bus idle state, the message buffer scanning procedure determines the highest priority message buffer for transmission.
   When the transmission message buffer is determined, the CFDTMSTSj.TMTSTS bit in the TX Message Buffer Status Registers is set (transmitting/transmitter), and CAN channel starts the transmission*1.
2. If the CFDTMCi.TMTAR bit is set when the related message buffer is already selected for transmission or currently transmitting, the message is not aborted, if no error occurs or arbitration is lost.
3. At the first CRC bit, the transmission scanning procedure starts for the next transmission. In this example, timing chart message buffer y is not selected as the next transmission message buffer.
4. If the message has been successfully transmitted, the CFDTMSTSj.TMTRF[1:0] bits in the corresponding TX Message Buffer Status Registers are set to 11b and the CFDTMSTSj.TMTSTS and CFDTMCi.TMTR bits are cleared.
   When the TMIE bit in the TX Message Buffer Interrupt Enable Configuration Registers is set (interrupt enabled), the CAN successful transmission interrupt request is generated.
   To clear the related interrupt line, clear the CFDTMSTSj.TMTRF[1:0] bits.
5. Another CAN node is transmitting on the CAN bus (CFDTMSTSj.TMTSTS is not set). If the CFDTMCi.TMTAR bit is set when the related channel is under transmission scan, the transmission request cannot be cleared.
6. After internal processing time, the transmission is aborted and the CFDTMSTSj.TMTRF[1:0] bits are set to 01b.
   If the message buffer is not transmitting or selected as the next transmission message buffer or under transmit scan, then the abort is immediately accepted and the corresponding CFDTMSTSj.TMTRF[1:0] bits in the TX Message Buffer Status Registers are set to 01b.
   In addition, CFDTMCi.TMTR, and CFDTMCi.TMTAR bits are cleared automatically.
   When the transmission abort interrupt enable TAIE bit of the related Channel Control Register is set then an interrupt is generated for successful transmission abort.
   To clear the related interrupt line the CFDTMSTSj.TMTRF[1:0] bits have to be cleared.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the CFDTMSTSj.TMTSTS bit is cleared.

The transmission scanning procedure is performed again to search for the highest priority transmission message buffer from the beginning of the first CRC bit.

If an error occurs, either during transmission, or following the loss of arbitration, then during the error frame, the transmission scanning procedure is performed again to search for the highest priority transmission message buffer.

34.8.2.3 TX FIFO Transmission

One common FIFO buffer is assigned to CANFD module. The FIFO buffer can be linked to any normal TX message buffer position for this channel with the CFDCFCC.CFTML bits in the Common FIFO Configuration/Control Register if configured in TX mode.

When the transmission scan starts and the FIFO buffer corresponding to this TX message buffer is enabled, the relevant message in the FIFO buffer participates in the transmission scan.

Configuration of a TX message buffer linked to a FIFO buffer configured in TX mode should not be done.

(1) TX FIFO Operation

CAN messages can be written into the TX FIFO by writing to the corresponding FIFO Access registers.

When the value 0xFF is written into the corresponding FIFO Pointer Control Register, the message count of the related FIFO is incremented by 1.

Only write to the FIFO Pointer Control register after writing the complete message to the corresponding FIFO Access registers. If the message count matches the FIFO depth, the FIFO Full flag is set.

The oldest message in the TX FIFO is included in the scan for transmission by the corresponding CANFD module channel logic.

When a message is successfully transmitted from the TX FIFO, the message count value is decremented by 1. When all the messages from the FIFO are transmitted, the FIFO Empty flag is set.

The interrupt generation conditions for the TX FIFO buffer can be configured by configuring the CFDCFCC.CFIM bit in the corresponding Common FIFO Configuration/Control Register.

If CFDCFCC.CFIM bit is 0, then interrupt is generated when the last message is successfully transmitted from the TX FIFO buffer.

If CFDCFCC.CFIM bit is 1, then interrupt is generated for every successfully transmitted message from the TX FIFO buffer.

The Common FIFO can set interrupt when CAN frame transmission is complete.

The Common FIFO buffer configured in TX Mode can be disabled by clearing the CFDCFCC.CFE bit in the Common FIFO Configuration/Control Register. If this bit is cleared to 0, the FIFO Empty flag is set as follows:

- Immediately if the message from the TX FIFO is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the TX FIFO is already scheduled for transmission or already in transmission.

Note: The Common FIFO buffer is considered as disabled after clearing the CFDCFCC.CFE bit only when the Empty flag is set for the corresponding Common FIFO buffer.

Other possible messages pending from the TX FIFO are lost and their transmission must be requested again. Before CFDCFCC.CFE is set again, ensure that CFDCFSTS.CFEMP bit is set and that there are no pending abort from the TX FIFO.

When the CFDCFCC.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Therefore, all messages in the FIFO buffers are lost and no further message can be stored into the FIFO.

The FIFO transmission request procedure after configuration is shown in Figure 34.45.

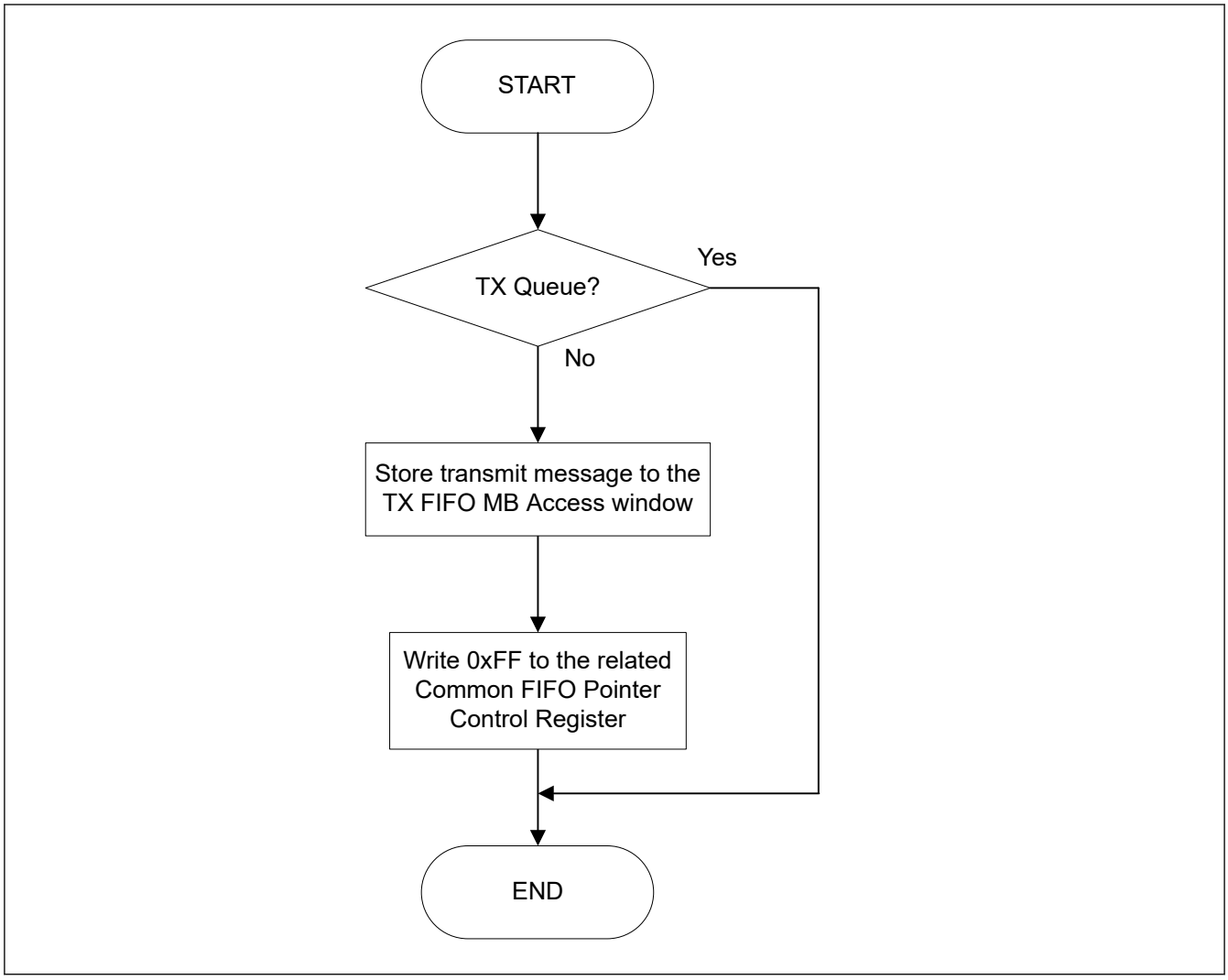


Figure 34.45 Request procedure for TX FIFO transmission

(2) Interval Timer for FIFO Transmission

For each Common FIFO in TX mode, it is possible to specify a delay between two consecutive messages that are configured for transmission from the same FIFO buffer. This delay is called interval time. This interval time starts after the first message has been successfully transmitted from the FIFO buffer after the CFDCFCC.CFE bit is set.

When the Common FIFO in TX mode is enabled, the first message is transmitted without considering this interval time.

The interval timer stops counting when:

- FIFO is disabled by clearing the CFDCFCC.CFE bit.
- CAN channel is in CH_RESET mode.

The interval time is specified by the CFDCFCC.CFITT value from 0 to 255 timer units in the Common FIFO Configuration/ Control Register.

The timer unit can be defined based on two different source clocks for the interval timer. To disable the interval timer for FIFO transmission, select a value of 0.

The timer source can be selected with the configuration bit CFITSS in the Common FIFO Configuration/Control Register.

If CAN channel bit time clock is configured as the clock source, and the CAN channel enters CH_HALT, CH_RESET, or CH_SLEEP mode, the interval timer is stopped for that channel.

If peripheral clock is selected as the interval timer clock source, the interval timer is stopped only when the CAN channel is in CH_RESET or CH_SLEEP mode.

The reference clock can be used to configure the interval time in fixed time units. It is based on the peripheral clock. The reference clock prescaler value CFDGCFG.ITRCP in the Global Configuration Register defines the relation between the peripheral clock frequency/period and the reference clock period.

See Table 34.27 for CFDGCFG.ITRCP configuration values to achieve different reference clock periods based on the peripheral clock frequency and period.

Table 34.27 Configuration example for the reference clock of the FIFO interval timer

Reference clock/Peripheral clock	1 µs	100 µs	500 µs
16 MHz/62.5 ns	16	1600	8000
20 MHz/50 ns	20	2000	10000
32 MHz/31.25 ns	32	3200	16000
50 MHz/20 ns	50	5000	25000

The reference clock resolution can be specified by the interval timer reference clock resolution value CFDCFCC.CFITR in the Common FIFO Configuration/Control Register.

The interval time is based on the reference clock period multiplied by the configured value (x1 or x10). The reference clock based interval timer can be used to satisfy the requirements of the ISO 15765-2 Separation Time. The whole range for the separation time from 100 µs to 127 ms can be covered.

The specified interval time starts after successful transmission event (after EOF7 state of the CAN protocol).

When the interval time has elapsed, the next transmission request is raised by the related TX FIFO. Therefore, the interval time defines the minimum time between two messages transmitted from one FIFO.

The next message is sent at earliest after this interval time. Figure 34.46 shows an example timing of the internal processing.

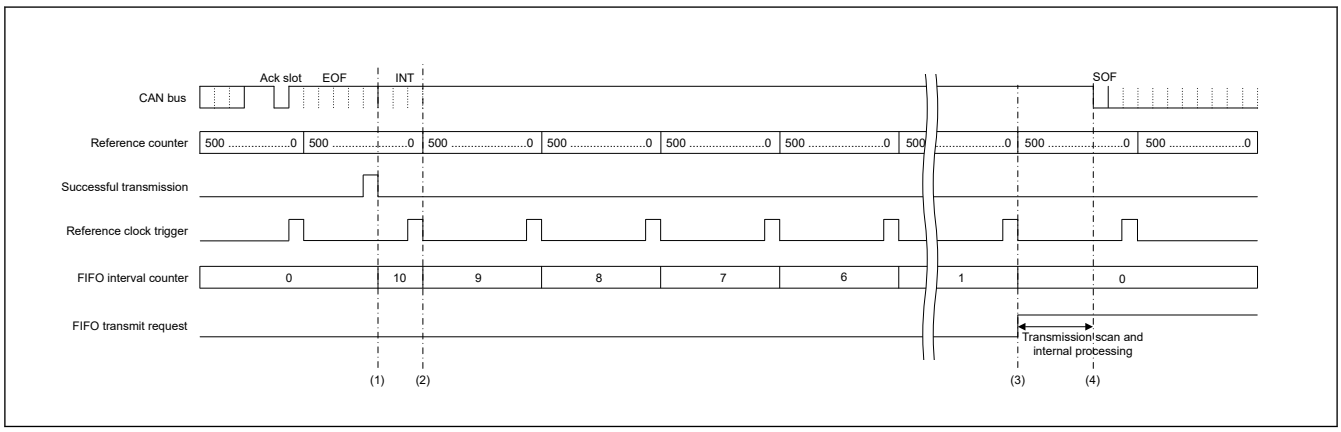


Figure 34.46 Example for interval processing time

The configuration for the timing in Figure 34.46 is as follows:

- Peripheral clock frequency = 50 MHz
- Interval timer reference clock (CFDGCFG.ITRCP) = 500 times
- Reference clock from the settings in Figure 34.46 = 10 µs
- Common FIFO interval timer source selection (CFDCFCC.CFITSS) = 0
- Common FIFO interval timer resolution (CFDCFCC.CFITR) = 0
- Common FIFO interval transmission time (CFDCFCC.CFITT) = 10 times
- Theoretical message separation interval = 100 µs

1. Internal FIFO interval timer is restarted with the occurrence of successful transmission result. This restart is not synchronized to the reference clock trigger. Therefore, the first interval is counting less or equal to 1 reference clock interval.
2. With the next reference clock trigger the FIFO interval timer is decremented.

3. When the FIFO interval timer reached the value 0, the FIFO transmit request is set.
4. When the FIFO is selected for transmission, the transmission starts. Due to internal processing, this usually takes less than 3 CAN bit time, between the internal FIFO transmit request set in step 3. and the actual transmission.

In the worst case when multiple events such as a reception scan, an internal message routing, a transmit scan on all channels occur, it can take up to 120 peripheral clock cycles.

As shown in Figure 34.46, it is not guaranteed that the minimum interval is always equal to the configured value. If a minimum time must never be breached, configure CFDCFCC.CFITT to the required minimum value plus 1.

If additional TX message buffers or TX FIFO are configured for transmission of the same channel, the real delay between two messages transmitted from a TX FIFO can be much longer than specified by the interval time. This is due to higher priority message transmission from these TX message buffers or TX FIFO.

Figure 34.47 shows a block diagram of the FIFO interval time generation circuit.

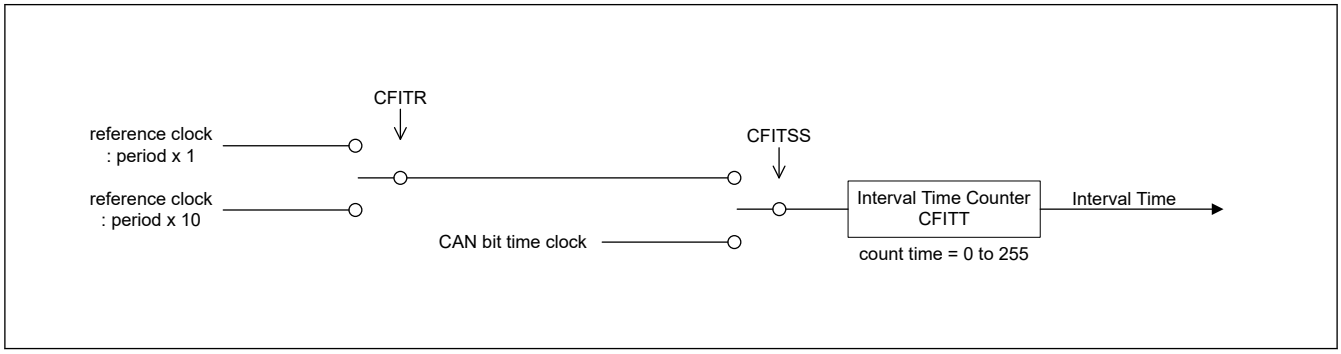


Figure 34.47 Block diagram of FIFO interval timer

34.8.2.4 TX Queue

Each enabled TX Queue for a specific channel consists of 3 to 4 TX message buffers, which are accessed through one access window.

- The first TX Queue can be configured with a depth of three up to four buffers and uses TX Message Buffer No. 0 as access window (referred to as TXQ)

All the TXQ messages enter the priority comparison for the transmission, which should be only ID Priority (CFDGCFG.TPRI = 0).

The registers for TXQ are:

- CFDTXQCC
- CFDTXQSTS
- CFDTXQPCTR

See related access registers TX Message Buffer ID Registers (TMID[m]), TX Message Buffer Pointer Registers (TMPTR[m]), TX Message Buffer Data Field 0 Registers, and TX Message Buffer Data Field 1 Registers (TMDF[0:1][m]) when access window TXQ0 is used.

The depth of each TXQ buffer can be configured by writing to the CFDTXQCC.TXQDC[1:0] bits of the TX Queue Configuration/Control Register. TXQ can be set from TXMB0 to TXMB3 as a queue buffer at the maximum.

The 4 available options for the depth configuration of TXQ buffer are:

- 0x00: TX Queue disabled
- 0x01: reserved
- 0x10: 3 Messages
- 0x11: 4 Messages

Do not access all the TX message buffers forming the TX Queue directly (except TX Message Buffer No. 0, which act as TX Queue access window).

When a system writes in TXQ, it writes in send data, after checking the state of TXQ.

Do not access or configure the related TX Message Buffer Control Registers.

The messages stored to the TX Queue access window are internally stored to a free buffer of the TX Queue.

When the buffer is full, no further access should be done to the queue, until it is no longer full. If access is a software write when the buffer of TXQ is full, send data is overwritten.

The TX Queue can be disabled by clearing the TXQE bit in the TX Queue Configuration/Control Register. If this bit is cleared, the TX Queue Empty flag is set as follows:

- Immediately if the message from the TX Queue is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the TX Queue is already scheduled for transmission or already in transmission.

Note: The TX Queue is disabled only when the Empty flag is set after clearing the TXQE bit for the corresponding TX Queue.

Other possible messages pending from the TX Queue are lost and their transmission must be requested again.

Before TXQE is set again, ensure that the CFDTXQSTS.TXQEMP bit is set and that there is no pending abort from the TX Queue.

When the TXQE bit is cleared, all messages in the TX Queue buffers are lost and no further message should be stored in the TX Queue.

When a message has been stored to the TX Queue, write 0xFF in the TX Queue Pointer Control Register. This sets the transmit request automatically and changes the internal message buffer pointer to the next free message buffer location of the TX Queue.

Note: If two messages with the same ID are stored in the TX Queue, the order of transmission of these messages can be different from the order in which they were stored in the TX Queue.

To avoid this condition, it is important to confirm that the previous message with the same ID was successfully transmitted before a new message with the same ID is stored in the TX Queue.

For the TX Queue, a dedicated interrupt can be enabled by setting the TXQIE bit of the TX Queue Configuration/Control Register.

The interrupt mode can be configured with the CFDTXQCC.TXQIM bit of the same register either to generate an interrupt for every transmitted message or for the last transmitted message.

The TX Queue transmission request procedure after configuration is shown in Figure 34.48.

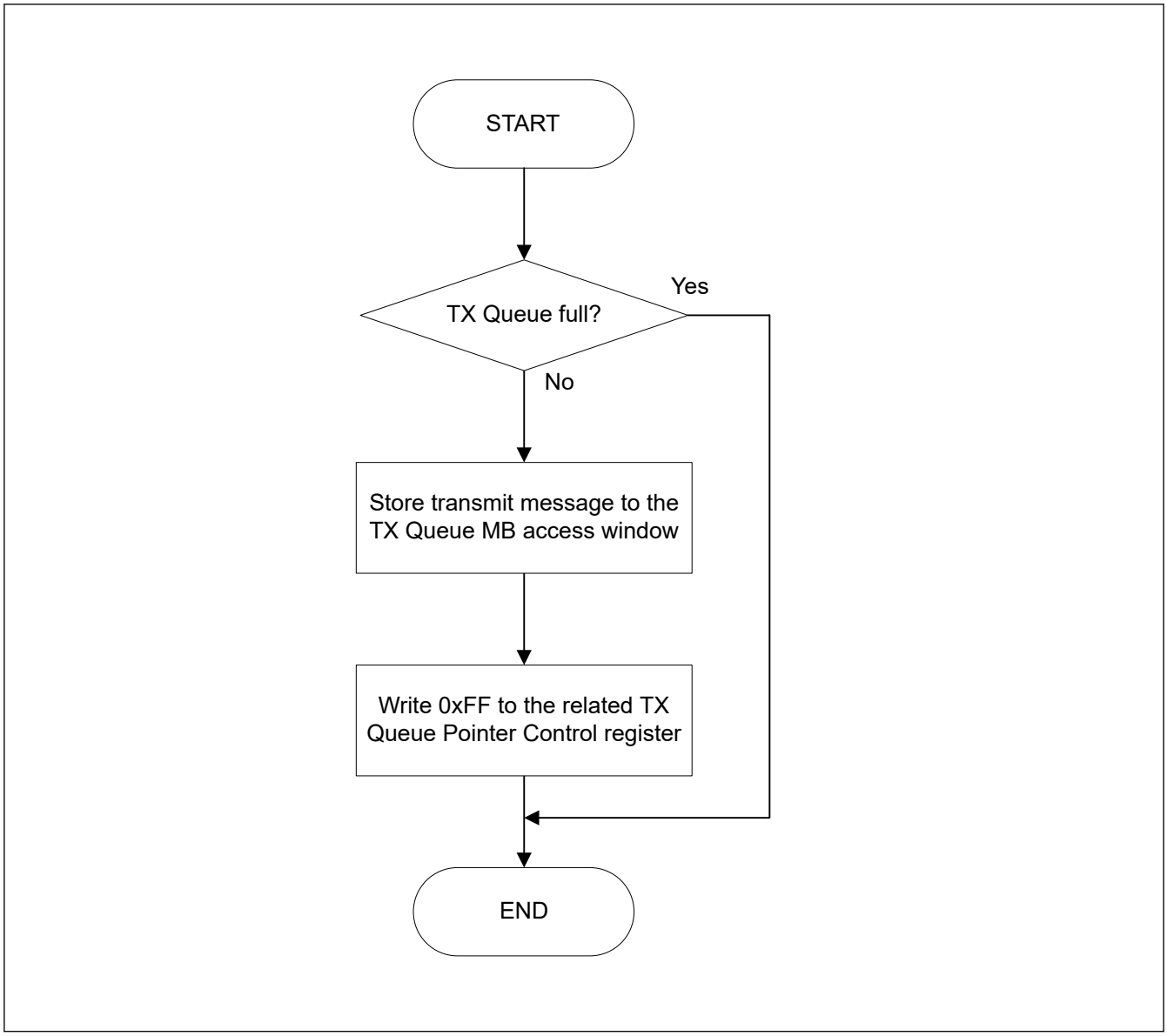


Figure 34.48 TX Queue transmission request

34.8.2.5 TX History List

The TX History List function records the information of the successfully transmitted message in the TX History List Buffers. Two TX History List buffers are provided and THL buffer can store up to 8 TX History List entries.

The CFDTHLCC.THLDTE bit of the TX History List Configuration/Control Register can be used to configure if only message information from TX FIFO or TX Queue is stored, or if all transmit message information from TX Queue, TX FIFO, or normal TX message buffers is stored in the TX History List.

Each transmit message can be individually configured for acceptance to the TX History List with the CFDCFID.THLEN bit in the Message Buffer Pointer Register.

The message information is stored to the TX History List Buffer of a CAN channel after the message is successfully transmitted.

Storing to the list is not synchronized with the status of CFDTMSTSj.TMTRF[1:0] bits in the TX Message Buffer Status Register.

Due to internal processing, the storage to the list can happen with a delay after the successful transmission indication.

Storing the TX History List data can be recognized by the condition that the THLIF is set to 1 when the THLIE bit is configured to 1 or when the TX History List counter CFDTHLSTS.THLMC[5:0] is increased.

In worst case when multi events like reception scan, internal message routing on happen.

- Maximum delay time from setting the CFDTMSTSj.TMTRF to store the TX History List data is 70 peripheral bus clock cycles.

The History list records the following information of a transmitted message:

- Buffer type:
  - 001: TX Message Buffer
  - 010: TX FIFO
  - 100: TX Queue
- Buffer number:
  TX message buffer, TX Queue message buffer or TX message buffer link for the Common FIFO buffer from which transmission occurred. The number depends on the buffer type. See Table 34.28.
- Transmission ID:
  Transmission pointer stored in the transmission message
- Transmit timestamp:
  Message timestamp captured at capture point as configured by CFDGFDCFG.TSCCFG.
- Transmission information label:
  Transmission information label stored in the transmission message.

Table 34.28 TX History List Buffer number entry

Buffer Number	BT[2:0] Buffer Type		
	001b TX Message Buffer	101b TX FIFO	100b TX Queue
00b	Message Buffer 0	Number shown corresponds to the common FIFO. TX Message Buffer Link CFTML of the related Common FIFO configuration	Number shown corresponds to the Message Buffer belonging to the TX Queue which the frame was transmitted
01b	Message Buffer 1		
10b	Message Buffer 2		
11b	Message Buffer 3		

The Transmission ID entry is used to identify which message of a TX FIFO or TX Queue has been successfully transmitted because the TX FIFO or TX Queue number alone is not sufficient.

Therefore, a unique number can be attached to each transmission message stored in a TX FIFO or TX Queue. This unique identification number should be written to the CFDCFFDCSTS.CFPTR[15:0] part of the Common FIFO Access Pointer Register for a TX FIFO or to the CFDTMFDCTRb.TMPTR[15:0] part of the TX Message Buffer Pointer Register of the TX Queue access window message buffer.

When the message is successfully transmitted, this identification number is stored together with the other message related information to the TX History List and can be read using the Transmission ID (TID) of the TX History List Access Register.

Also, for normal TX message buffers, the CFDTMFDCTRb.TMPTR[15:0] part of the TX Message Buffer Pointer Register is stored in the Transmission History List and the information label is the same.

Figure 34.49 shows a transmission preparation flow when TX History List is used.

Read access to the TX History List Access Register is done for every single entry.

After reading one entry, 0xFF must be written to the corresponding TX History List Pointer Control Register to be able to access the next entry until TX History List is empty.

Figure 34.50 shows an example flow for processing the TX History List information.

The TX History Lists have dedicated interrupts, which can be configured with the CFDTHLCC.THLIM bit of the corresponding TX History List Configuration/Control Register and enabled with the CFDTHLCC.THLIE bit of the same

registers, either to generate an interrupt when the History List reached a filling level of 75% or for every new TX History List entry.

An entry lost indication is flagged by the CFDTHLSTS.THLELT bit in the TX History List Status Register. The status of this bit is also shown by the THLES bit in the Global Error Flag Register.

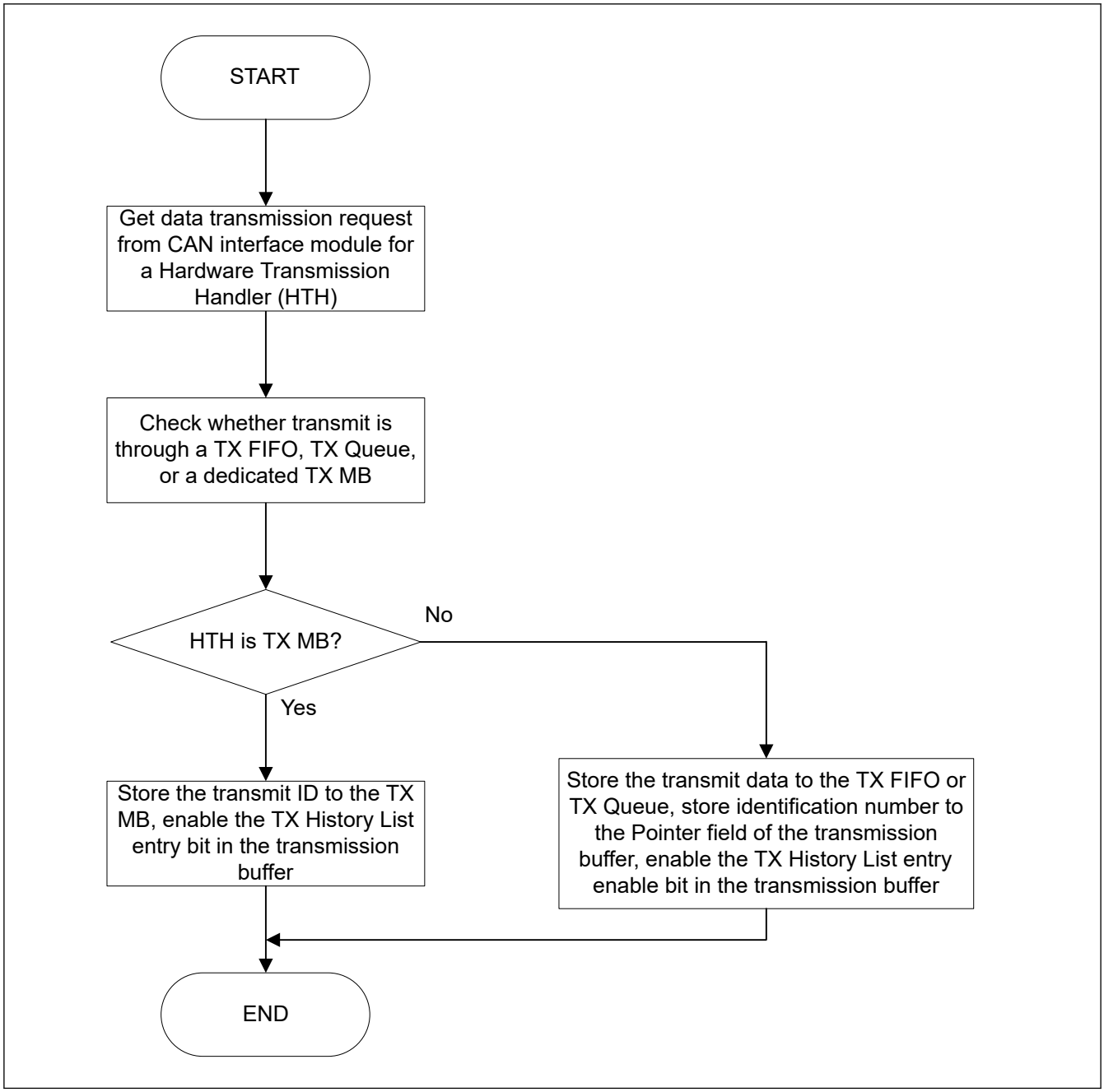


Figure 34.49 TX History List preparation flow

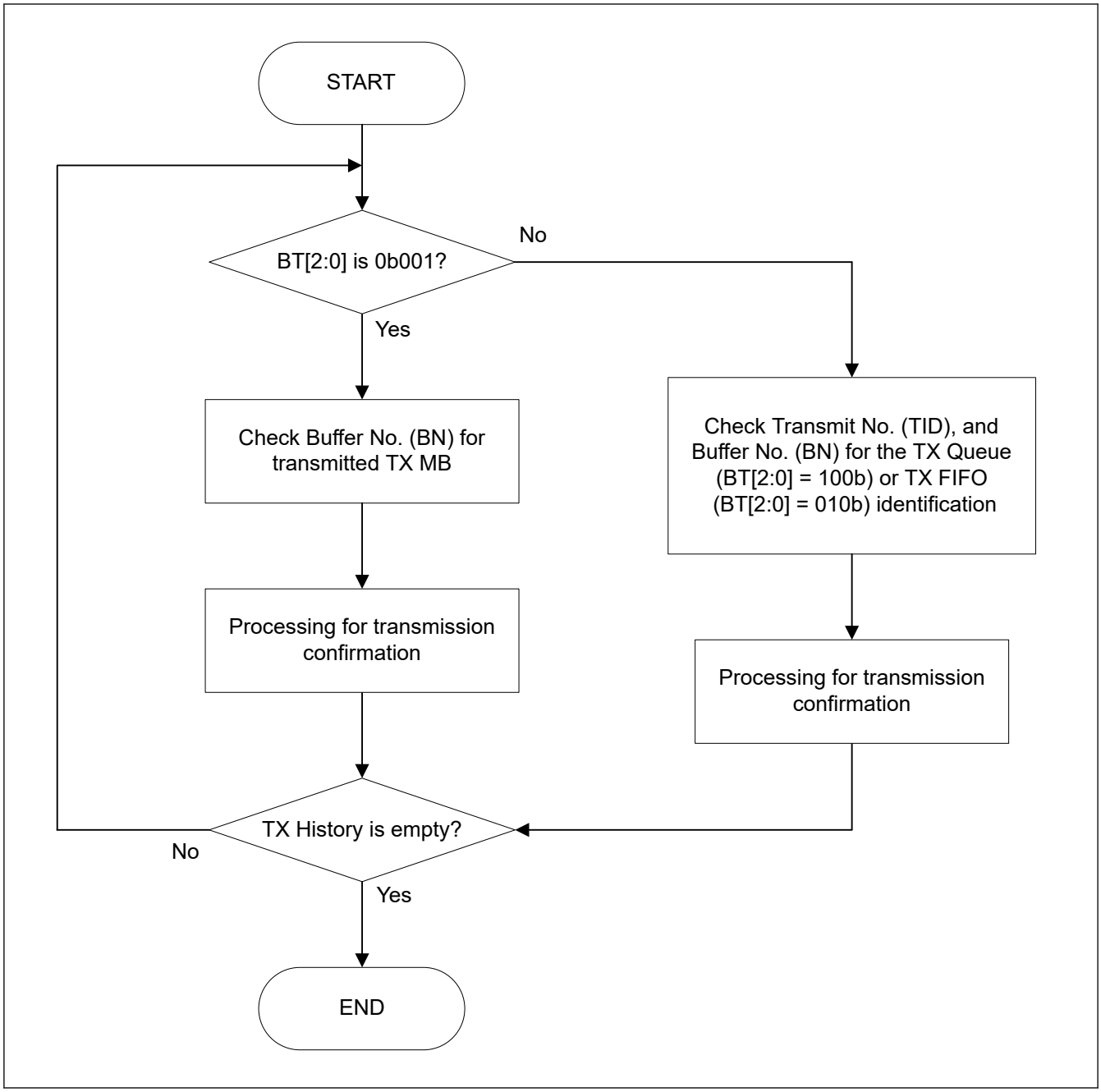


Figure 34.50 TX History List processing flow

34.8.2.6 TX Data Padding

This chapter is not valid for classical CAN.

If the data length code (DLC) of the transmitting message has a higher number of data bytes than the buffer size, the data bytes beyond the restricted range are replaced by bytes with the value of 0xCC.

This can happen for Common FIFO configured as (TX mode) when the transmit message DLC is higher than the CFDCFCC.CFPLS.

This can also happen in FD only mode, if a Classical frame is configured with a DLC bigger than 8.

34.9 Test Mode

The CANFD module can be configured into test modes to allow testing of certain features. These features are provided only for special purposes and care must be taken when configuring the CANFD module in test modes.

Note: All test modes are mutually exclusive unless it is explicitly stated that some functions can be enabled across other test modes.

Do not enable any combination of the various test modes specified in this section.

The test modes can be broadly split into 2 groups:

- Channel specific test modes
- Global test modes.

34.9.1 Channel Specific Test Modes

CAN channel can be configured into the following test modes:

- Basic test mode
- Listen-only mode
- Self-test mode 0 (External loop back mode)
- Self-test mode 1 (Internal loop back mode)
- Restricted operation mode.

34.9.1.1 Basic Test Mode

The basic test mode should be used when there is requirement for a particular test setting to be enabled other than when in Listen-only and Self-test modes.

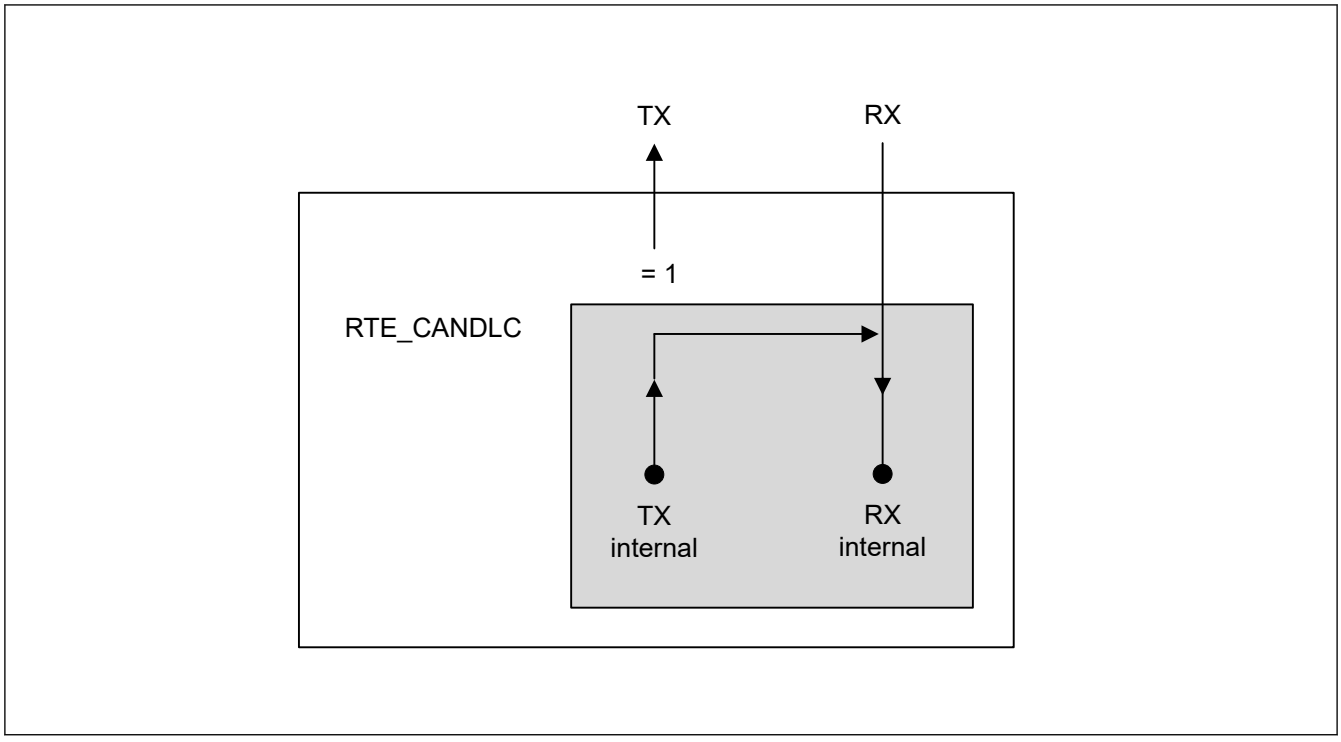
34.9.1.2 Listen-only Mode

The ISO 11898-1 recommends an optional bus-monitoring mode. In this mode, the CAN channel is able to receive valid data frames and valid remote frames. However, it sends only recessive bits on the CAN bus and is not allowed to transmit.

If the CAN engine is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is routed internally so that the CAN engine monitors this as dominant. The external TX pin remains in recessive state.

This mode can be used for baud rate detection. In this mode, an error interrupt is generated if a bus error occurs and the interrupt is enabled.

In this mode, it is not permitted to request transmission from any normal TX message buffer or TX FIFO.

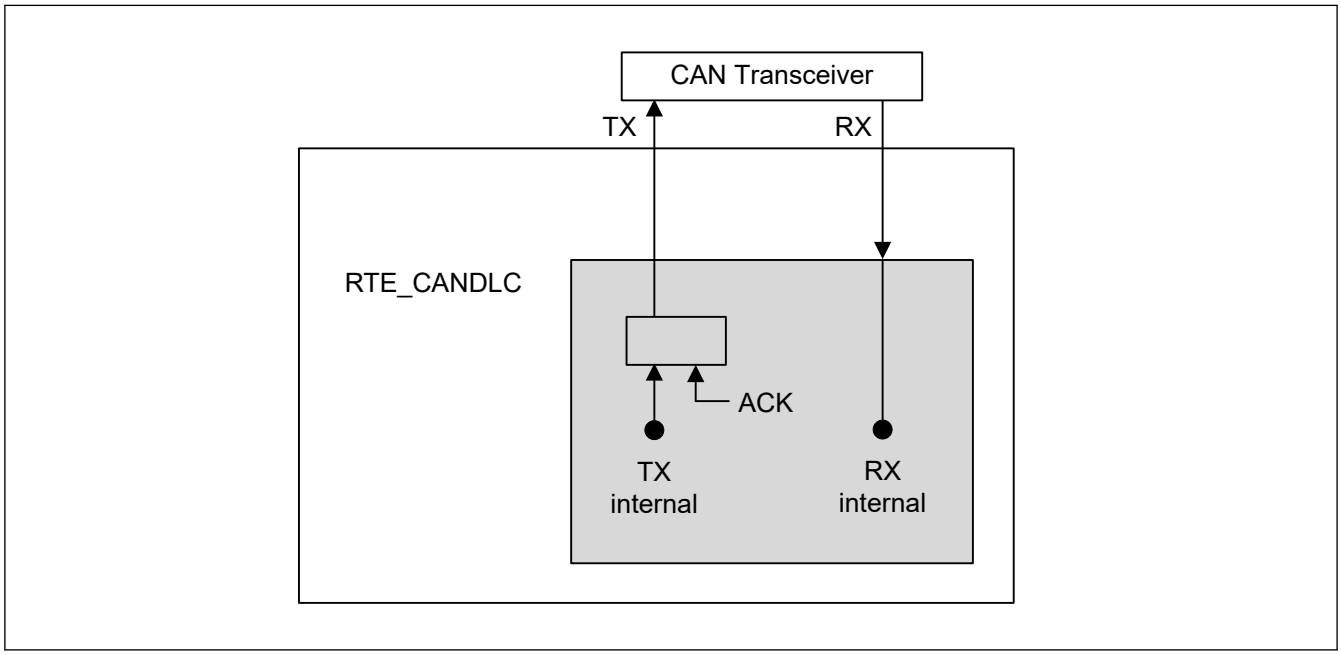


34.9.1.3 Self-test Mode 0 (External loopback mode)

In Self-test mode 0, the CAN engine treats its own transmitted messages as received messages through the CAN transceiver and stores them into its receive message buffers.

To be independent from external stimulation, the engine generates its own Acknowledge bit.

This test can be used for CAN transceiver tests and the RX/TX pins should be connected to the transceiver.

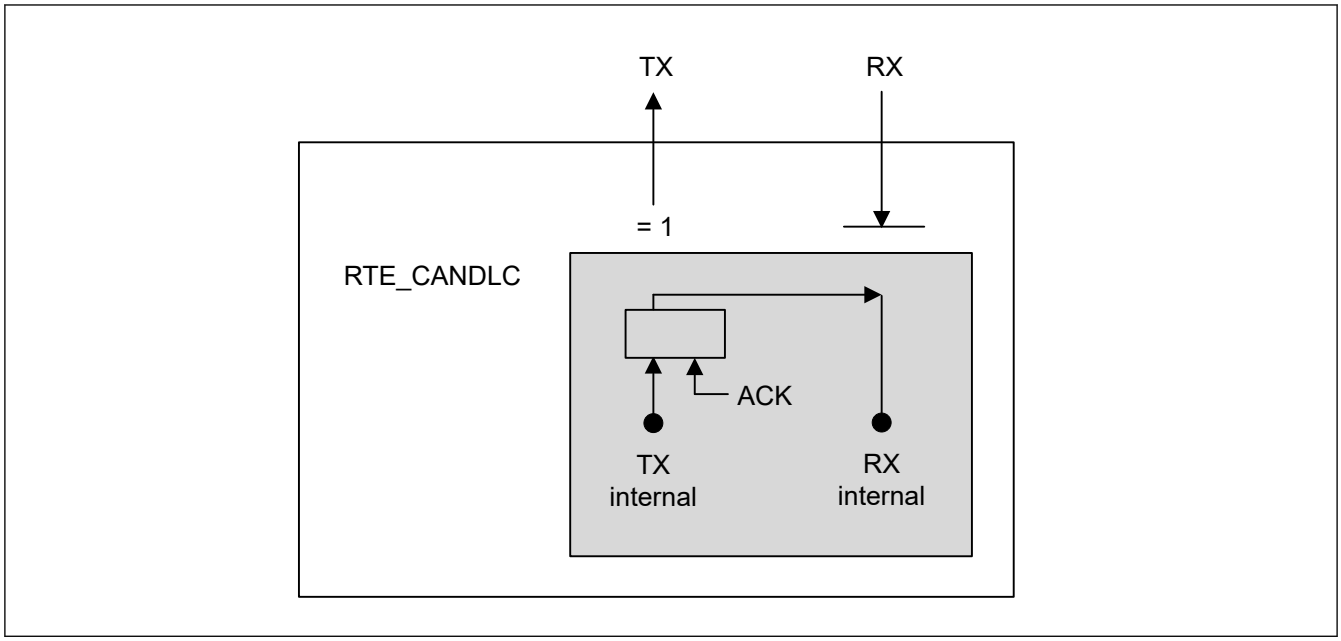


34.9.1.4 Self-test Mode 1 (Internal loopback mode)

In Self-test mode 1, the CAN engine treats its own transmitted messages as received messages and stores them into the receive buffer. This mode is provided for self-test functions. To be independent from external stimulation, the CAN engine generates its own Acknowledge bit. In this mode the CAN engine performs an internal feedback from TX internal to RX internal. The actual value of the external RX input is disregarded by the CAN engine.

The external TX pin outputs only recessive bits. The RX/TX pins do not need to be connected to the CAN bus or any external device.

Note: The channel pins are also disconnected from the internal CAN bus communication line.



34.9.1.5 Restricted Operation Mode

This chapter is not valid for classical CAN.

In Restricted operation mode, the CAN node is able to receive valid data and remote frames generating the Acknowledge bit.

Active error or overload frames cannot be transmitted, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication after an error or overload condition occurs.

Additionally, the Receive and Transmit Error Counter (REC and TEC) are frozen independently from the occurrence of errors. The mode is specified in ISO 11898-1 and the setting of transmit request is permitted.

34.9.2 Global Test Modes

The CANFD module can be configured into the following test modes:

- RAM test mode
- Bit Flip Test

The test modes in the following table are protected by a special software procedure to enable the mode. This software procedure enables write access to the test mode by a specific unlock key as shown in the table.

Test mode	Unlock key 1	Unlock key 2
RAM test mode	0x7575	0x8A8A

If the software sequence of the two consecutive unlock key write accesses (half-word or word access) is interrupted by any other write access to the register or if incorrect data is written to the Global Unlock Key Register, the corresponding test mode cannot be set and the sequence must be restarted.

After the two unlock key write accesses, the next write access should be to set the corresponding test mode enable bit. If this is not followed, the unlock mechanism reset and the test mode enable bit cannot be set and the unlock sequence must be restarted.

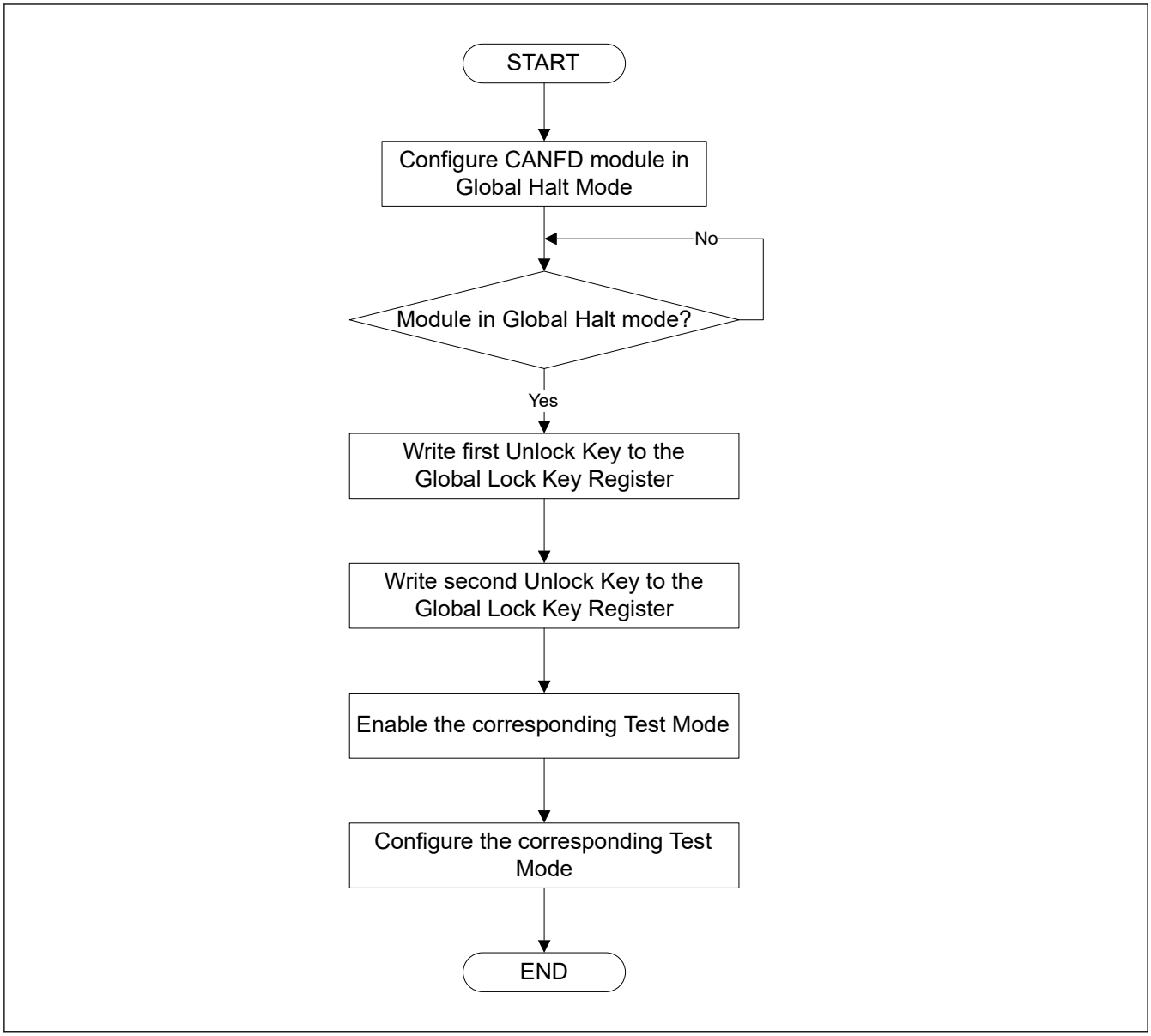


Figure 34.51 Unlock software protection routine

34.9.2.1 RAM Test Mode

The CANFD module can be configured in RAM test mode by setting the CFDGTSTCTR.RTME bit in the Global Test Control Register when the corresponding lock key is previously written. This is a special test mode, in which, the complete RAM area can be accessed.

Note: The actual RAM size is bigger than the RAM area initialized after a hardware reset. Therefore, ECC error flag (of the ECC macro) may be set if CPU reads data from this uninitialized RAM area while CANFD module is in RAM test mode.

In this mode, the RAM area is split into number of pages (pn) of 256 bytes, each which can be accessed with the CFDRPGACCk register.

The page should be selected for read/write access by writing to the CFDGTSTCFG.RTMPS[3:0] bits in the Global Test Control Register. Data can then be read from or written in to the RAM Test Page Access Registers.

Figure 34.52 shows the structure of the pages in the RAM when performing a RAM test mode.

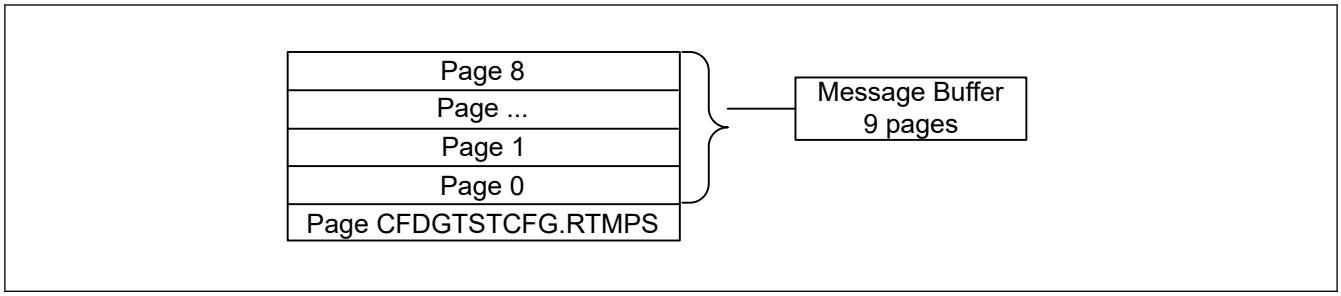


Figure 34.52 RAM page structure

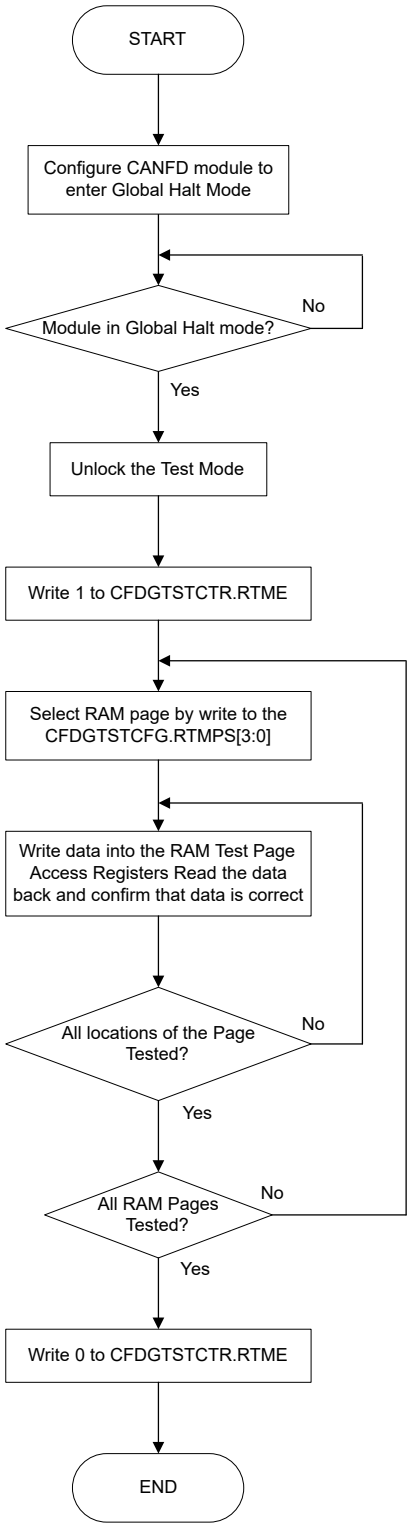
The total available RAM size is 2072 bytes for the Message Buffer RAM.

The pn and CFDGTSTCFG.RTMPS[3:0] values for the MB RAMs are calculated in the following way:

pn = ceil (total RAM size in bytes / number of bytes per page)

- MB RAM:
  pn = ceil (2072 / 256) = 9 pages
  CFDGTSTCFG.RTMPS[3:0] = 0 to 8 inclusive
  (User should not access more than 24 Bytes in the last page)

Figure 34.53 shows the software flow for RAM test mode.



Note 1. Change into the following status before changing to RAMTEST.
- Cancel of a request for transmission.
- Disable of all the FIFO and TXQ.
- Clear of the receiving flag for a receiving buffer.

Figure 34.53 Software flow for RAM test mode

To exit this test mode, the CFDGTSTCTR.RTME bit must be cleared. The CFDGTSTCTR.RTME bit is cleared by writing 0 to it.

The CFDGTSTCTR.RTME bit is cleared automatically when the CANFD module enters Global Reset mode from the test mode.

34.9.2.2 Bit Flip Test

Bit Flip Test can invert the bit (the 1st bit of ID) of the beginning of the bit stream to receive.

If this function is used by a transmitting node, a bit error or an arbitration lost will occur.

If this function is used by a receiving node, a CRC error or a stuff error will occur.

Users should refer to the bit stuffing rule when using this feature, as there is the possibility of receiving a stuff error (due to the inversion) rather than a CRC error.

The following sequence should be used to perform CRC Error testing. In the sequence below CANFD module is the receiver.

1. Set the CFDC0CTR.BFT bit to 1’b1, in order to invert the first bit of the incoming bit stream from sending node
2. Wait for the CANn_CHERR (n = 0, 1) output signal to set to 1’b1
3. Read either the CFDC0ERFL.CRCREG or the CFDC0FDCRC.CRCREG (depending on the received frame type: Classical or FD). The value should be different from the received CRC value of the reference message from sending node.
4. Check that CFDC0ERFL.CERR is 1’b1

As the CRC generator logic is shared for RX and TX there is no need to create a separate TX CRC Error test.

34.10 RAM Area Configuration

The RAM area used in CANFD (refers to as MRAM) can be split into the following groups as shown in Figure 34.54.

- AFL Rule Table area
- PFL Rule Table area
- Message Buffer*1 area (RX MB + FIFO Buffer)
- OTB area
- THL area
- TX MB area.

Physically, the RAM is the Message Buffer RAM*2 (RX MB, RX FIFO, Common FIFO*3 TX MB, THL, OTB, AFL Rule Table, PFL Rule Table).

Note 1. Refers to as MB
Note 2. Refers to as MRAM
Note 3. Refers to as CFIFO

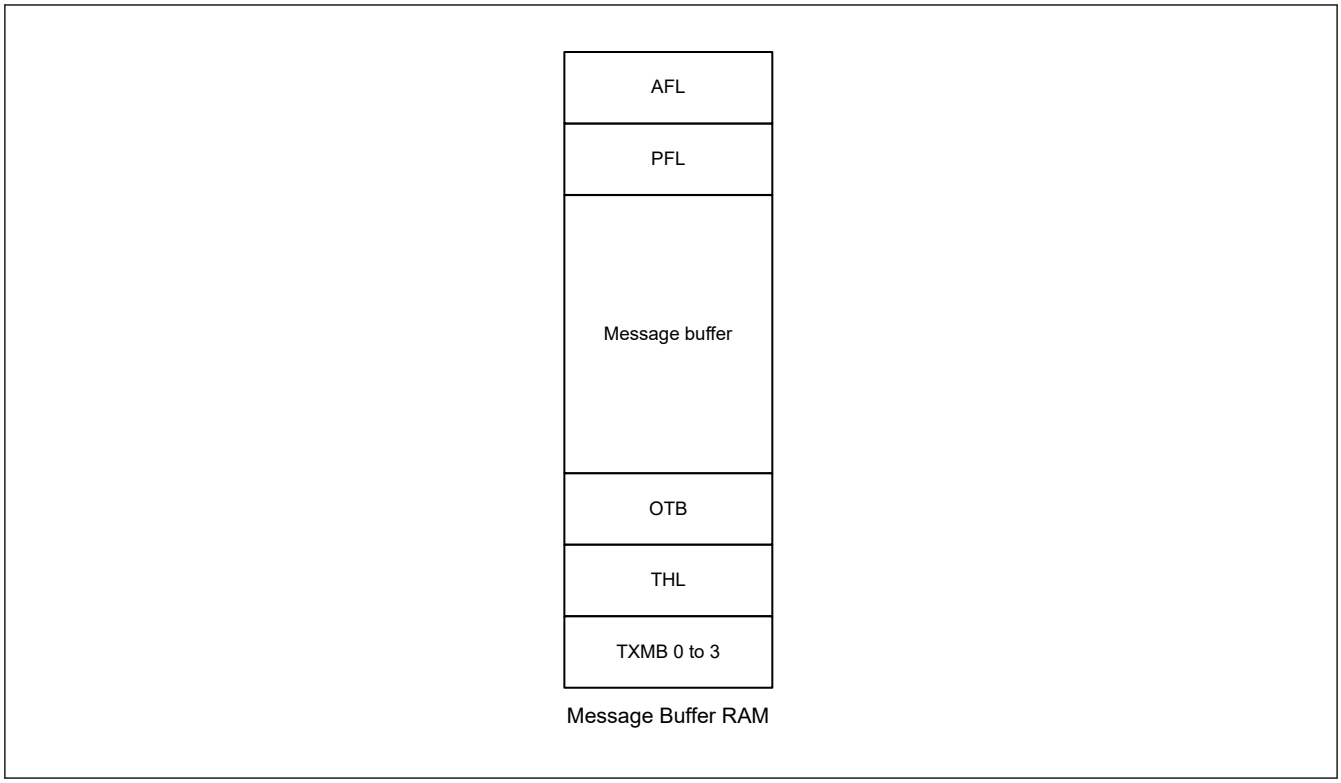


Figure 34.54 RAM area grouping

The MRAM area starts with the TX MB area at address 0x0000. The TX MB area is followed immediately by the THL area which is then followed immediately by the OTB area. The size of the TX MB, THL, and OTB areas is fixed. The OTB area is followed by the Message Buffer area. The Message Buffer area size depends on the configuration of the flat RXMBs, RXFIFOs, and CFIFO. When all are configured, the RX MB area is followed by the RX FIFO area which is followed by the CFIFO area.

The configured MRAM area can then be calculated as follows:

MRAM_cfg = RXMB_MRAM_cfg + RXFIFO_MRAM_cfg + CFIFO_MRAM_cfg + TXMB_MRAM_cfg + THL_MRAM_cfg + OTB_MRAM_cfg + AFL_MRAM_cfg + PFL_MRAM_cfg

RXMB_MRAM_cfg = (12 bytes + CFDRMNB.RMPLS) × CFDRMNB.NRXMB

RXFIFO_MRAM_cfg = SUM ((12 bytes + CFDRFCCa.RFPLS) × CFDRFCCa.RFDC)

CFIFO_MRAM_cfg = (12 bytes + CFDCFCC.CFPLS) × CFDCFCC.CFDC

TXMB_MRAM_cfg = 304 bytes

THL_MRAM_cfg = 64 bytes

OTB_MRAM_cfg = 160 bytes

PFL_MRAM_cfg = 72 bytes

AFL_MRAM_cfg = 256 bytes

“a” means RX FIFO index = [0...no_of_RFIFOs - 1]

no_of_RFIFOs : Number of configured RX FIFOs

Note: For CFDRFCCa.RFDC, CFDCFCC.CFDC, CFDRMNB.RMPLS, CFDRMNB.NRXMB, CFDRFCCa.RFPLS, and CFDCFCC.CFPLS the related number of bytes must be used.

Table 34.29 shows the calculation of the different RAM areas used for the AFL entries, OTB buffers, TX/RX message buffers, RX/Common FIFOs, and PFL entries.

Table 34.29 MRAM area calculation

RAM name	RAM property	RAM area calculation method	RAM values
AFL	Average rule entries	—	16
	Number of bytes in a rule entry	Fixed	16
	Number of bytes in AFL area	Average rule entries × Number of bytes in a rule entry	256
PFL	Average rule entries	—	2
	Number of bytes in a rule entry	Fixed	36
	Number of bytes in PFL area	Average rule entries × Number of bytes in a rule entry	72
TX MB	Number of TX MBs	Fixed	4
	Number of bytes required for each TX MB	Fixed	76
	Number of bytes in TX MB area	Number of TX MBs × Number of bytes required for each TX MB	304
THL	Number of entries in one THL buffer	Fixed	8
	Number of bytes required for each THL entry	Fixed	8
	Number of bytes in THL area	Number of entries in one THL buffer × Number of bytes required for each THL entry	64
OTB	Average number of buffers	—	2
	Number of bytes for OTB entry	Fixed	80
	Number of bytes in OTB area	Average number of buffers × Number of bytes for OTB entry	160
Message Buffer	Number of RX MBs	Fixed	32
	Number of RX FIFOs	Fixed	2
	Number of Common FIFO	Fixed	1
	Average number of messages for RX MB and FIFO buffers	—	16
	Number of bytes for each stored message	Fixed	—
	Average size of a Message Buffer in bytes	—	76
	Number of bytes in Message Pool area	Average number of messages for RX MB and FIFO buffers × Average size of a Message Buffer in bytes	1216
	Number of bytes message RAM	Number of bytes in Message Pool area + Number of bytes in OTB area + Number of bytes in THL area + Number of bytes in TX MB area + Number of bytes in PFL area + Number of bytes in AFL area	2072

34.10.1 Examples

Figure 34.55 shows one possible configuration.

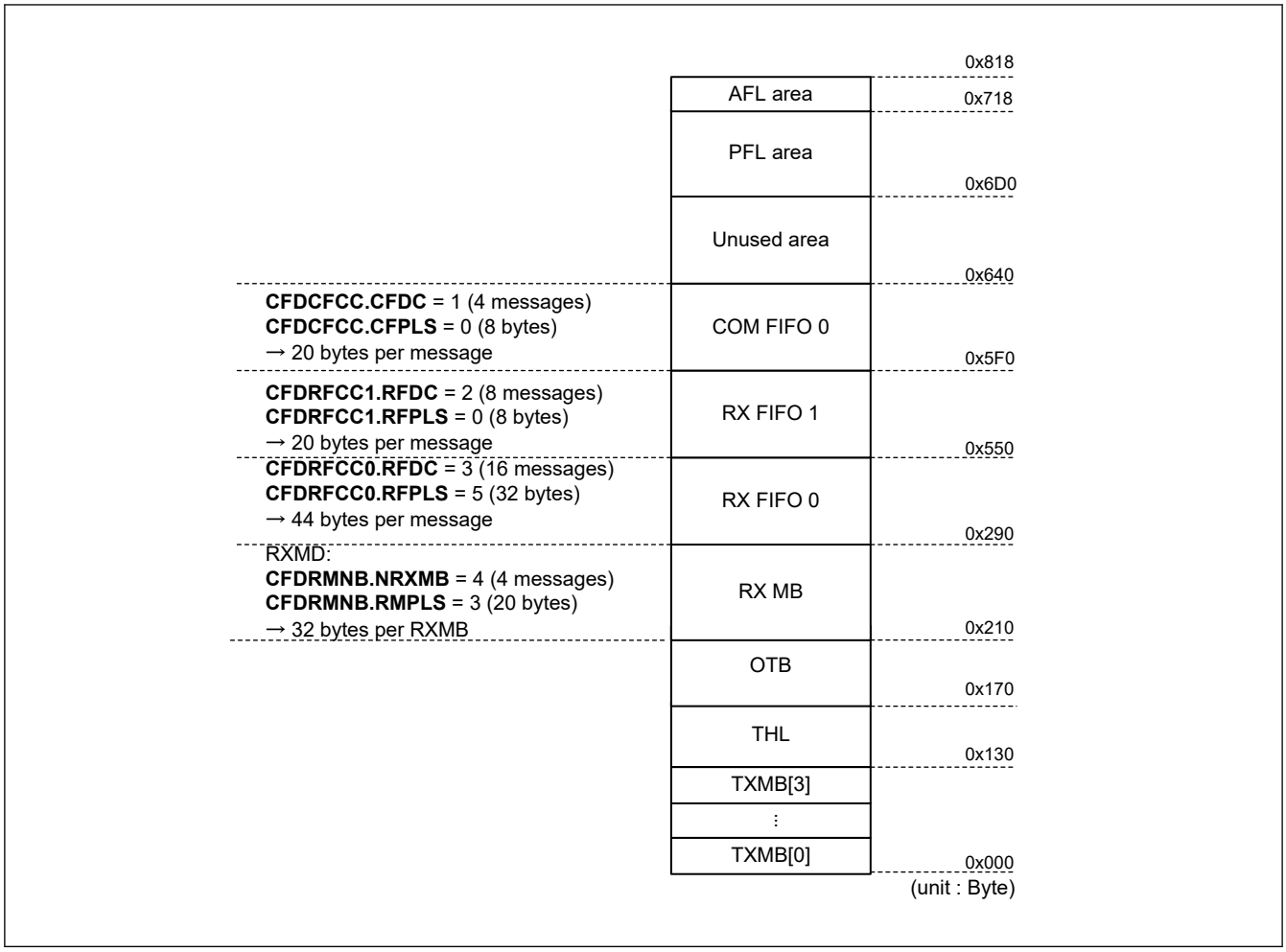


Figure 34.55 RX MB + FIFO buffers RAM area configuration examples

34.10.2 OTB Area

The OTB area starts immediately after the area allocated for THL buffers. The OTB is a special purpose buffer used by the CANFD. This section of RAM area can be accessed only by the CPU in RAM Test mode. Buffer requires 80 bytes and the average number of buffers is 2. Therefore, the total number of bytes allocated for the OTB is 2 × 80 bytes.

34.10.3 RAM Initialization Cycle

The number of RAM initialization cycles and the RAM number of pages are shown in Table 34.30.

Table 34.30 RAM initialization cycle

MRAM area size	RAM initialization cycles [PCLKA cycle]	RAM Test RTMPS range
2072	520	0x0 to 0x8

34.11 Usage notes

34.11.1 Module-stop function

CANFD operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The CANFD module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

35. CANFD ECC (CNECC)

35.1 Overview

MBRAM have ECC function of 2-bit ECC error detection and 1-bit ECC error detection and correction*1. The ECC module adds 7 bits ECC data to 32 bits RAM data.

Note 1. The ECC module cannot detect 3 or more bits error. In this case, the ECC module detects 1-bit or 2-bit error, does not detect errors, or corrects the erroneous bit to erroneous data by setting. When all RAM data are fixed to 0 or 1, it is detected as 2-bit ECC error.

35.2 Register Descriptions

35.2.1 EC710CTL : ECC Control Register

Base address: ECCMBn = 0x4036_F200 + 0x0100 × n (n = 0, 1)
ECCMBn_NS = 0x5036_F200 + 0x0100 × n (n = 0, 1)

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECDEDF0	ECSEDF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EMCA[1:0]		—	—	ECOVFF	ECER2C	ECER1C	—	—	ECERVF	EC1ECP	EC2EDIC	EC1EDIC	ECER2F	ECER1F	ECEMF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	ECEMF	ECC Error Message Flag 0: There is no bit error in present RAM output data 1: There is bit error in present RAM output data	R
1	ECER1F	ECC Error Detection and Correction Flag 0: After clearing this bit, 1-bit error correction has not occurred 1: 1-bit error has occurred	R
2	ECER2F	2-bit ECC Error Detection Flag 0: After clearing this bit, 2-bit error has not occurred 1: 2-bit error has occurred	R
3	EC1EDIC	ECC 1-bit Error Detection Interrupt Control 0: Disable 1-bit error detection interrupt request 1: Enable 1-bit error detection interrupt request	R/W
4	EC2EDIC	ECC 2-bit Error Detection Interrupt Control 0: Disable 2-bit error detection interrupt request 1: Enable 2-bit error detection interrupt request	R/W
5	EC1ECP	ECC 1-bit Error Correction Permission 0: At 1-bit error detection, the error correction is executed 1: At 1-bit error detection, the error correction is not executed	R/W
6	ECERVF	ECC Error Judgment Enable Flag 0: Error judgment disable 1: Error judgment enable	R/W
8:7	—	These bits are read as 0. The write value should be 0.	R/W
9	ECER1C	Accumulating ECC Error Detection and Correction Flag Clear 0: No effect 1: Clear accumulating ECC error detection and correction flag	R/W
10	ECER2C	2-bit ECC Error Detection Flag Clear 0: No effect 1: Clear 2-bit ECC error detection flag	R/W

Bit	Symbol	Function	R/W
11	ECOVFF	ECC Overflow Detection Flag 0: No effect 1: ECC overflow detection flag	R
13:12	—	These bits are read as 0. The write value should be 0.	R/W
15:14	EMCA[1:0]	Access Control to ECC Mode Select bit These bits enable or disable write access to ECERVF bit.	R/W
16	ECSEDF0	ECC Single bit Error Address Detection Flag 0: There is no bit error in EC710EAD0 after reset or clearing ECER1F bit 1: Address captured in EC710EAD0 shows that 1-bit error occurred and captured	R
17	ECDEDF0	ECC Dual Bit Error Address Detection Flag 0: There is no bit error in EC710EAD0 after reset or clearing ECER2F bit 1: Address captured in EC710EAD0 shows that 2-bit error occurred and captured	R
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

ECEMF bit (ECC Error Message Flag)

The ECEMF bit shows that there is error in present read data bus. This bit is updated by every RAM output data.

When RAM output data is undefined and the ECERVF bit is set to 1, the value of this bit is undefined.

[Setting condition]

There is bit error in present RAM output data under the condition that error judgement is enabled.

[Clearing condition]

- Under the condition that there is no 1-bit error in input data to decode circuit
- When ECC error judgement is disabled (ECERVF = 0).

ECER1F bit (ECC Error Detection and Correction Flag)

The ECER1F bit shows that the bit errors are detected in the one part of RAM read data [38:0] at RAM read access when the error judgment is enabled.

When the 1-bit error interrupt output is enabled, error interrupt is generated by setting this flag.

This bit is read-only, so writing 1 or 0 has no effect.

At clearing, write 1 to the ECER1C bit.

When 1-bit error is detected again under the condition that this bit is set, the interrupt is not generated.

[Setting condition]

When the error judgment is enabled and there is 1-bit error to RAM output data (when not setting ECER1C = 1).

[Clearing condition]

- Writing ECER1C = 1
- When ECC error judgement is disabled (ECERVF = 0).

ECER2F bit (2-bit ECC Error Detection Flag)

The ECER2F bit shows that the bit errors are detected in the two parts of RAM read data [38:0] at RAM read access when the error judgment is enabled.

When the 2-bit error interrupt output is enabled, error interrupt is generated by setting this flag.

This bit is read-only, so writing 1 or 0 has no effect.

At clearing, write 1 to the ECER2C bit.

When 2-bit error is detected again under the condition that this bit is set, the interrupt is not generated.

[Setting condition]

When the error judgment is enabled and there is 2-bit error to RAM output data (when not setting ECER2C = 1).

[Clearing condition]

- Writing ECER2C = 1
- When ECC error judgement is disabled (ECERVF = 0).

EC1EDIC bit (ECC 1-bit Error Detection Interrupt Control)

The EC1EDIC controls the interrupt output at detecting 1-bit error. By setting 1 to this bit, the 1-bit error interrupt is outputted when 1-bit error detected.

EC2EDIC bit (ECC 2-bit Error Detection Interrupt Control)

The EC2EDIC controls the interrupt output at detecting 2-bit error. By setting 1 to this bit, the 2-bit error interrupt is outputted when 2-bit error detected.

EC1ECP bit (ECC 1-bit Error Correction Permission)

The EC1ECP sets enable or disable to correct the 1-bit error when ECC error detection and correction is valid. By setting 1 to this bit, the non-corrected data is outputted if 1-bit error is detected.

ECERVF bit (ECC Error Judgment Enable Flag)

Setting the ECERVF bit to 1 enables the judgment of error. The correction of output data and the interrupt output depend on setting of the EC1ECP bit, EC2EDIC bit, and EC1EDIC bit.

The write access to this bit is valid when the write value of the EMCA[1:0] is 01b. So only the 16 bits or 32 bits operation command is valid in the case of the write access to this bit.

ECER1C bit (Accumulating ECC Error Detection and Correction Flag Clear)

The ECER1C bit clears the status flag of the ECER1F bit.

The read value is always 0. By writing 0, the internal condition is not changed. When the competition between writing 1 to this bit and setting the ECER1F bit, the former has priority.

The ECER1F bit is cleared by writing 1 to this bit while the ECER1F bit is set. Additionally, the Overflow Detection flag (ECOVFF), ECC Dual Bit Error flag (ECDEDF0) and ECC Single Bit Error flag (ECSEDF0) are also cleared.

ECER2C bit (2-bit ECC Error Detection Flag Clear)

The ECER2C bit clears the status flag of the ECER2F bit.

The read value is always 0. By writing 0, the internal condition is not changed. When the competition between writing 1 to this bit and setting the ECER2F bit, the former has priority.

The ECER2F bit is cleared by writing 1 to this bit while the ECER2F bit is set. Additionally, the Overflow Detection flag (ECOVFF), ECC Dual Bit Error flag (ECDEDF0), and ECC Single Bit Error flag (ECSEDF0) are also cleared.

ECOVFF bit (ECC Overflow Detection Flag)

The ECOVFF bit is set and the overflow interruption is outputted by detecting the new error address under the condition that error address is already captured in the EC710EAD0 register. The overflow interrupt is outputted again when this bit is set and new error is detected.

This bit is read-only, so writing 1 or 0 has no effect.

To clear this bit, write 1 to the ECER2C bit and the ECER1C bit.

[Setting condition]

When new error address is captured under the condition that error address is already captured in the EC710EAD0 register (when not setting ECER2C = 1 or ECER1C = 1).

[Clearing condition]

- Writing ECER2C = 1 or ECER1C = 1
- When ECC error judgement is disabled (ECERVF = 0).

EMCA[1:0] bit (Access Control to ECC Mode Select bit)

The EMCA[1:0] bits are the write trigger reserved bits to the ECERVF bit. The read value is always 0. When the value of these bits is 01b, it is possible to have write access to the ECERVF bit. If these bits are not 01b, write access to the ECERVF bit is ignored and the value is not written.

ECSEDF0 bit (ECC Single bit Error Address Detection Flag)

The ECSEDF0 bit shows that the error is captured in the error address register when error detection is valid. This bit is set by 1-bit error detection.

When 1-bit error is detected after the 2-bit error address is already captured in the EC710EAD0 register, this bit is not updated but the EC710EAD0 register is updated.

This bit is read-only, so writing 1 or 0 has no effect. To clear these bits, write 1 to the ECER1C bit.

[Setting condition]

When there is 1-bit error to RAM output data and error address is captured in EC710EAD0 under the condition that the error judgment is permitted (when not setting ECER1C = 1).

[Clearing condition]

- Writing ECER1C = 1
- When ECC error judgement is disabled (ECERVF = 0).

ECDEDF0 bit (ECC Dual Bit Error Address Detection Flag)

The ECDEDF0 bit shows that the error is captured in the error address register when error detection is valid. This bit is set by 2-bit error detection.

When 2-bit error is detected after the 1-bit error address is already captured in the EC710EAD0 register, this bit is not updated and the EC710EAD0 register is updated.

This bit is read-only, so writing 1 or 0 has no effect. To clear these bits, write 1 to the ECER2C bit.

[Setting condition]

When there is 2-bit error to RAM output data and error address is captured in EC710EAD0 under the condition that the error judgment is permitted (when not setting ECER2C = 1).

[Clearing condition]

- Writing ECER2C = 1
- When ECC error judgement is disabled (ECERVF = 0).

35.2.2 EC710TMC : ECC Test Mode Control Register

Base address: ECCMBn = 0x4036_F200 + 0x0100 × n (n = 0, 1)
ECCMBn_NS = 0x5036_F200 + 0x0100 × n (n = 0, 1)

Offset address: 0x04

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ETMA[1:0]		—	—	—	—	—	—	ECTMCE	—	—	—	—	—	ECDCS	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	ECDCS	ECC Decode Input Select 0: Input lower 32 bits of RAM output data to data area of decode circuit 1: Input ECEDB31-0 in EC710TED register to data area of decode circuit	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	ECTMCE	ECC Test Mode Control Enable 0: The access to test mode register and bit is disabled 1: The access to test mode register and bit is enabled	R/W
13:8	—	These bits are read as 0. The write value should be 0.	R/W
15:14	ETMA[1:0]	ECC Test Mode Bit Access Control These bits enable or disable write access to ECTMCE bit.	R/W

Note: S-TYPE-3, P-TYPE-3

ECDCS bit (ECC Decode Input Select)

The ECDCS bit selects either the lower 32 bits data value from RAM or value from the internal test register (EDEDB[31:0] in EC710TED) as input signal to decoder.

The write access to this bit is valid under the condition of ECTMCE = 1 (it is possible to set them at the same time.)

This bit is cleared by setting ECTMCE = 0.

ECTMCE bit (ECC Test Mode Control Enable)

The ECTMCE bit selects the access enable or disable to test register and test control bit.

The write access to this bit is valid under the condition that the value of the ETMA[1:0] bits is 10b.

ETMA[1:0] bits (ECC Test Mode Bit Access Control)

The ETMA[1:0] bits are the write trigger reserved bits to the ECTMCE bit. The read value is always 0. When the value of these bits is 10b, it is possible to have write access to the ECTMCE bit. If these bits are not 10b, the write access to the ECTMCE bit is ignored and the value is not written.

35.2.3 EC710TED : ECC Test Substitute Data Register

Base address: ECCMBn = 0x4036_F200 + 0x0100 × n (n = 0, 1)
ECCMBn_NS = 0x5036_F200 + 0x0100 × n (n = 0, 1)

Offset address: 0x0C

Bit position:	31																												0
Bit field:	ECEDB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	ECEDB[31:0]	ECC Test Substitute Data Substitute data in ECC test mode.	R/W

Note: S-TYPE-3, P-TYPE-3

This register is for the 32 bits data for ECC decode. It is possible to read and write using 32 bits operation command when ECTMCE = 1. When ECTMCE = 0, all bits are always 0.

ECEDB[31:0] bits (ECC Test Substitute Data)

When ECDCS in EC710TMC register is 1, the value of this register is bits [31:0] of the input data to the decode circuit.

35.2.4 EC710EAD0 : ECC Error Address Register

Base address: ECCMBn = 0x4036_F200 + 0x0100 × n (n = 0, 1)
ECCMBn_NS = 0x5036_F200 + 0x0100 × n (n = 0, 1)

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	ECEAD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	ECEAD[9:0]	ECC Error Address	R

Bit	Symbol	Function	R/W
31:10	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

This is a read-only register to hold the ECC error address.

ECEAD[9:0] bits (ECC Error Address)

When ECC error is detected for permitting ECC error judgment, RAM address is captured by the detected signal as a trigger and is hold as the error occurring address. The error address is not captured when the error occurred again to the one held by the same factor.

If 2-bit error occurred under the condition that 1-bit error address is already captured, the 2-bit error address is over-written and the ECDEDF0 bit is set to 1.

If 1-bit error occurred under the condition that 2-bit error address is already captured, the 1-bit error address is not overwritten and the ECSEDF0 bit is not set to 1.

35.3 Operation

35.3.1 ECC Function Setting

Figure 35.1 shows a procedure for ECC function setting.

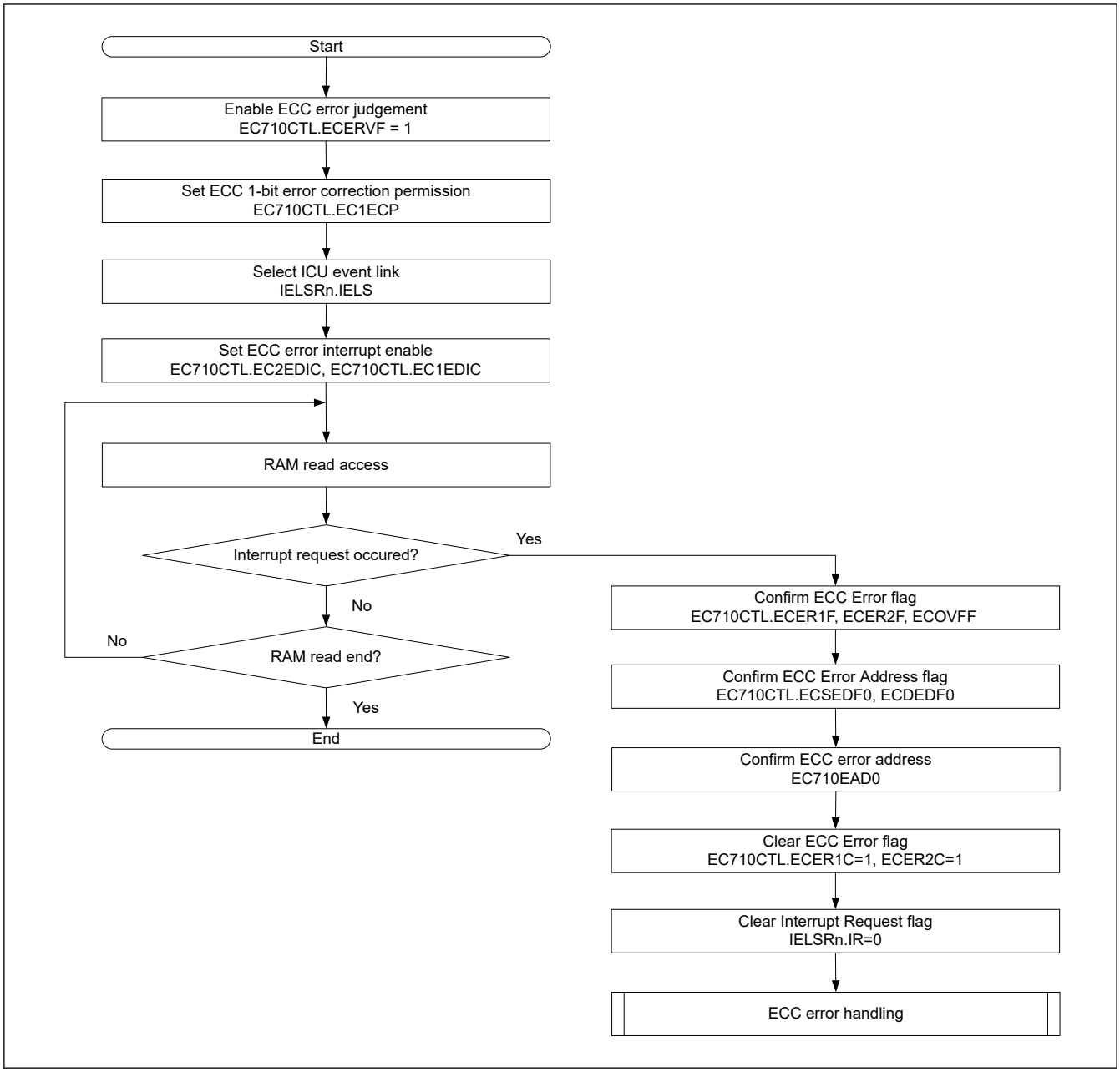


Figure 35.1 Setting procedure for ECC function

35.3.2 ECC Decoder Testing

ECC interrupts can be intentionally generated by ECC test mode. Figure 35.2 shows a procedure for ECC decoder testing.

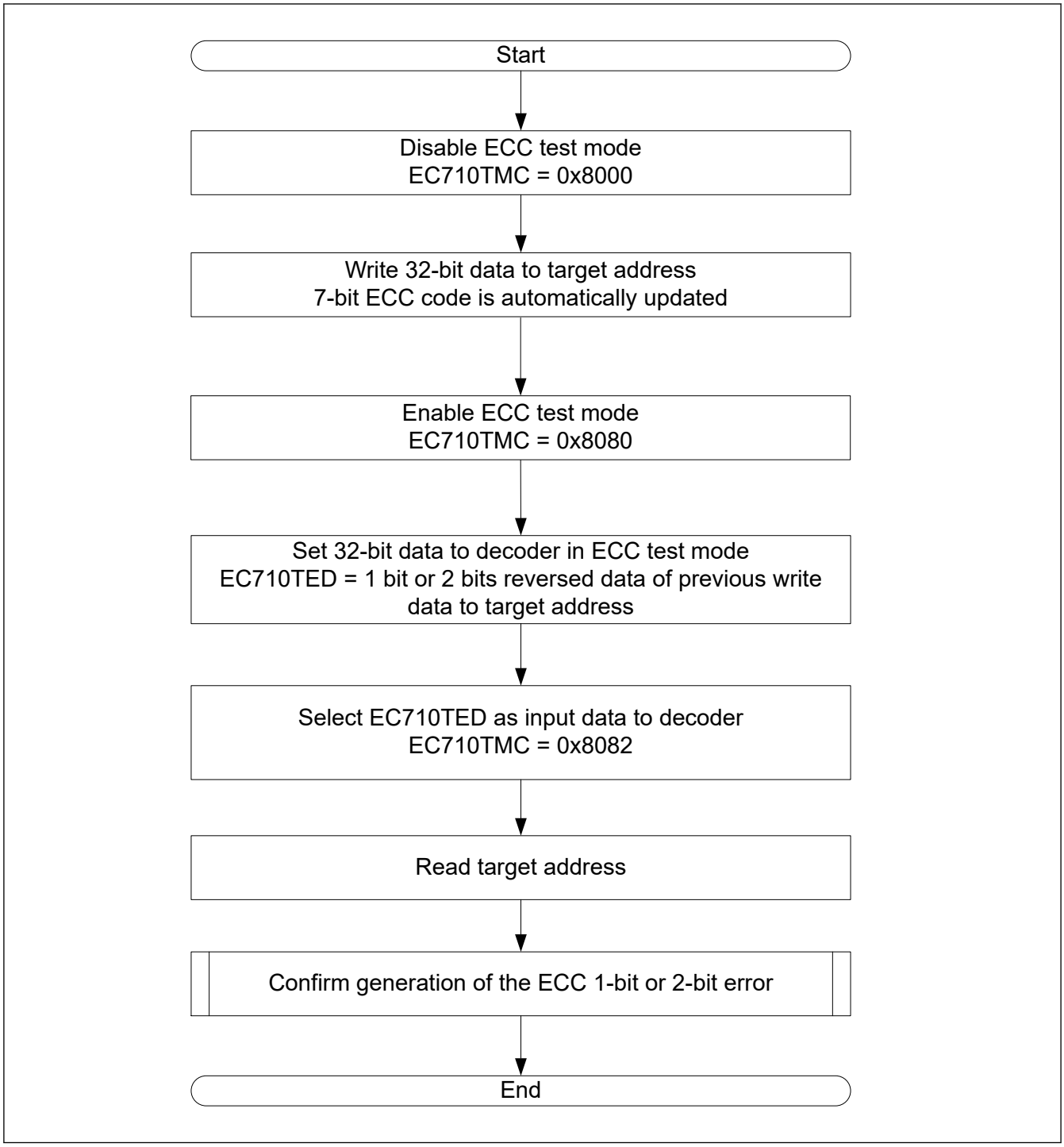


Figure 35.2 Testing procedure for ECC decoder

35.4 Interrupts

The ECC module issues three interrupt requests:

- CANn_MRAM_ERI (n = 0, 1)

Interrupt sources of each interrupt request include:

- 1-bit ECC error
- 2-bit ECC error
- ECC error overflow.

36. Serial Peripheral Interface (SPI)

This is the SPI_B version of the SPI peripheral module.

SPI_B is referred to as SPI in this chapter.

36.1 Overview

The Serial Peripheral Interface (SPI) has 2 channels. The SPI provides high-speed full-duplex synchronous serial communications with multiple processors and peripheral devices. Table 36.1 lists the SPI specifications, Figure 36.1 shows a block diagram of SPI, Figure 36.2 shows a clock source selector block diagram and Table 36.2 lists the I/O pins.

In this section, PCLK refers to PCLKA.

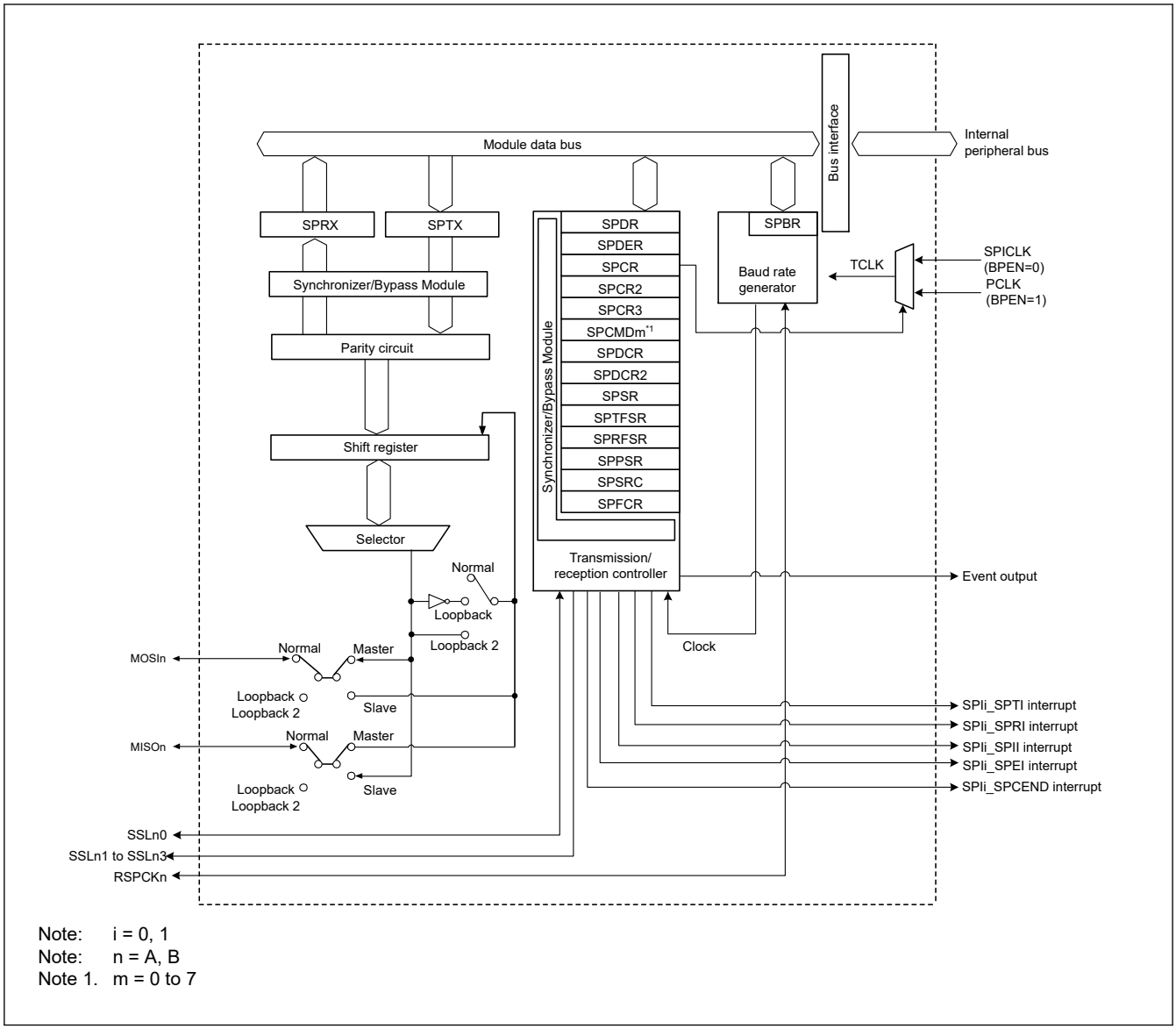
Table 36.1 SPI specifications (1 of 2)

Parameter	Specifications
Number of channels	Two channels
SPI transfer functions	• Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (SPI clock) signals allows serial communications through SPI operation (4-wire method) or clock synchronous operation (3-wire method) • Transmit-only operation available • Receive-only operation is available • Communication mode selectable to full-duplex, transmit-only or receive-only • RSPCK polarity switching • RSPCK phase switching
Data format	• MSB-first or LSB-first selectable • Transfer bit length selectable from 4 to 32 bits • 32 bit × 4 stages FIFO is available as transmit buffer or receive buffer • Byte swap operating function • Transmit/receive data can be inverted.
Operation clock (TCLK)	Synchronized clock (PCLK) or independent clock (SPICLK) can be selected.
Bit rate	• In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing TCLK (the division ratio ranges from divided by 2 to divided by 4096) • In slave mode, the minimum TCLK clock divided by 2 can be input as RSPCK (TCLK divided by 2 is the maximum RSPCK frequency) Width at high level: 1 TCLK cycle; width at low level: 1 TCLK cycle
Buffer configuration	• Double buffer configuration for the transmit and receive buffers
Error detection	• Mode fault error detection • Underrun error detection • Overrun error detection*1 • Parity error detection • Receive data ready detection
SSL control function	[motorola SPI mode/TI SSP mode common] • Four SSL pins (SSLni: SSLn0 to SSLn3) (n = A, B) for each channel • In single-master mode, SSLn0 to SSLn3 pins are output • In multi-master mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins either for output or unused • In slave mode, SSLn0 pin for input and SSLn1 to SSLn3 pins unused • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Function for changing SSL polarity • Delay between frames in burst transfer is settable [only Motorola mode] • Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) [only TI SSP mode] • Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 0 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from OE output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)
Communication protocol	• Motorola SPI • TI SSP(Synchronous Serial Protocol)

Table 36.1 SPI specifications (2 of 2)

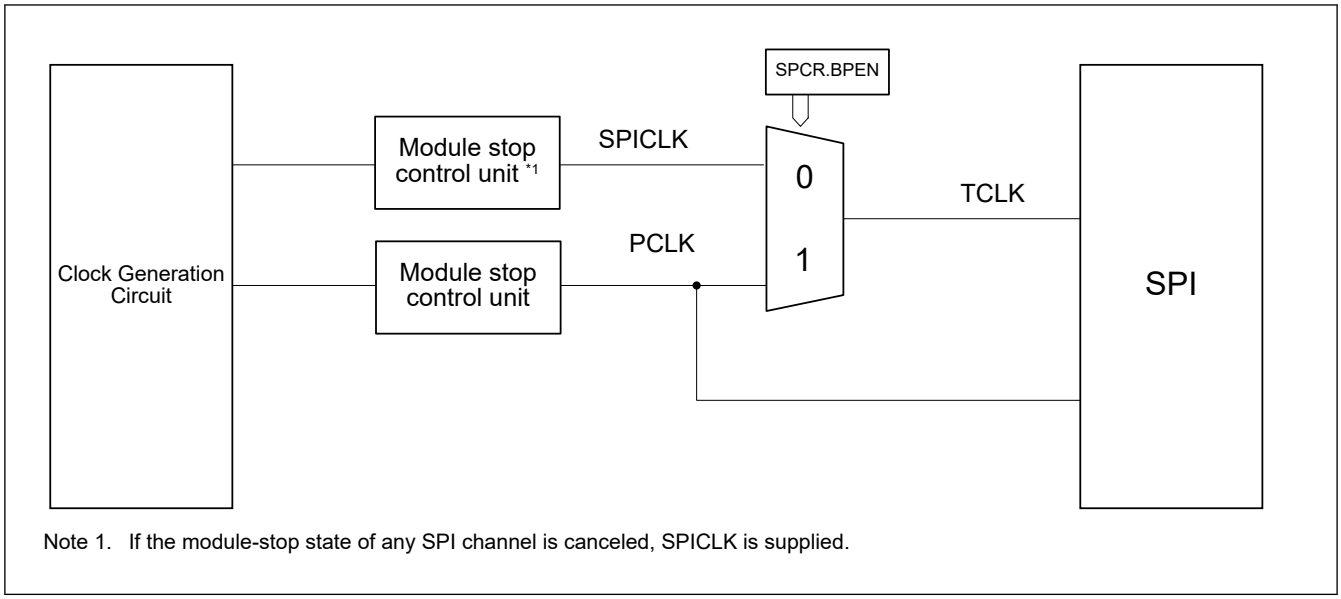
Parameter	Specifications
Synchronization bypass function	Synchronization circuit can be bypassed only when the same clock is input to bus clock (PCLK) and operation clock (TCLK).
Control in master transfer	● Transfers of up to eight commands each can be executed sequentially in looped execution ● For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity and phase, transfer data length, MSB- or LSB-first, burst, RSPCK delay, SSL negation delay, and next-access delay ● Transfers can be initiated by writing to the transmit buffer ● MOSI signal value specifiable in SSL negation ● RSPCK auto-stop function
Interrupt sources	Interrupt sources: ● Receive buffer full / Receive data ready interrupt ● Transmit buffer empty interrupt ● SPI error interrupt (mode fault error, under run error, over run error, parity error, receive data ready) ● SPI idle interrupt (SPI idle) ● Communication end interrupt
Event link function	The following events can be output to the Event Link Controller (ELC): ● Receive buffer full / receive data ready signal ● Transmit buffer empty signal ● Mode fault, underrun, overrun, parity error, or receive data ready signal ● SPI idle signal ● Communication end signal
Other functions	● Switching between CMOS output and open-drain output ● SPI initialization function ● Loopback mode ● SPE bit status polling function
Module-stop function	Module-stop state can be set for each channels to reduce power consumption.
TrustZone Filter	Security and Privilege attribution can be set for each channels.

Note 1. In master reception and when the RSPCK auto-stop function is enabled, an overrun error does not occur because the transfer clock is stopped on overrun error detection.



Note: i = 0, 1
Note: n = A, B
Note 1. m = 0 to 7

Figure 36.1 SPI block diagram



Note 1. If the module-stop state of any SPI channel is canceled, SPICLK is supplied.

Figure 36.2 Clock source selector block diagram

The SPI automatically switches the I/O direction of the SSLn0 pin. SSLn0 is set as an output when the SPI is a single master, and as an input when the SPI is a multi-master or a slave. The RSPCKn, MOSIn, and MISOn pins are automatically set as inputs or outputs based on the master or slave setting and the level input on the SSLn0 pin. For details, see section 36.3.2. Controlling the SPI Pins.

Table 36.2 SPI I/O pins

Channel	Pin name	I/O	Description
SPI0	RSPCKA	I/O	Clock input/output pin
	SSLA0	I/O	Slave selection input/output
	SSLA1 to SSLA3	Output	Slave selection output
	MOSIA	I/O	Master transmit data input/output
	MISOA	I/O	Slave transmit data input/output
SPI1	RSPCKB	I/O	Clock input/output pin
	SSLB0	I/O	Slave selection input/output
	SSLB1 to SSLB3	Output	Slave selection output
	MOSIB	I/O	Master transmit data input/output
	MISOB	I/O	Slave transmit data input/output

Note: Pin names are indicated as “...A” or “...An” for SPI0, and “...B” or “...Bn” for SPI1 (n = 0, 1, 2, or 3).

36.2 Register Descriptions

36.2.1 SPDR : SPI Data Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x00

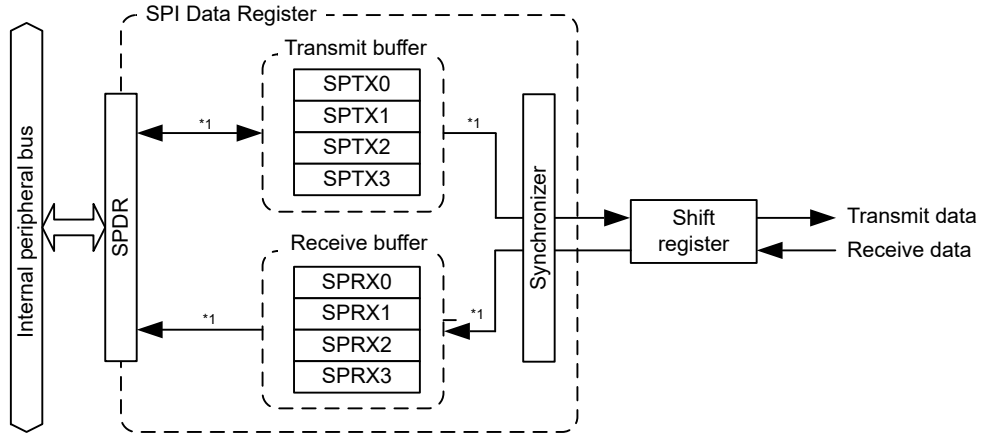
Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SPD[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SPD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SPD[31:0]	These bits are the interface with the buffers that hold data for transmission and reception by the SPI.	R/W

Note: S-TYPE-3, P-TYPE-3

SPDR is the interface with the buffers that hold data for transmission and reception by the SPI. When accessing this register in words, access SPDR. The transmit buffer (SPTX) and receive buffer (SPRX) are independent but both are mapped to SPDR. Figure 36.3 shows the configuration of the SPDR register.



Note 1. The destination buffer and stage for access is automatically switched by hardware.

Figure 36.3 Structure of SPDR

32 bits × 4 stage transmit FIFO and 32 bits × 4 stage receive FIFO are provided. These 8 stage FIFO are mapped to one address in the SPDR. Transmit buffers (SPTXn, n = 0 to 3) can be written by writing data to SPDR to transmit written data.

Upon completion of receiving data, receive buffers store received data. When an overrun error occurs, data in the receive buffer is not updated.

(1) Bus Interface

The SPI data register has 32 bits × 4 stage transmit FIFO and 32 bits × 4 stage receive FIFO (32 bytes in total). These 32 bytes are mapped to the 4-byte space of SPDR. Write transmit data from the LSB. Received data is stored from the LSB.

SPDR register write operation and read operation are described below.

1. Write

A transmit buffer write pointer is provided for transmit buffers. When data is written to SPDR, the pointer automatically switches to the next buffer. The following illustrates the structure of the transmit buffer bus interface (write).

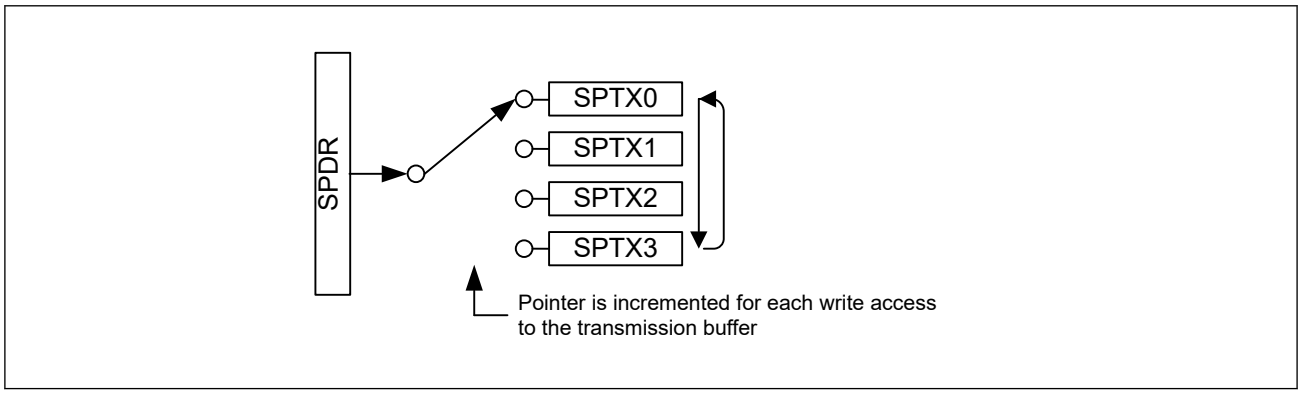


Figure 36.4 Structure of SPDR (Write)

The transmit buffer (SPTX0 to SPTX3) switching order:
SPTX0→SPTX1→SPTX2→SPTX3→SPTX0→SPTX1→…
When writing transmit data to transmit buffers (SPTXn), write transmit data of frames +1 specified by the Transmission FIFO threshold setting bits of SPI data control register 2 (SPDCR2.TTRG[1:0]) while an SPI transmit buffer empty interrupt is present (SPSR.SPTEF flag = 1). Writing to the transmit buffer (SPTXn, n = 0 to 3) in the state where there is no empty stage in the transmit FIFO does not update the buffer value.

2. Read

Values can be read from receive buffers (SPRXn, n = 0 to 3) or transmit buffers (SPTXn, n = 0 to 3) by reading the SPDR register. Reading a receive buffer or reading a transmit buffer can be selected by the SPI receive data or transmit data select bit (SPDCR.SPRDTD) in the SPI data control register.
The SPDR register is read according to the independent receive buffer read pointer and the transmit buffer read pointer. The following illustrates the structure of the receive buffer and transmit buffer bus interface (read).

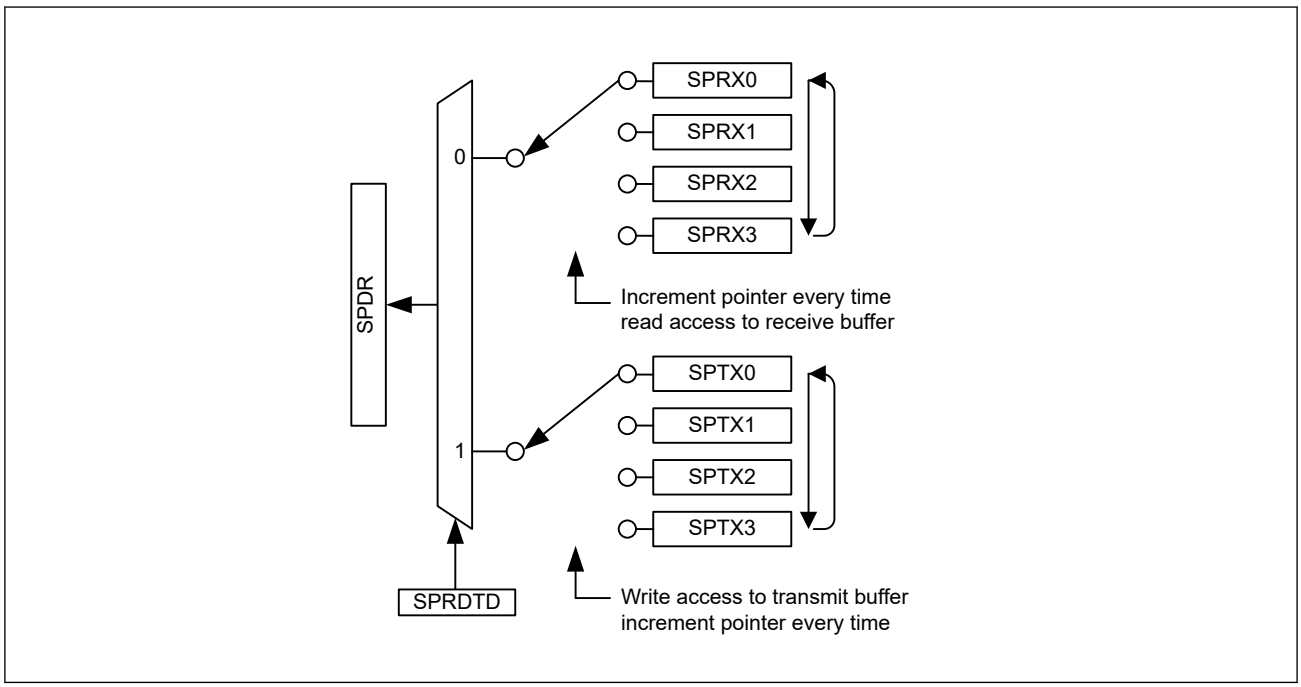


Figure 36.5 Structure of SPDR (Read)

When a receive buffer is read, the receive buffer read pointer automatically switches to the next buffer. The receive buffer read pointer switches in the same order as the transmit buffer write pointer.
The transmit buffer read pointer is updated during the SPDR write access, but it is not updated during the transmit buffer read access. When a transmit buffer is read, the value written to SPDR last can be read.

36.2.2 SPDECR : SPI Delay Control Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SPNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SLNDL[2:0]			—	—	—	—	—	SCKDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SCKDL[2:0]	RSPCK Delay 0 0 0: 1RSPCK 0 0 1: 2RSPCK 0 1 0: 3RSPCK 0 1 1: 4RSPCK 1 0 0: 5RSPCK 1 0 1: 6RSPCK 1 1 0: 7RSPCK 1 1 1: 8RSPCK	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
10:8	SLNDL[2:0]	SSL Negation Delay [Master Mode] 0 0 0: 1RSPCK 0 0 1: 2RSPCK 0 1 0: 3RSPCK 0 1 1: 4RSPCK 1 0 0: 5RSPCK 1 0 1: 6RSPCK 1 1 0: 7RSPCK 1 1 1: 8RSPCK [Motorola-SPI case in Slave Mode] 0 0 0: 1RSPCK Others: Setting prohibited [TI-SSP case in Slave Mode] 0 0 0: 1 TCLK 0 0 1: 2 TCLK 0 1 0: 3 TCLK 0 1 1: 4 TCLK 1 0 0: 5 TCLK 1 0 1: 6 TCLK 1 1 0: 7 TCLK 1 1 1: 8 TCLK	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
18:16	SPNDL[2:0]	SPI Next-Access Delay 0 0 0: 1RSPCK + 5TCLK 0 0 1: 2RSPCK + 5TCLK 0 1 0: 3RSPCK + 5TCLK 0 1 1: 4RSPCK + 5TCLK 1 0 0: 5RSPCK + 5TCLK 1 0 1: 6RSPCK + 5TCLK 1 1 0: 7RSPCK + 5TCLK 1 1 1: 8RSPCK + 5TCLK	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

SCKDL[2:0] bit (RSPCK Delay)

[In the Motorola-SPI case]

The RSPCK delay bits (SCKDL) are used to set the period (RSPCK delay) from SSL signal assertion start until RSPCK oscillates while the SCKDEN bit in the SPI command register (SPCMD) is 1. If SCKDL is modified while the MSTR bit and the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

To use the SPI in slave mode, set SCKDL[2:0] bits to 000b.

[In the TI-SSP case]

The RSPCK delay bits (SCKDL) are used to set the period (RSPCK delay) from SSL signal assertion start until RSPCK oscillates while the SCKDEN bit in the SPI command register (SPCMD) is 1. Also that is used to set the period until the SSL signal is negated. If SCKDL is modified while the MSTR bit and the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

To use the SPI in slave mode, set SCKDL[2:0] bits to 000b.

SLNDL[2:0] bit (SSL Negation Delay)

[In the Motorola-SPI case]

The SSL negation delay bits (SLNDL) are used to set the period (SSL negation delay) after the SPI in master mode sends the final RSPCK edge during serial transfer until it negates the SSL signal while the SLNDEN bit in the SPI command register (SPCMD) is 1. If SLNDL is modified while the MSTR bit and the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

To use the SPI in slave mode except TI-SSP, set SLNDL[2:0] bits to 000b.

[In the TI-SSP case]

The SSL negation delay bits (SLNDL) are used to set the period (OE negation delay) after the SPI in master mode sends the final RSPCK edge during serial transfer until it negates the OE signal while the SLNDEN bit in the SPI command register (SPCMD) is 1. Also, that is used to set the period from when the SPI in slave mode detects the last RSPCK edge of serial transfer to when the OE signal is negated. If SLNDL is modified while the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

SPNDL[2:0] bit (SPI Next-Access Delay)

The SPI next-access delay register (SPDECR.SPNDL) is used to set the SSL signal inactive period (next-access delay) after completion of serial transfer while the SPNDEN bit in the SPI command register (SPCMD) is 1. If SPNDL is modified while the MSTR bit and the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

These bits are used to set the next-access delay value when the SPNDEN bit in SPCMD is 1. To use the SPI in slave mode, set SPNDL[2:0] bits to 000b.

36.2.3 SPCR : SPI Control Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BPEN	MSTR	TXMD[1:0]		—	—	SPFRF	SPMS	—	—	CENDIE	SPTIE	SPDRES	SPIIE	SPRIE	SPEIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MODFEN	BFDS	SCKASE	PTE	—	SPOE	SPPE	—	—	—	—	—	—	—	SPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPE	SPI Function Enable 0: SPI function is disabled. 1: SPI function is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	SPPE	Parity Enable 0: A parity bit is not added to transmit data. Received-data parity check is not performed. 1: A parity bit is added to transmit data. Received-data parity check is performed.	R/W
9	SPOE	Parity Mode 0: Even parity is used for transmission and reception. 1: Odd parity is used for transmission and reception.	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	PTE	Parity Self-Diagnosis Enable 0: Parity circuit self-diagnosis function is disabled. 1: Parity circuit self-diagnosis function is enabled.	R/W
12	SCKASE	RSPCK Auto-Stop Function Enable 0: RSPCK auto-stop function is disabled. 1: RSPCK auto-stop function is enabled.	R/W
13	BFDS	Between Burst Transfer Frames Delay Select 0: Delay (RSPCK delay, SSL negation delay and next-access delay) between frames is inserted in burst transfer 1: Delay between frames is not inserted in burst transfer.	R/W
14	MODFEN	Mode Fault Error Detection Enable 0: Mode fault error detection is disabled. 1: Mode fault error detection is enabled.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
16	SPEIE	SPI Error Interrupt Enable 0: SPI error interrupt request is disabled. 1: SPI error interrupt request is enabled.	R/W
17	SPRIE	SPI Receive Buffer Full Interrupt Enable 0: SPI receive buffer full interrupt request is disabled. 1: SPI receive buffer full interrupt request is enabled.	R/W
18	SPIIE	SPI Idle Interrupt Enable 0: Idle interrupt request is disabled. 1: Idle interrupt request is enabled.	R/W
19	SPDRES	SPI receive data ready error select Select the interrupt request to be generated when the reception data ready is detected 0: Receive data full interrupt 1: Error interrupt	R/W
20	SPTIE	SPI Transmit Buffer Empty Interrupt Enable 0: SPI transmit buffer empty interrupt request is disabled. 1: SPI transmit buffer empty interrupt request is enabled.	R/W
21	CENDIE	SPI Communication End Interrupt Enable 0: Communication end interrupt request is disabled. 1: Communication end interrupt request is enabled.	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
24	SPMS	SPI Mode Select 0: SPI operation (4-wire) 1: Clock synchronous operation (3-wire)	R/W
25	SPFRF	SPI Frame Format Select 0: Motorola-SPI 1: TI-SSP Note: When SPMS = 1 (clock synchronous operation (3-wire)), this bit setting is invalid.	R/W

Bit	Symbol	Function	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R/W
29:28	TXMD[1:0]	Communication Mode Select 0 0: Transmit-Receive 0 1: Transmit only Others: Receive only	R/W
30	MSTR	SPI Master/Slave Mode Select 0: Slave mode 1: Master mode	R/W
31	BPEN	Synchronization Circuit Bypass Enable 0: Non-Bypass 1: Bypass	R/W

Note: S-TYPE-3, P-TYPE-3

The SPI control register (SPCR) is used to set operating mode of the SPI. If the set BPEN, MSTR, TXMD[1:0], SPFRF, SPMS, MODFEN, BFDS, SCKASE, PTE, SPOE, SPPE bit value is modified while the SPE bit = 1, subsequent operation is not guaranteed.

SPE bit (SPI Function Enable)

This bit is used to enable or disable SPI functions. Setting this bit to 1 enables SPI functions. When the MODF flag in the SPI status register (SPSR) is 1, the SPE bit is cleared to 0 and the SPE bit cannot be set to 1 until the MODF flag is cleared to 0. (See section 36.3.10. Error Detection) Setting the SPE bit to 0 disables SPI functions and initializes a part of module functions. (See section 36.3.11. Initializing the SPI)

SPPE bit (Parity Enable)

This bit is used to enable or disable the parity function.

SPOE bit (Parity Mode)

This bit is used to specify even parity or odd parity.

In even parity mode, the parity bit is determined so that the sum of 1 (parity bit + transmit characters or receive characters) becomes an even number. In the same way, in odd parity mode, a parity bit is determined so that the sum of 1 (parity bit + transmit characters or receive characters) becomes an odd number. The SPOE bit is valid only when the SPPE bit in SPCR is set to 1.

PTE bit (Parity Self-Diagnosis Enable)

This bit is used to enable or disable self-diagnosis of the parity circuit to confirm that the parity function is normal.

SCKASE bit (RSPCK Auto-Stop Function Enable)

This bit is used to enable or disable the RSPCK auto-stop function. When this function is enabled, the RSPCK clock stops immediately before an overrun error occurs during data reception in master mode. For details, see section 36.3.10.1. Overrun errors.

BFDS bit (Between Burst Transfer Frames Delay Select)

This bit controls whether insert the delay time between the burst transfer frames.

Valid in the master mode (SPCR.MSTR = 1) for frames with the SPCMDn.SSLKP bit set to 1.

This bit should be set to 0 in slave mode. The usage of SSL delay control between transfer frames is shown as below. For details, see section 36.3.12.1. Master mode operation.

1. Non-burst transmits
2. Burst transmit with delay between frames
   2-1. From the 1st frame to the last previous frame
   2-2. The last frame
3. Burst transmit with no delay between frames
   3-1. From the 1st frame to the last previous frame

3-2. The last frame

Table 36.3 Usage of SSL delay control between transfer frames (Master mode)

	SPCMDn. SSLKP bit	SPCR.BFDS bit	SSL delay control register*1 (RSPCK clock delay, SSL negation delay, next access delay)
1	0	0	Any given value. You can control each delay value according to setting for RSPCK clock delay, SSL negation delay and next access delay
2-1	1	0	
2-1	0	0	
3-1	1	1	Any given value. But delay is inserted only below. ● RSPCK clock delay of the 1st frame ● SSL negation delay and next access delay of the last frame
3-2	0	1	

Note 1. Whether the setting value of following bits are valid or not depends on the setting value of the SPCMD.SPNDEN bit. (See section 36.2.6. SPCMDm : SPI Command Register (m = 0 to 7).)
The SPDECR.SCKDL[2:0] bits: RSPCK delay
The SPDECR.SLNDL[2:0] bits: SSL negate delay
The SPDECR.SPNDL[2:0] bits: Next access delay

< Setting / operation example > (Motorola SPI, BFDS = 1 Case)

SPCMD0.SSLKP = 1 → Burst transfer / no interframe delay between 0 and 1 (SSL keep active)

SPCMD1.SSLKP = 1 → Burst transfer / no interframe delay between 1 and 2 (SSL keep active)

SPCMD2.SSLKP = 1 → Burst transfer / no interframe delay between 2 and 3 (SSL keep active))

SPCMD3.SSLKP = 1 → Burst transfer / no interframe delay between 3 and 4 (SSL keep active)

SPCMD4.SSLKP = 0 → do not Burst Transfer, and inactive SSL. (BFDS setting is invalid because it does not Burst Transfer.)

SPCMD5.SSLKP = 1 → Burst transfer / no interframe delay between 5 and 6 (SSL keep active)

SPCMD6.SSLKP = 1 → Burst transfer / no interframe delay between 6 and 7 (SSL keep active)

SPCMD7.SSLKP = 0 → do not Burst Transfer, and inactive SSL. (BFDS setting is invalid because it does not Burst Transfer.)

MODFEN bit (Mode Fault Error Detection Enable)

This bit is used to enable or disable detection of a mode fault error. (See section 36.3.10. Error Detection.) The SPI determines SSL0 pin input or output direction according to the combination of the MODFEN and MSTR bits. (See section 36.3.2. Controlling the SPI Pins.)

SPEIE bit (SPI Error Interrupt Enable)

This bit is used to enable or disable an SPI error interrupt request when the SPI detects a mode fault error or an underrun error and sets the MODF flag in the SPI status register (SPSR) to 1, when the SPI detects an overrun error and sets the OVRF flag in SPSR to 1, or when the SPI detects a parity error and sets the PERF flag in SPSR to 1. (See section 36.3.10. Error Detection)

SPRIE bit (SPI Receive Buffer Full Interrupt Enable)

This bit is used to enable or disable a receive buffer full interrupt request of the SPI.

SPIIE bit (SPI Idle Interrupt Enable)

This bit is used to enable or disable an idle interrupt request of the SPI after the SPI detects the idle state and sets the IDLNF flag in the SPI status register (SPSR) to 0.

SPDRES bit (SPI receive data ready error select)

When a receive data ready is detected (SPSR.SPDRF = 1), select whether to use SPIi_SPRI (i = 0, 1) interrupt request or SPIi_SPEI (i = 0, 1) interrupt request.

SPTIE bit (SPI Transmit Buffer Empty Interrupt Enable)

This bit is used to enable or disable a transmit buffer empty interrupt request of the SPI.

A transmit buffer empty interrupt request at the beginning of transmission is generated by setting the SPE bit to 1 simultaneously when or after the SPTIE bit is set to 1. Note that a transmit buffer empty interrupt is generated while the SPTIE bit is 1 even though SPI functions are disabled (SPE bit = 0).

CENDIE bit (SPI Communication End Interrupt Enable)

This bit controls generation of a communication end interrupt request.

SPMS bit (SPI Mode Select)

This bit is used to select SPI operation (4-wire) or clock synchronous operation (3-wire).

For clock synchronous operation, the SSL pin is not used but three pins RSPCK, MOSI, and MISO are used for communication. When SPMS = 1 (clock synchronous operation (3-wire)), the setting of the SPFRF bit is invalid.

To perform clock synchronous operation in master mode (SPCR.MSTR = 1), set the CPHA bit in the SPI command register (SPCMD) to 0 or 1. To perform clock synchronous operation in slave mode (SPCR.MSTR = 0), set the CPHA bit to 1. If this bit is set to 0 for clock synchronous operation in slave mode (SPCR.MSTR = 0), subsequent operation is not guaranteed.

The communication status according to the settings of the MSTR bit, TXMD[1:0] bits, SPFRF bit, and SPMS bit of the SPI control register (SPCR) as follows.

Table 36.4 SPI Communication Status

SPCR.MSTR	SPCR.TXMD[1]	SPCR.TXMD[0]	SPCR.SPFRF	SPCR.SPMS	Communication Status	Communication Status No
1	0	0	0	0	Transmit-Receive Master / Motorola SPI / SPI operation (4-wire)	1-(1)
1	0	0	1	0	Transmit-Receive Master / TI-SSP / SPI operation (4-wire)	1-(2)
1	0	0	—	1	Transmit-Receive Master /Clock synchronous operation (3-wire)	1-(3)
1	0	1	0	0	Transmit only Master / Motorola SPI / SPI operation (4-wire)	1-(4)
1	0	1	1	0	Transmit only Master / TI-SSP / SPI operation (4-wire)	1-(5)
1	0	1	—	1	Transmit only Master /Clock synchronous operation (3-wire)	1-(6)
1	1	—	0	0	Receive only Master / Motorola SPI / SPI operation (4-wire)	1-(7)
1	1	—	1	0	Receive only Master / TI-SSP / SPI operation (4-wire)	1-(8)
1	1	—	—	1	Receive only Master /Clock synchronous operation (3-wire)	1-(9)
0	0	0	0	0	Transmit-Receive Slave / Motorola SPI / SPI operation (4-wire) (default)	0-(1)
0	0	0	1	0	Transmit-Receive Slave / TI-SSP / SPI operation (4-wire)	0-(2)
0	0	0	—	1	Transmit-Receive Slave /Clock synchronous operation (3-wire)	0-(3)
0	0	1	0	0	Transmit only Slave / Motorola SPI / SPI operation (4-wire)	0-(4)
0	0	1	1	0	Transmit only Slave / TI-SSP / SPI operation (4-wire)	0-(5)
0	0	1	—	1	Transmit only Slave /Clock synchronous operation (3-wire)	0-(6)
0	1	—	0	0	Receive only Slave / Motorola SPI / SPI operation (4-wire)	0-(7)
0	1	—	1	0	Receive only Slave / TI-SSP / SPI operation (4-wire)	0-(8)
0	1	—	—	1	Receive only Slave /Clock synchronous operation (3-wire)	0-(9)

SPFRF bit (SPI Frame Format Select)

This bit selects the communication protocol.

The format of the SPI terminal (RSPCK, SSL0 to 7) can be set according to the set communication protocol.

When SPMS = 1 (clock synchronous operation (3-wire)), this bit is invalid because SSL is not used.

TXMD[1:0] bit (Communication Mode Select)

This bit is used to select the transmit-receive, transmit-only, and receive-only serial communication.

When TXMD[1:0] is set to 01 for communication, transmit-only is performed without reception.

When TXMD[1] is set to 1 for communication, receive-only is performed without transmission.

When TXMD[1:0] is set to 01 for communication, a receive buffer full interrupt request cannot be used.

When TXMD[1] is set to 1 for communication, a transmit buffer empty interrupt request cannot be used.

(See section 36.3.6. Communication Operating Mode.)

MSTR bit (SPI Master/Slave Mode Select)

This bit is used to select master mode or slave mode of the SPI. The SPI determines input/output directions of pins RSPCK, MOSI, MISO, and SSL1 to SSL3 according to the MSTR bit setting.

BPEN bit (Synchronization Circuit Bypass Enable)

This bit selects whether to enable or disable the synchronization bypass function. Set this bit to 1 when the bus clock (PCLK) is also used as operation clock (TCLK), see Figure 36.2.

36.2.4 SPCR2 : SPI Control Register 2

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SPDRC[7:0]								RMSTTG	RMEDTG	—	RMFM[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	RMFM[4:0]	Frame processing count setting in Master Receive only The number of received frames can be adjusted in master receive only. 0x00: This function is not used*1 0x01: Automatically stop communication after processing 1 received frame ⋮ 0x1F: Automatically stop communication after processing 31 received frames	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	RMEDTG	End Trigger in Master Receive only 1: Receive End (Writable only when Master Receive only) Reading value is always 0	W
7	RMSTTG	Start Trigger in Master Receive only 1: Receive Start (Writable only when Master Receive only) Reading value is always 0	W
15:8	SPDRC[7:0]	SPI received data ready detect adjustment 0x00: Disable receive data ready detection function 0x01: Performs reception data ready judgment after 1 TCLK ⋮ 0xFF: Performs reception data ready judgment after 255 TCLK	R/W
16	SPLP	SPI Loopback 0: Normal mode 1: Loopback mode (inverted transmit data = receive data)	R/W
17	SPLP2	SPI Loopback 2 0: Normal mode 1: Loopback mode (transmit data = receive data)	R/W

Bit	Symbol	Function	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
20	MOIFV	MOSI Idle Fixed Value 0: The fixed value of MOSI idle = 0. 1: The fixed value of MOSI idle = 1.	R/W
21	MOIFE	MOSI Idle Fixed Value Enable 0: The MOSI output value is the last data of previous transfer. 1: The MOSI output value is the set MOIFV bit value.	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. See SW flow in the section 36.3.12.1. Master mode operation.

RMFM[4:0] bit (Frame processing count setting in Master Receive only)

The number of received frames can be adjusted when operating in master receive only. Valid only when the master mode (SPCR.MSTR = 1) and the communication operation mode select bits (SPCR.TXMD [1:0]) are 10b.

Only the start bit in master mode reception automatically stops communication after starts frame processing according to the value set in this bit after reception starts.

If the RMFM [4:0] bits are rewritten while the SPE bit of the SPI control register (SPCR) is 1, subsequent operations are not guaranteed.

RMEDTG bit (End Trigger in Master Receive only)

This bit is used to end reception when master receive only. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.TXMD [1:0]) are 10b.

RMSTTG bit (Start Trigger in Master Receive only)

This bit is used to start reception when master receive only. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.TXMD [1:0]) are 10b.

Writing 1 to this bit during reception is not accepted. Write again after reception is completed.

SPDRC[7:0] bit (SPI received data ready detect adjustment)

The receive data ready detection function can be disabled or, if used, the period until detection can be set from 1 to 255 TCLK.

The value set in the SPDRC [7:0] bits is used to 1 set the SPDRF flag. For details, see the description of SPDRF in section 36.2.9. SPSR : SPI Status Register.

If the set value is changed while the SPE bit is 1, subsequent operations are not guaranteed.

SPLP bit (SPI Loopback)

When the SPLP bit is set to 1, the SPI shuts down the route between the MISO pin and the shift register (when the MSTR bit in the SPI control register is 1) or shuts down the route between the MOSI pin and the shift register, inverts the input route value in the shift register, and then connects the route to the output route (when the MSTR bit in the SPI control register is 0) (loopback mode).

SPLP2 bit (SPI Loopback 2)

When the SPLP2 bit is set to 1, the SPI shuts down the route between the MISO pin and the shift register (when the MSTR bit in the SPI control register is 1) or shuts down the route between MOSI pin and the shift register and then connects the route to the output route without inverting the input route value in the shift register (when the MSTR bit in the SPI control register is 0) (loopback mode). If this bit is set to 1 together with the SPLP bit, setting this bit takes precedence.

MOIFV bit (MOSI Idle Fixed Value)

This bit is used to select the MOSI pin output value during the SSL negation period (including SSL retention period in burst transfer) when the MOIFE bit is 1 in master mode.

If this bit is modified with the SPE bit in the SPI control register (SPCR) set to 1, subsequent operation is not guaranteed.

MOIFE bit (MOSI Idle Fixed Value Enable)

This bit is used for the SPI in master mode to fix the MOSI output value during the SSL negation period (including SSL retention period in burst transfer). When MOIFE bit = 0, the SPI outputs the last data of the previous serial transfer to MOSI during the SSL negation period. When MOIFE bit = 1, the SPI outputs the fixed MOIFV bit value to MOSI.

If this bit is modified with the SPE bit in the SPI control register (SPCR) set to 1, subsequent operation is not guaranteed.

36.2.5 SPCR3 : SPI Control Register 3

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	SPSLN[2:0]			—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SPBR[7:0]								—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SSL0P	SSL0 Signal Polarity [In the Motorola-SPI case] 0: The SSL0 signal is active low. 1: The SSL0 signal is active high. [In the TI-SSP case] 0: The SSL0 signal is active high. 1: The SSL0 signal is active low.	R/W
1	SSL1P	SSL1 Signal Polarity [In the Motorola-SPI case] 0: The SSL1 signal is active low. 1: The SSL1 signal is active high. [In the TI-SSP case] 0: The SSL1 signal is active high. 1: The SSL1 signal is active low.	R/W
2	SSL2P	SSL2 Signal Polarity [In the Motorola-SPI case] 0: The SSL2 signal is active low. 1: The SSL2 signal is active high. [In the TI-SSP case] 0: The SSL2 signal is active high. 1: The SSL2 signal is active low.	R/W
3	SSL3P	SSL3 Signal Polarity [In the Motorola-SPI case] 0: The SSL3 signal is active low. 1: The SSL3 signal is active high. [In the TI-SSP case] 0: The SSL3 signal is active high. 1: The SSL3 signal is active low.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
15:8	SPBR[7:0]	SPI Bit Rate	R/W
23:16	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
26:24	SPSLN[2:0]	SPI Sequence Length 0 0 0: Sequence Length is 1 (Referenced SPCMDn, n = 0→0→…) 0 0 1: Sequence Length is 2 (Referenced SPCMDn, n = 0→1→0→…) 0 1 0: Sequence Length is 3 (Referenced SPCMDn, n = 0→1→2→0→…) 0 1 1: Sequence Length is 4 (Referenced SPCMDn, n = 0→1→2→3→0→…) 1 0 0: Sequence Length is 5 (Referenced SPCMDn, n = 0→1→2→3→4→0→…) 1 0 1: Sequence Length is 6 (Referenced SPCMDn, n = 0→1→2→3→4→5→0→…) 1 1 0: Sequence Length is 7 (Referenced SPCMDn, n = 0→1→2→3→4→5→6→0→…) 1 1 1: Sequence Length is 8 (Referenced SPCMDn, n = 0→1→2→3→4→5→6→7→0→…)	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

SSLiP bits (SSL Signal Polarity Bits)

These bits are used to specify the polarity of SSL signals. The set SSLiP bit (i = 0 to 3) values indicate the active polarity of SSLi signals.

If any of these SSLiP bits is modified with the SPE bit in the SPI control register (SPCR) set to 1, subsequent operation is not guaranteed.

Note: SSL0 is different from SSL1-SSL3. When slave or multi-master, it functions as an input.

For details, see section 36.3.3.2. Single-master/single-slave with the MCU as a slave, and section 36.3.3.5. Multi-master/multi-slave with the MCU as a master.

SPBR[7:0] bit (SPI Bit Rate)

The SPI bit rate bits (SPBR) is used to set the bit rate in master mode. If SPBR is modified while the MSTR bit in the SPI control register (SPCR) is 1, subsequent operation is not guaranteed.

When the SPI is used in slave mode, the bit rate depends on the input clock bit rate regardless of the SPCMD.BRDV setting. (Specify a bit rate that meets electrical characteristics.)

The bit rate is determined by a combination of the set SPBR value and the set BRDV[1:0] bits value in the SPI command register (SPCMD0 to SPCMD7).

The bit rate is calculated by the following expression, where n is the set SPBR value (0 to 255) and N is the set BRDV[1:0] bits value (0 to 3).

$$\text{Bit rate} = \frac{f(\text{TCLK})}{2 \times (n+1) \times 2^N}$$

The following table shows an example of correspondence between bit rates and set values of SPBR and BRDV[1:0].

Table 36.5 Corresponding Between Bit Rates and Set Values (Example)

SPBR Value (n)	BRDV Value (N)	Division Ratio	Bit Rate				
			TCLK = 32 MHz	TCLK = 36 MHz	TCLK = 40 MHz	TCLK = 50 MHz	TCLK = 120MHz
0	0	2	16.0 Mbps	18.0 Mbps	20.0 Mbps	25.0 Mbps	60.0Mbps
1	0	4	8.00 Mbps	9.00 Mbps	10.0 Mbps	12.5 Mbps	30.0Mbps
2	0	6	5.33 Mbps	6.00 Mbps	6.67 Mbps	8.33 Mbps	20.0Mbps
3	0	8	4.00 Mbps	4.50 Mbps	5.00 Mbps	6.25 Mbps	15.0Mbps
4	0	10	3.20 Mbps	3.60 Mbps	4.00 Mbps	5.00 Mbps	12.0Mbps
5	0	12	2.67 Mbps	3.00 Mbps	3.33 Mbps	4.16 Mbps	10.0Mbps
5	1	24	1.33 Mbps	1.50 Mbps	1.67 Mbps	2.08 Mbps	5.0Mbps
5	2	48	677 kbps	750 kbps	833 kbps	1.04 Mbps	2.5Mbps
5	3	96	333 kbps	375 kbps	417 kbps	521 kbps	1.25Mbps
255	3	4096	7.81 kbps	8.80 kbps	9.78 kbps	12.2 kbps	29.3kbps

SPSLN[2:0] bit (SPI Sequence Length)

These bits are used to set the sequence length for the SPI in master mode to perform sequence operation. According to the sequence length specified by SPSLN[2:0] bits, the SPI in master mode changes SPI command registers 0 to 7 (SPCMD0 to SPCMD7) to be referenced and the reference sequence. For details, see section 36.3.13.1. Master mode operation.

The SPI in slave mode always references SPCMD0.

36.2.6 SPCMDm : SPI Command Register (m = 0 to 7)

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x14 + 0x04 × m

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	SSLA[2:0]			—	—	—	SPB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SCKDEN	SLNDEN	SPNDEN	LSBF	—	—	—	—	SSLKP	—	—	—	BRDV[1:0]		CPOL	CPHA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPHA	RSPCK Phase 0: Data is sampled at an odd edge and changes at an even edge. 1: Data changes at an odd edge and is sampled at an even edge.	R/W
1	CPOL	RSPCK Polarity 0: RSPCK in idle state is 0. 1: RSPCK in idle state is 1.	R/W
3:2	BRDV[1:0]	Bit Rate Division 0 0: Base bit rate 0 1: Base bit rate divided by 2 1 0: Base bit rate divided by 4 1 1: Base bit rate divided by 8	R/W
6:4	—	These bits are read as 0. The write value should be 0.	R/W
7	SSLKP	SSL Signal Level Hold 0: All SSL signals are negated at the end of transfer. 1: SSL signal level is held after the transfer ends until the next access starts.	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
12	LSBF	SPI LSB First 0: MSB first 1: LSB first	R/W
13	SPNDEN	SPI Next-Access Delay Enable 0: Next-access delay is 1RSPCK + 5TCLK 1: Next-access delay is the set value of the SPI next-access delay (SPDECR.SPNDL).	R/W
14	SLNDEN	SSL Negation Delay Setting Enable 0: [Master] SSL negation delay is 1RSPCK. [Slave in the TI-SSP] SSL negation delay is 1TCLK 1: SSL negation delay is the set value of the slave select negation delay (SPDECR.SLNDL).	R/W

Bit	Symbol	Function	R/W
15	SCKDEN	RSPCK Delay Setting Enable [In the Motorola-SPI case] 0: RSPCK delay is 1 RSPCK. 1: RSPCK delay is the set value of the RSPCK delay (SPDECR.SCKDL). [In the TI-SSP case] 0: RSPCK delay is 0 RSPCK. 1: RSPCK delay is the set value of the RSPCK delay (SPDECR.SCKDL).	R/W
20:16	SPB[4:0]	SPI Data Length 0x00 to 0x02: Setting prohibited 0x03: 4 bits 0x04: 5 bits 0x05: 6 bits ⋮ 0x1E: 31 bits 0x1F: 32 bits	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
26:24	SSLA[2:0]	SSL Signal Assertion 0 0 0: SSL0 0 0 1: SSL1 0 1 0: SSL2 0 1 1: SSL3 Others: Setting prohibited	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The SPI has eight SPI command registers (SPCMD0 to SPCMD7) that are used to set the transfer format of the SPI in master mode. Furthermore, some bits in SPCMD0 are used to set the transfer format of the SPI in slave mode. The SPI in master mode sequentially references SPCMD0 to SPCMD7 according to the setting of the SPSLN[2:0] bits in the SPI Control register 3 (SPCR3), and then performs serial transfer specified in the referenced SPCMD.

SPI set the SPCMD register before setting data to be transmitted by referencing the SPCMD when the transmit buffer is empty (while the next-transfer data has not been set).

The SPCMD referenced by the SPI in master mode is indicated by SPCP[2:0] in the SPI status register (SPSR). If SPCMD0 is modified while the SPI in slave mode is enabled (SPCR.SPE = 1), subsequent operation is not guaranteed.

CPHA bit (RSPCK Phase)

This bit is used to set the RSPCK phase of the SPI in master mode or slave mode. To perform data communication between SPI modules, the same RSPCK phase must be set for both modules.

When SPCR.SPMS = 0 and SPCR.SPFRF = 1 (in TI SSP mode), setting CPHA = 0 is invalid.

CPOL bit (RSPCK Polarity)

This bit is used to set the RSPCK polarity of the SPI in master mode or slave mode. To perform data communication between SPI modules, the same RSPCK polarity must be set for both modules.

BRDV[1:0] bit (Bit Rate Division)

This register is used to determine the bit rate with a combination of the set values of the BRDV[1:0] bits and the SPI bit rate register (SPCR3.SPBR). The set SPBR value determines the base bit rate. The set BRDV[1:0] bits value is used to select undivided, 2-divided, 4-divided, or 8-divided base bit rate. SPCMD0 to SPCMD7 enable setting of different BRDV[1:0] values. This makes it possible to perform serial transfer with a different bit rate for each command.

SSLKP bit (SSL Signal Level Hold)

This bit is used to set whether to hold or negate the SSL signal level of the current command during a period from the SSL negation timing for the current command to the SSL assertion timing for the next command when the SPI in master mode performs serial transfer. Setting this bit to 1 enables burst transfer in SPI operation master mode. For details, see section 36.3.12.1. Master mode operation.

To use the SPI in slave mode, set SSLKP bit to 0.

LSBF bit (SPI LSB First)

This bit is used to set the data format of the SPI in master mode or slave mode to MSB first or LSB first.

SPNDEN bit (SPI Next-Access Delay Enable)

This bit is used to set the period (next-access delay) after the SPI in master mode inactivates the SSL signal at the end of serial transfer until it enables SSL signal assertion of the next access. When SPNDEN bit = 0, the SPI sets the next-access delay to 1 RSPCK + 5TCLK. When SPNDEN bit = 1, the SPI inserts the next-access delay in accordance with the SPI next-access delay register (SPDECR.SPNDL) setting.

To use the SPI in slave mode, set SPNDEN bit to 0.

SLNDEN bit (SSL Negation Delay Setting Enable)

[In the Motorola-SPI case]

This bit is used to set the period (SSL negation delay) after the SPI in master mode stops RSPCK oscillation until it inactivates the SSL signal. When SLNDEN bit = 0, the SPI sets the SSL negation delay to 1 RSPCK. When SLNDEN bit = 1, the SPI negates the SSL signal with the RSPCK delay in accordance with the slave select negation delay register (SPDECR.SLNDL) setting.

To use the SPI in slave mode, set SLNDEN bit to 0.

[In the TI-SSP case]

This bit is used to set the period from when the master mode SPI stops RSPCK oscillation to when the OE signal is inactivated, or when the slave mode SPI detects the last edge of RSPCK and then negates the OE signal. When the SLNDEN bit is 0, the SSL negate delay is 1 RSPCK in master mode and 1 TCLK in slave mode. When SLNDEN bit = 1, the SPI negates the SSL signal with the RSPCK delay in accordance with the slave select negation delay register (SPDECR.SLNDL) setting.

When using SPI in slave mode except TI SSP setting, set the SLNDEN bit to 0.

SCKDEN bit (RSPCK Delay Setting Enable)

[In the Motorola-SPI case]

This bit is used to set the period (RSPCK delay) after the SPI in master mode activates the SSL signal until it oscillates RSPCK. When SCKDEN bit = 0, the SPI sets the RSPCK delay to 1 RSPCK. When SCKDEN bit = 1, the SPI starts RSPCK oscillation with the RSPCK delay in accordance with the RSPCK delay register (SPDECR.SCKDL) setting.

To use the SPI in slave mode, set SCKDEN bit to 0.

[In the TI-SSP case]

This bit is used to set the period from the start of assertion of the SSL signal to the RSPCK oscillation (RSPCK delay) and the period of the SSL signal to negation by the SPI in master mode. When SCKDEN bit = 0, the SPI does not sets the RSPCK delay. When SCKDEN bit = 1, the SPI starts RSPCK oscillation with the RSPCK delay in accordance with the RSPCK delay register (SPDECR.SPCKDL) setting.

To use the SPI in slave mode, set SCKDEN bit to 0.

SPB[4:0] bit (SPI Data Length)

These bits are used to set the transfer data length of the SPI in master mode or slave mode.

SSLA[2:0] bit (SSL Signal Assertion)

These bits are used to control SSL signal assertion for the SPI in master mode to perform serial transfer. The set SSLA[2:0] bits value controls assertion of the SSL3 to SSL0 signals. The signal polarity when the SSL signal is asserted depends on the set value of the SPI slave select polarity register (SPCR3.SSLiP). When SSLA[2:0] bits are set to 000b in multi-master mode, serial transfer is performed with all SSL signals negated (because SSL0 is input).

To use the SPI in slave mode, set SSLA[2:0] bits to 000b.

36.2.7 SPDCR : SPI Data Control Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SPFC[1:0]		—	—	—	SINV	SPRDTD	—	—	BYSW
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BYSW	Byte Swap Operating Mode Select 0: Byte Swap OFF 1: Byte Swap ON	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	SPRDTD	SPI Receive Data or Transmit Data Select 0: The SPDR reads the receive buffer. 1: The SPDR reads the transmit buffer	R/W
4	SINV	Serial data invert bit 0: Not invert serial data 1: Invert serial data.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
9:8	SPFC[1:0]	Frame Count 0 0: 1 frame 0 1: 2 frames 1 0: 3 frames 1 1: 4 frames	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The SPI data control register (SPDCR) controls the data format.

If the value set in this register is changed while the SPE bit is 1, subsequent operations are not guaranteed.

BYSW bit (Byte Swap Operating Mode Select)

It is a setting bit, that is to swap a transmit/receive data in byte units. A data after byte swap is different by a data length (setting of SPCMD.SPB[4:0]).

When byte swap, A data length (setting of SPB[4:0]) must be set to 32bit or 16bit. Other case of data length (i.e. 4 to 15, 17 to 31-bit length), byte swap is not guaranteed. For the arrangement of data before and after swapping data lengths of 32 bits and 16 bits, see section 36.3.4.3. Byte Swap Transmission and section 36.3.4.4. Byte Swap Reception.

When the parity function set to valid, the behavior is not guaranteed.

SPRDTD bit (SPI Receive Data or Transmit Data Select)

This bit is used to select receive buffer or transmit buffer from which the SPI data register (SPDR) value is read.

When the transmit buffer is read, the value that was written to SPDR immediately before is read.

SINV bit (Serial data invert bit)

This bit is used to invert transmit data and receive data.

When the SINV bit is set to 1, transmit buffer (SPTX) data is inverted to invert transmit data and receive data, and then the inverted data is stored in the receive buffer (SPRX). The parity bit is the value corresponding to the inverted transmission/reception data.

SPFC[1:0] bit (Frame Count)

Used for the condition to set the CENDF flag in slave receive only mode.

For details on the CENDF flag setting conditions, see section 36.2.9. SPSR : SPI Status Register.

Note that this bit is invalid except in the slave receive only mode.

36.2.8 SPDCR2 : SPI Data Control Register 2

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x44

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TTRG[1:0]		—	—	—	—	—	—	RTRG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RTRG[1:0]	Receive FIFO threshold setting 0 0: threshold 0 0 1: threshold 1 1 0: threshold 2 1 1: threshold 3	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	TTRG[1:0]	Transmission FIFO threshold setting 0 0: threshold 0 0 1: threshold 1 1 0: threshold 2 1 1: threshold 3	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

SPI data control register 2 (SPDCR2) controls the FIFO threshold. If the value set in this register is changed while the SPE bit is 1, subsequent operations are not guaranteed.

RTRG[1:0] bit (Receive FIFO threshold setting)

Set the receive FIFO threshold.

When the number of data stored in the receive FIFO > the number of frames set by RTRG[1:0], the receive buffer full flag is set.

TTRG[1:0] bit (Transmission FIFO threshold setting)

Set the transmit FIFO threshold.

When the number of empty stages in the transmit FIFO > the number of frames set in TTRG[1:0], the transmit buffer empty flag is set.

36.2.9 SPSR : SPI Status Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SPRF	CENDF	SPTEF	UDRF	PERF	MODF	IDLNF	OVRF	SPDRF	—	—	—	—	—	—	—
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	SPECM[2:0]			—	SPCP[2:0]			—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R
10:8	SPCP[2:0]	SPI Command Pointer 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
11	—	This bit is read as 0. The write value should be 0.	R
14:12	SPECM[2:0]	SPI Error Command 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
22:15	—	These bits are read as 0. The write value should be 0.	R
23	SPDRF	SPI Receive Data Ready Flag 0: Receive data ready not detected 1: Receive data ready detected	R
24	OVRF	Overrun Error Flag 0: No overrun error is present. 1: An overrun error is present.	R
25	IDLNF	SPI Idle Flag 0: The SPI is in the idle state. 1: The SPI is in the transfer state.	R
26	MODF	Mode Fault Error Flag 0: Neither mode fault error nor underrun error is present. 1: A mode fault error or underrun error is present.	R
27	PERF	Parity Error Flag 0: No parity error is present. 1: A parity error is present.	R
28	UDRF	Underrun Error Flag 0: When MODF=0, neither mode fault error nor underrun error is present. When MODF=1, a mode fault error is present. 1: When MODF=0, neither mode fault error nor underrun error is present. When MODF=1, an underrun error is present.	R

Bit	Symbol	Function	R/W
29	SPTEF	SPI Transmit Buffer Empty Flag 0: The number of empty stages in the transmit FIFO ≤ the value set in SPDCR2.TTRG 1: The number of empty stages in the transmit FIFO > the value set in SPDCR2.TTRG	R
30	CENDF	Communication End Flag 0: The SPI is not communicating or communicating. 1: The SPI communication completed.	R
31	SPRF	SPI Receive Buffer Full Flag 0: The number of data stored in the receive FIFO ≤ number of frames set by the SPDCR2.RTRG bit. 1: The number of data stored in the receive FIFO > number of frames set by the SPDCR2.RTRG bit.	R

Note: S-TYPE-3, P-TYPE-3

The SPI status register (SPSR) stores flags that indicate SPI's operating status.

SPCP[2:0] bit (SPI Command Pointer)

These bits indicate SPI command registers 0 to 7 (SPCMD0 to SPCMD7) indicated by the current pointer in the SPI sequence control. For details about the SPI sequence control, see section 36.3.13.1. Master mode operation.

SPECM[2:0] bit (SPI Error Command)

These bits indicate SPI command registers 0 to 7 (SPCMD0 to SPCMD7) indicated by the command pointer (SPCP[2:0] bits) when an error was detected in the SPI sequence control. The SPI updates the SPECM[2:0] bits value only when an error is detected. When no error is present (OVRF, MODF, and PERF flags in SPSR are 0), the SPECM[2:0] bits value has no meaning. For the SPI's error detection function, see section 36.3.10. Error Detection. For the SPI's sequence control, see section 36.3.13.1. Master mode operation.

SPDRF bit (SPI Receive Data Ready Flag)

During communication (SPCR.SPE = 1), a certain period of time has elapsed while the number of data stored in the reception FIFO ≤ the reception FIFO threshold.

This bit is set to 0 when the reception operation is not performed (SPCR.TXMD[1:0] = 01b).

[Setting condition]

All the following two conditions are met.

- SPCR2.SPDRC[7:0] ≠ 0x00.
- After the receive FIFO has been written, when the number of data stored in the receive FIFO ≤ the receive FIFO threshold and the value set by SPDRC[7:0] has elapsed

[Clearing condition]

- When 1 is written to the SPSRC.SPDRFC bit.

OVRF bit (Overrun Error Flag)

This flag indicates whether an overrun error is present. When the RSPCK clock auto-stop function is enabled (SPCR.SCKASE = 1) in master mode (SPCR.MSTR = 1), no overrun error occurs and, therefore, this flag is not set to 1. For details, see section 36.3.10.1. Overrun errors.

[Setting condition]

When serial transfer is completed in one of the following two conditions with data stored in the receive FIFO for the number of FIFO stages.

- SPCR.TXMD[1:0] = 00b. (transmit-receive mode)
- SPCR.TXMD[1:0] = 10b. (receive only)

[Clearing condition]

- When 1 is written to the SPSRC.OVRFC bit.

IDLNF bit (SPI Idle Flag)

This flag indicates transfer status of the SPI.

[Setting condition]

[Transmit-Receive, Transmit-only Master mode]

- None of the clearing conditions (Transmit-Receive / Transmit-only in Master mode) below is met.

[Receive-only Master mode]

- When 1 is written to RMSTTG of SPCR2.

[Slave mode]

- The SPE bit in SPCR is 1 (SPI function enabled).

[Clearing conditions]

Communication status: 1-(1) to (6) * For details of communication status, see Table 36.4.

[Transmit-Receive, Transmit-only Master mode]

Any of the following two conditions is met.

- The SPE bit in SPCR is 0 (SPI initialization).
- All the following three conditions are met.
  - The next transfer data is not set in the transmission buffer (SPTXn, n = 0 to 3)
  - The SPCP bits in SPSR are 000b (at the beginning of sequence control).
  - The operation completed by the next access delay (the master main state machine has transitioned to the idle state)

[Receive-only Master mode]

Communication status: 1-(7) to (9)

Any of the following two conditions is met.

- The SPE bit in SPCR is 0 (SPI initialization).
- Any of the following 3 conditions is met.
  When RMFM[4:0] = 0x00, after writing 1 to RMEDTG, the operation completed by the next access delay (the master main state machine has transitioned to the idle state)
  When RMFM[4:0] ≠ 0x00, after writing 1 to RMEDTG, the operation completed by the next access delay (the master main state machine has transitioned to the idle state)
  When RMFM[4:0] ≠ 0x00, the operation completed by the next access delay after processing is completed for the number of received frames set in RMFM[4:0] (the master main state machine has transitioned to the idle state)

[Slave mode]

Communication status: 0-(1) to (9)

- The SPE bit in SPCR is 0 (SPI initialization).

MODF bit (Mode Fault Error Flag)

This flag indicates whether a mode fault error or an underrun error is present. The UDRF flag allows you to see which error (mode fault error or underrun error) has occurred.

[Setting condition]

[Multi-master mode]

- The SSL0 pin input level becomes active level while the SPCR.MSTR bit = 1 (master mode) and the SPCR.MODFEN bit = 1 (mode fault error detection enabled), and then the SPI has detected a mode fault error.

[Slave, Motorola-SPI mode]

Any of the following two conditions is met.

- The SSL0 pin is negated before the RSPCK cycles necessary for data transfer end while the SPCR.MSTR bit = 0 (slave mode), SPCR.SPFRF bit = 0 (Motorola-SPI) and the SPCR.MODFEN bit = 1 (mode fault error detection enabled), and then the SPI has detected a mode fault error.
- Serial transfer is started before transmit data output becomes ready while the SPCR.SPE bit = 1 (SPI function enabled), and then the SPI has detected an underrun error.

[Slave, TI-SSP mode]

Any of the following two conditions is met.

- The SSL0 pin is asserted before the RSPCK cycles necessary for data transfer end while the SPCR.MSTR bit = 0 (slave mode), SPCR.SPFRF bit = 1 (TI-SSP) and the SPCR.MODFEN bit = 1 (mode fault error detection enabled), and then the SPI has detected a mode fault error.
- Serial transfer is started before transmit data output becomes ready while the SPCR.SPE bit = 1 (SPI function enabled), and then the SPI has detected an underrun error.

The SSL0 signal active level depends on the SPCR3.SSLiP bits (SSL signal polarity bits).

[Clearing condition]

- When 1 is written to the SPSRC.MODFC bit.

PERF bit (Parity Error Flag)

This flag indicates whether a parity error is present.

[Setting condition]

When the serial transfer ends and a parity error is detected with the SPPE bit of SPCR set to 1 under any of the following 2 conditions.

- SPCR.TXMD[1:0] = 00b. (transmit-receive master mode or transmit-receive slave mode)
- SPCR.TXMD[1:0] = 10b. (receive-only master mode or receive-only slave mode)

[Clearing condition]

- When 1 is written to the SPSRC.PERFC bit.

UDRF bit (Underrun Error Flag)

This flag indicates that a mode fault error or an underrun error is present.

[Setting condition]

- Serial transfer is started before transmit data output becomes ready while the SPCR.MSTR bit = 0 and the SPCR.TXMD[1:0] bit = 00b or 01b (transmit-receive slave mode or transmit-only slave mode) and the SPCR.SPE bit = 1 (SPI function enabled), and then the SPI has detected an underrun error.

[Clearing condition]

- When 1 is written to the SPSRC.UDRFC bit.

SPTEF bit (SPI Transmit Buffer Empty Flag)

This flag indicates the transmit buffer (SPTX) status in the SPI data register (SPDR).

[Setting condition]

Any of the following 3 conditions is met.

- The SPE bit is set to 0 (SPI initialization).
- When the number of empty transmission FIFO stages > the value set in SPDCR2.TTRG[1:0].
- When 1 is written to SPFCR.SPFRST.

[Clearing condition]

Any of the following two conditions is met.

- At the time of final access when transmission data is written to SPDR (SPTXn, n = 0 to 3) in one processing routine using DTC / DMAC.
- When 1 is written to the SPSRC.SPTEFC bit.

Writing a value to the SPDR register is enabled only while the SPTEF flag = 1. If a value is written to the SPDR register while the SPTEF flag = 0, transmit buffer data is not updated.

CENDF bit (Communication End Flag)

This flag indicates communication end status of SPI. It turns 1 at communication end and turns 0 at starting next communication.

[Setting condition]

[Transmit-Receive / Transmit-only Master mode]

Communication status: 1-(1) to (6) * For details of communication status, see Table 36.4.

The following 3 conditions are met.

- The next transfer data is not set in the transmission buffer (SPTXn, n = 0 to 3)
- The SPSR.SPCP[2:0] are 000b. (It means the head of the sequential control.)
- The operation completed by the next access delay (the master main state machine has transitioned to the idle state)

[Receive-only Master mode]

Communication status: 1-(7) to (9)

Any of the following 3 conditions is met.

- When RMFM[4:0] = 0x00, after writing 1 to RMEDTG, the operation completed by the next access delay (the master main state machine has transitioned to the idle state)
- When RMFM[4:0] ≠ 0x00, after writing 1 to RMEDTG, the operation completed by the next access delay (the master main state machine has transitioned to the idle state)
- When RMFM[4:0] ≠ 0x00, the operation completed by the next access delay after processing is completed for the number of received frames set in RMFM[4:0] (the master main state machine has transitioned to the idle state)

[Transmit-receive / transmit-only slave, Motorola-SPI mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0)]

Communication status: 0-(1), (4)

The following 3 conditions are met.

- The next transfer data is not set in the transmission buffer
- The transmission shift register is empty. (It means SPI does not do serial transfer.)
- SSL0 was negated.

[Transmit-receive / transmit-only slave, TI-SSP mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0)]

Communication status: 0-(2), (5)

The following 3 conditions are met.

- The next transfer data is not set in the transmission buffer
- The transmission shift register is empty. (It means SPI does not do serial transfer.)
- When the SSL0 negate delay is completed.

[Transmit-receive / transmit only slave mode at clock synchronous (3-wire: the SPCR.SPMS bit is 1)]

Communication status: 0-(3), (6)

The following 3 conditions are met.

- The next transfer data is not set in the transmission buffer
- The transmission shift register is empty. (It means SPI does not do serial transfer.)
- The last even edge of RSPCK of the frame was detected. (When the SPCMD.CPHA bit is 1.)

[Receive only slave, Motorola-SPI mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0)]

Communication status: 0-(7)

The following condition is met.

- SSL0 input was negated after getting frames for SPDCR.SPFC set value in the receive buffer.

[Receive only slave, TI-SSP mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0)]

Communication status: 0-(8)

The following condition is met.

- SSL0 negate delay is completed after getting frames for SPDCR.SPFC set value in the receive buffer.

[Receive only slave mode at clock synchronous (3-wire: the SPCR.SPMS bit is 1)]

Communication status: 0-(9)

The following condition is met.

- The last even edge of RSPCK of the Last frame received for SPFC sets value. (When the SPCMD.CPHA bit is 1.)

[Clearing condition]

[Transmit-Receive / Transmit-only Master mode]

Communication status: 1-(1) to (6)

Any of the following 2 conditions is met.

- The next transmit data was written to the transmit buffer (SPTX).
- When 1 is written to the SPSRC.CENDFC bit.

[Receive -only Master mode]

Communication status: 1-(7) to (9)

Any of the following 2 conditions is met.

- When 1 is written to the SPCR2.RMSTTG bit with SPE = 1.
- When 1 is written to the SPSRC.CENDFC bit.

[Transmit-receive / transmit only slave mode]

Communication status: 0-(1) to (6)

Satisfy one of following.

- The next transmit data was written to the transmit buffer (SPTX).
- When 1 is written to the SPSRC.CENDFC bit.

[Receive only slave mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0)]

Communication status: 0-(7) to (8)

Satisfy one of following.

- SSL0 assertion of next data was detected.
- When 1 is written to the SPSRC.CENDFC bit.

[Receive only slave mode at clock synchronous (3-wire: the SPCR.SPMS bit is 1)]

Communication status: 0-(9)

Satisfy one of following.

- The first edge of RSPCK of the next data was detected.
- When 1 is written to the SPSRC.CENDFC bit.

SPRF bit (SPI Receive Buffer Full Flag)

This flag indicates the receive buffer (SPRX) status in the SPI data register (SPDR).

[Setting condition]

When the number of data stored in the receive FIFO > the number of frames set in the SPDCR2.RTRG[1:0] bits in Transmit-Receive, receive-only mode. However, the SPRF flag does not change from 0 to 1 while the OVRF flag = 1. (See section 36.3.10. Error Detection.)

[Clearing condition]

Any of the following 3 conditions is met.

- At the last access when read data is read from SPDR (SPRXn, n = 0 to 3) in one processing routine using DTC / DMAC
- When 1 is written to the SPSRC.SPRFC bit
- When 1 is written to the SPFCR.SPFRST bit

36.2.10 SPTFSR : SPI Transfer FIFO Status Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x58

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	TFDN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TFDN[2:0]	Transmit FIFO data empty stage number 0 0 0: Number of empty stages 0 ⋮ 1 0 0: Number of empty stages 4	R
31:3	—	These bits are read as 0. The write value should be 0.	R

TFDN[2:0] bit (Transmit FIFO data empty stage number)

Displays the number of empty transmission FIFO stages. By clearing the SPCR.SPE bit, TFDN[2:0] will be the initial value after reset (= all empty).

36.2.11 SPRFSR : SPI Receive FIFO Status Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x5C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	RFDN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
2:0	RFDN[2:0]	Receive FIFO data store stage number 0 0 0: Number of store stages 0 ⋮ 1 0 0: Number of store stages 4	R
31:3	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE-3, P-TYPE-3

RFDN[2:0] bit (Receive FIFO data store stage number)

Displays the number of stores receive FIFO stages. RFDN [2:0] is cleared by clearing the SPCR.SPE bit.

36.2.12 SPPSR : SPI Polling Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x60

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPEPS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPEPS	SPI Polling Status 0: SPCR.SPE is 0 1: SPCR.SPE is 1	R
31:1	—	These bits are read as 0. The write value should be 0.	R

SPEPS bit (SPI Polling Status)

This bit indicates status of SPCR.SPE bit after synchronization from bus clock (PCLK) to operation clock (TCLK).

36.2.13 SPSRC : SPI Status Clear Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x68

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SPRFC	CENDFC	SPTEFC	UDRFC	PERFC	MODFC	—	OVRFC	SPDRFC	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
22:0	—	The write value should be 0.	W
23	SPDRFC	SPI Receive Data Ready Flag Clear By writing 1, the SPI Receive Data Ready Flag can be cleared. Reading value is always 0.	W

Bit	Symbol	Function	R/W
24	OVRFC	Overrun Error Flag Clear By writing 1, the Overrun Error Flag can be cleared. Reading value is always 0.	W
25	—	The write value should be 0.	W
26	MODFC	Mode Fault Error Flag Clear By writing 1, the Mode Fault Error Flag can be cleared. Reading value is always 0.	W*1
27	PERFC	Parity Error Flag Clear By writing 1, the Parity Error Flag can be cleared. Reading value is always 0.	W
28	UDRFC	Underrun Error Flag Clear By writing 1, the Underrun Error Flag can be cleared. Reading value is always 0.	W*2
29	SPTEFC	SPI Transmit Buffer Empty Flag Clear By writing 1, the SPI Transmit Buffer Empty Flag can be cleared. Reading value is always 0.	W
30	CENDFC	Communication End Flag Clear By writing 1, the Communication End Flag can be cleared. Reading value is always 0.	W
31	SPRFC	SPI Receive Buffer Full Flag Clear By writing 1, the SPI Receive Buffer Full Flag can be cleared. Reading value is always 0.	W

Note: S-TYPE-3, P-TYPE-3
Note 1. Before setting MODFC and UDRFC, make sure that SPSR.MODF and UDRF are set to 1.
Note 2. When clearing the UDRF flag, clear the MODF flag at the same time (MODFC = 1).

The SPI status clear register (SPSRC) is a register that clears the status flag (SPSR) that indicates the operating status of SPI.

36.2.14 SPFCR : SPI FIFO Clear Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x6C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPFRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPFRST	SPI FIFO clear By writing 1, the pointer in the FIFO and the stored data are initialized. Reading value is always 0.	W
31:1	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The FIFO clear register (SPFCR) is used to clear the FIFO.

If SPFCR is rewritten while the SPE bit of the SPI control register (SPCR) is 1, subsequent operations are not guaranteed.

SPFRST bit (SPI FIFO clear)

Initializing the pointer and stored data in the transmit / receive FIFO by writing 1

36.3 Operation

In this section, the serial transfer period refers to the period from the beginning of driving valid data to the fetching of the final valid data.

36.3.1 Overview of SPI Operation

The SPI is capable of synchronous serial transfers in the following modes:

- Slave mode (SPI operation)
- Single-master mode (SPI operation)
- Multi-master mode (SPI operation)
- Slave mode (clock synchronous operation)
- Master mode (clock synchronous operation)

The SPI mode can be selected by using the MSTR, MODFEN, SPMS, and SPFRF bits in SPCR. Table 36.6 lists the relationship between SPI modes and SPCR settings, and a description of each mode.

Table 36.6 Relationship between SPCR settings and SPI modes (1 of 2)

Mode	Slave (SPI operation)	Single-master (SPI operation)	Multi-master (SPI operation)	Slave (clock synchronous operation)	Master (clock synchronous operation)
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0 or 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
SPFRF bit setting	valid	valid	valid	Invalid	Invalid
RSPCKn pins	Input	Output	Output/Hi-Z	Input	Output
MOSIn pin	Input	Output	Output/Hi-Z	Input	Output
MISOn pin	Output/Hi-Z	Input	Input	Output	Input
SSLn0 pins	Input	Output	Input	Hi-Z*1	Hi-Z*1
SSLn1 to SSLn3 pins	Hi-Z*1	Output	Output/Hi-Z	Hi-Z*1	Hi-Z*1
SSL polarity change function	Supported	Supported	Supported	—	—
Max transfer rate	TCLK/2	TCLK/2	TCLK/2	TCLK/2	TCLK/2
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	Two				
Clock phase	Two*6	Two*6	Two*6	One (CPHA = 1)	Two
Transfer data length	4 to 32 bits				
Burst transfer	Possible (CPHA = 1)	Possible (CPHA = 0, 1)	Possible (CPHA = 0, 1)	—	—
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported*7	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported
Transfer trigger	SSL input active or RSPCK oscillation	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)	RSPCK oscillation	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported*5				
Receive buffer full detection	Supported*2				
Overrun error detection	Supported*2	Supported*2 *4	Supported*2 *4	Supported*2	Supported*2

Table 36.6 Relationship between SPCR settings and SPI modes (2 of 2)

Mode	Slave (SPI operation)	Single-master (SPI operation)	Multi-master (SPI operation)	Slave (clock synchronous operation)	Master (clock synchronous operation)
Parity error detection	Supported*3 *2				
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported
Underrun error detection	Supported*5	Not supported	Not supported	Supported*5	Not supported

Note 1. This function is not supported in this mode.
Note 2. When SPI is transmit-master mode or transmit-slave mode (see Table 36.4), detection of receive buffer full, overrun error, and parity error are not performed.
Note 3. When the SPCR.SPPE bit is 0, parity error detection is not performed.
Note 4. When the SPCR.SCKASE bit is 1, overrun error detection does not proceed.
Note 5. When SPI is receive only slave mode, none of transmit buffer empty and underrun error is detected.
Note 6. CPHA = 0 is invalid in TI SSP mode. (If it is set, the operation is the same as when CPHA = 1.)
Note 7. Supported only in TI SSP mode.

36.3.2 Controlling the SPI Pins

Based on the settings of the MSTR, MODFEN, and SPMS bits in SPCR and the PmnPFS.NCODR bit for I/O Ports, the SPI can switch pin states. Table 36.7 lists the relationship between pin states and bit settings. Setting the PmnPFS.NCODR bit for an I/O port to 0 selects the CMOS output. Setting it to 1 selects the open-drain output. The I/O port settings must follow this relationship.

Table 36.7 Relationship between pin states and bit settings (1 of 2)

Mode	Pin	Pin state*2	
		PmnPFS.NCODR bit for I/O ports = 0	PmnPFS.NCODR bit for I/O ports = 1
Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0)	RSPCKn	CMOS output	Open-drain output
	SSLn0 to SSLn3	CMOS output	Open-drain output
	MOSIn	CMOS output	Open-drain output
	MISOn	Input	Input
Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCKn*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSLn0	Input	Input
	SSLn1 to SSLn3*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSIn*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MISOn	Input	Input
Slave mode (SPI operation) (MSTR = 0, SPMS = 0)	RSPCKn	Input	Input
	SSLn0	Input	Input
	SSLn1 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	Input	Input
	MISOn*4	CMOS output/Hi-Z	Open-drain output/Hi-Z
Master mode (clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1)	RSPCKn	CMOS output	Open-drain output
	SSLn0 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	CMOS output	Open-drain output
	MISOn	Input	Input

Table 36.7 Relationship between pin states and bit settings (2 of 2)

Mode	Pin	Pin state*2	
		PmnPFS.NCODR bit for I/O ports = 0	PmnPFS.NCODR bit for I/O ports = 1
Slave mode (clock synchronous operation) (MSTR = 0, SPMS = 1)	RSPCKn	Input	Input
	SSLn0 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	Input	Input
	MISOn	CMOS output	Open-drain output

Note 1. This function is not supported in this mode.
Note 2. SPI settings are not reflected in multiplexed pins for which the SPI function is not selected.
Note 3. Motorola-SPI: When SSLn0 is at the active level, the pin state is Hi-Z. Whether or not the input signal is at the active level determines the setting of the SPCR3.SSL0P bit.
TI-SSP: From when SSL0 is at the active level until communication is completed, the pin state is Hi-Z under the condition SPCR.SPE = 1.
Note 4. Motorola-SPI: When SSLn0 is at the non-active level or the SPCR.SPE bit is 0, the pin state is Hi-Z. Whether or not the input signal is at the active level determines the setting of the SPCR3.SSL0P bit.
TI-SSP: When SSL0 is except the communication period or when the SPE bit of SPCR is 0 (assertion after SPE = 1 and communication is completed), the pin status changes to Hi-Z.
Note 5. These pins are available for use as I/O port pins.

The SPI in single-master mode (SPI operation) or multi-master mode (SPI operation) determines the MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer) based on the MOIFE and MOIFV bit settings in SPCR2, as listed in Table 36.8.

Table 36.8 MOSI signal value determination during SSL negation

MOIFE bit	MOIFV bit	MOSIn signal value during SSL negation
0	0, 1	Final data from previous transfer
1	0	Low
1	1	High

36.3.3 SPI System Configuration Examples

This configuration example describes that 0 level of SSL0n signals is active level.

When connecting and using in a multi-slave or multi-master mode, the transfer format of the connected device should be unified to either Motorola-SPI or TI-SSP.

36.3.3.1 Single-master/single-slave with the MCU as a master

Figure 36.6 shows a single-master/single-slave SPI system configuration example where the MCU is used as a master. In the single-master/single-slave configuration, the SSLni outputs of the MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave is maintained in the selected state.*1

Note 1. In the transfer format configured when the SPCMDm.CPHA bit is 0, the SSL signal for some slave devices cannot be fixed to an active level. In this case, always connect the SSLni output of the MCU to the SSL input of the slave device.

The MCU (master) drives the RSPCKn and MOSIn signals. The SPI slave drives the MISO signal.

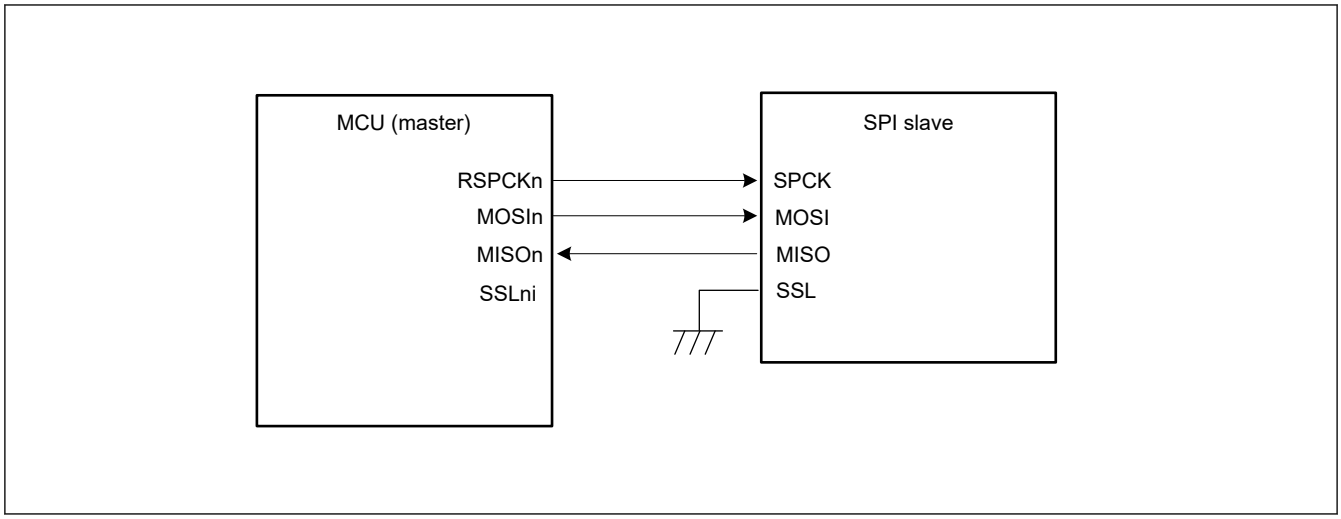


Figure 36.6 Single-master/single-slave configuration example with the MCU as a master

36.3.3.2 Single-master/single-slave with the MCU as a slave

Figure 36.7 shows a single-master/single-slave SPI system configuration example where the MCU is used as a slave. When the MCU operates as a slave, the SSLn0 pin is used as SSL input. The SPI master drives the RSPCK and MOSI signals. The MCU (slave) drives the MISOn signal.*1

Note 1. When SSLn0 is at a non-active level, the pin state is Hi-Z.

In the single-slave configuration when the SPCMDm.CPHA bit is set to 1, the SPCR.SPFRF bit is set to 0, and SPCR.SPMS is set to 0, the SSLn0 input of the MCU (slave) is fixed to the low level and the MCU (slave) is maintained in the selected state. This enables serial transfer execution (Figure 36.8). However, the communication end interrupt does not output when SSL0 input is fixed as Figure 36.8.

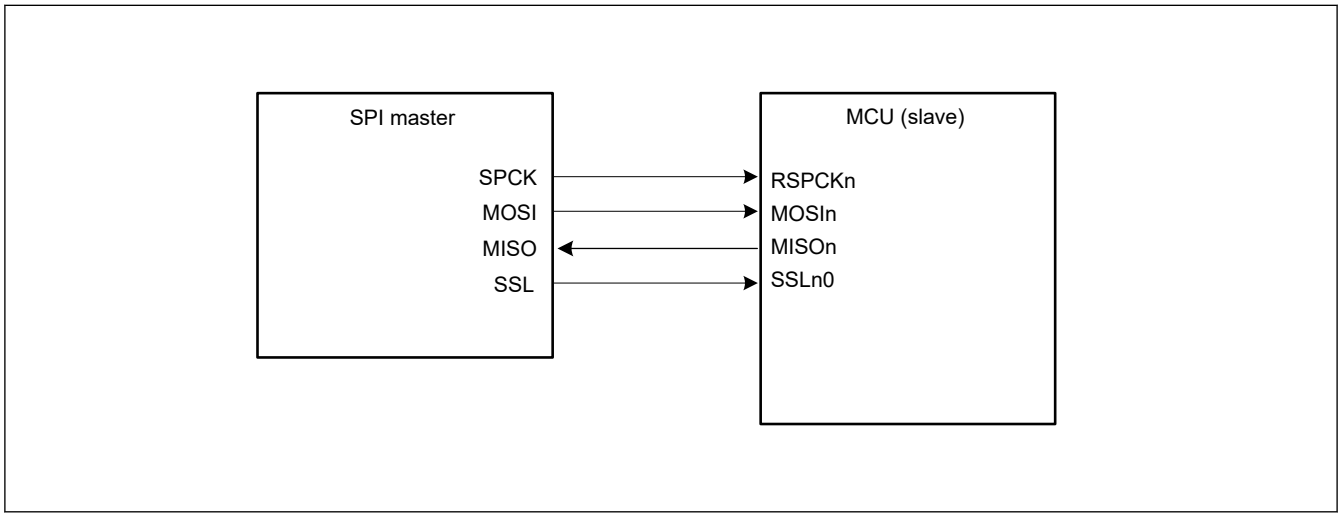


Figure 36.7 Single-master/single-slave configuration example with the MCU as a slave and CPHA = 0

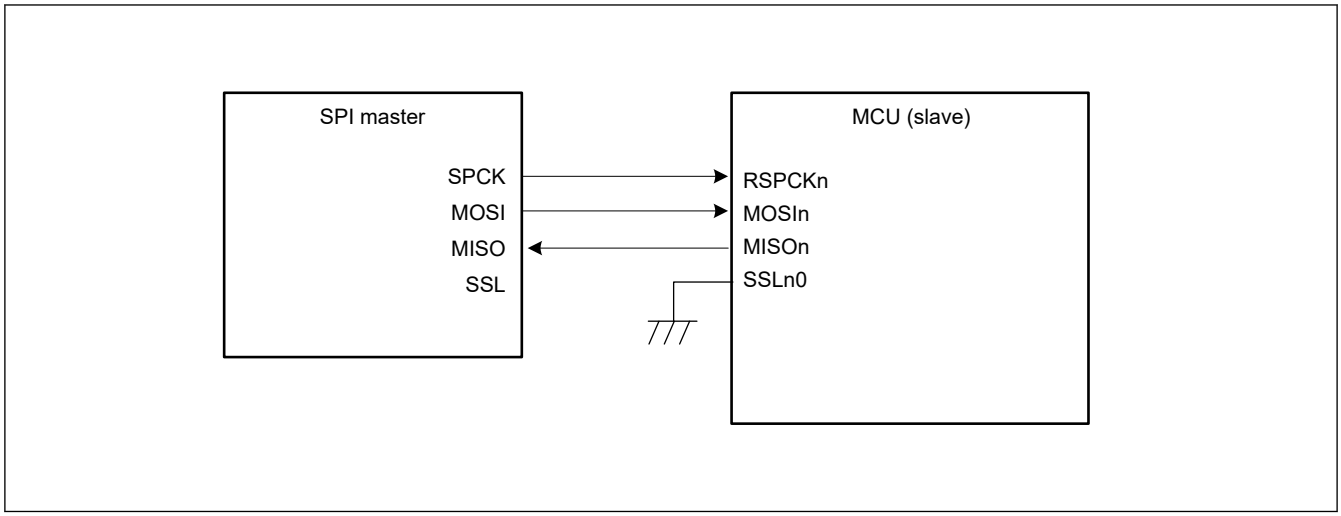


Figure 36.8 Single-master/single-slave configuration example with the MCU as a slave and CPHA = 1

36.3.3.3 Single-master/multi-slave with the MCU as a master

Figure 36.9 shows a single-master/multi-slave SPI system configuration example where the MCU is used as a master. In this example, the SPI system includes the MCU (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCKn and MOSIn outputs of the MCU (master) are connected to the RSPCK and MOSI inputs of SPI slaves 0 to 3. The MISO outputs of SPI slaves 0 to 3 are all connected to the MISOn input of the MCU (master). The SSLn0 to SSLn3 outputs of the MCU (master) are connected to the SSL inputs of SPI slaves 0 to 3, respectively.

The MCU (master) drives the RSPCKn, MOSIn, and SSLn0 to SSLn3 signals. Out of the SPI slaves 0 to 3, the slave that receives low-level input into the SSL input drives the MISO signal.

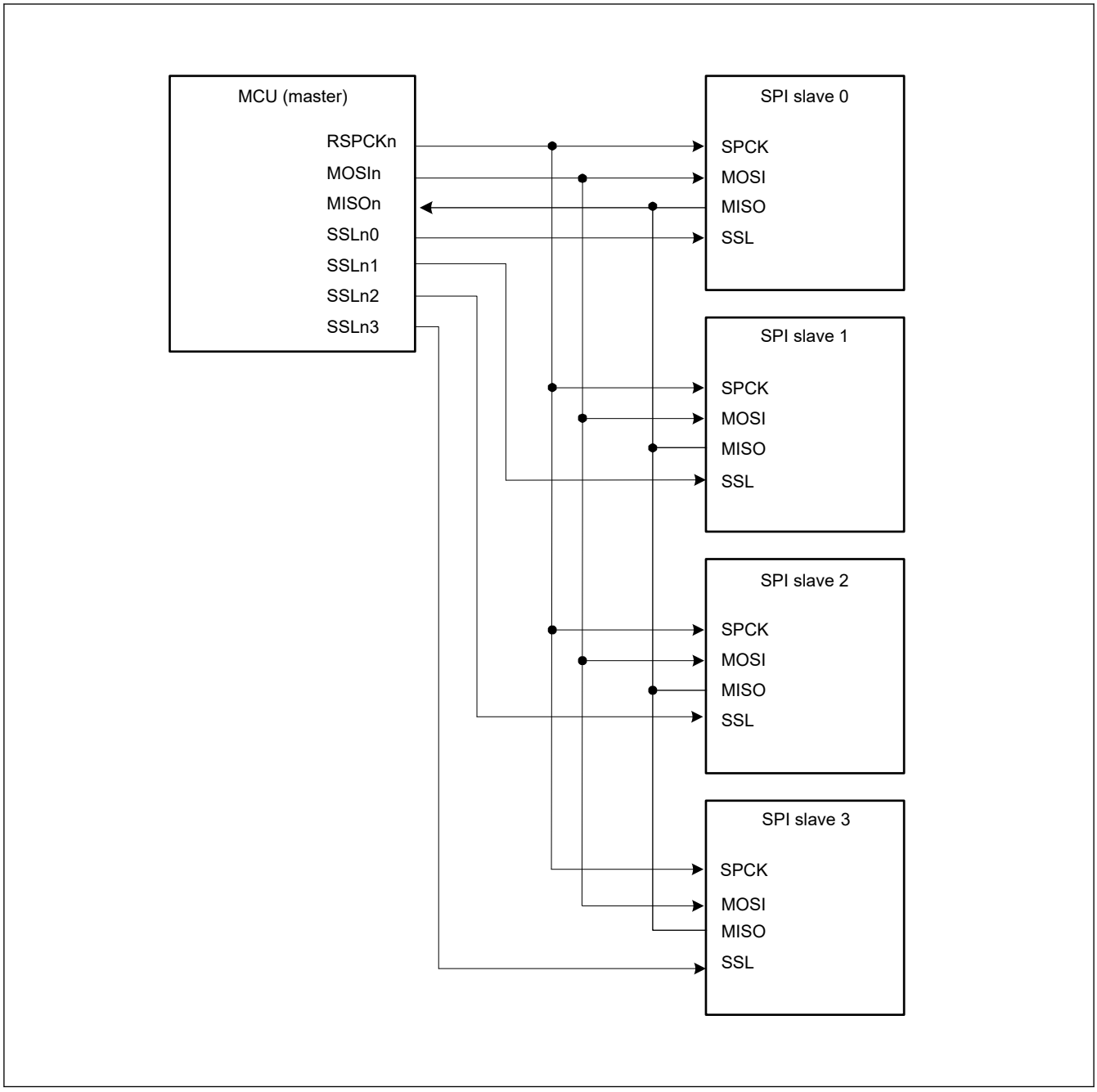


Figure 36.9 Single-master/multi-slave configuration example with the MCU as a master

36.3.3.4 Single-master/multi-slave with the MCU as a slave

Figure 36.10 shows a single-master/multi-slave SPI system configuration example where the MCU is used as a slave. In this example, the SPI system includes an SPI master and two MCUs (slaves X and Y).

The SPCK and MOSI outputs of the SPI master are connected to the RSPCKn and MOSIn inputs of the MCUs (slaves X and Y). The MISOn outputs of the MCUs (slaves X and Y) are all connected to the MISO input of the SPI master. The SSLX and SSLY outputs of the SPI master are connected to the SSLn0 inputs of the MCUs (slaves X and Y, respectively).

The SPI master drives the SPCK, MOSI, SSLX, and SSLY signals. Of the MCUs (slaves X and Y), the slave that receives low-level input into the SSLn0 input drives the MISOn signal.

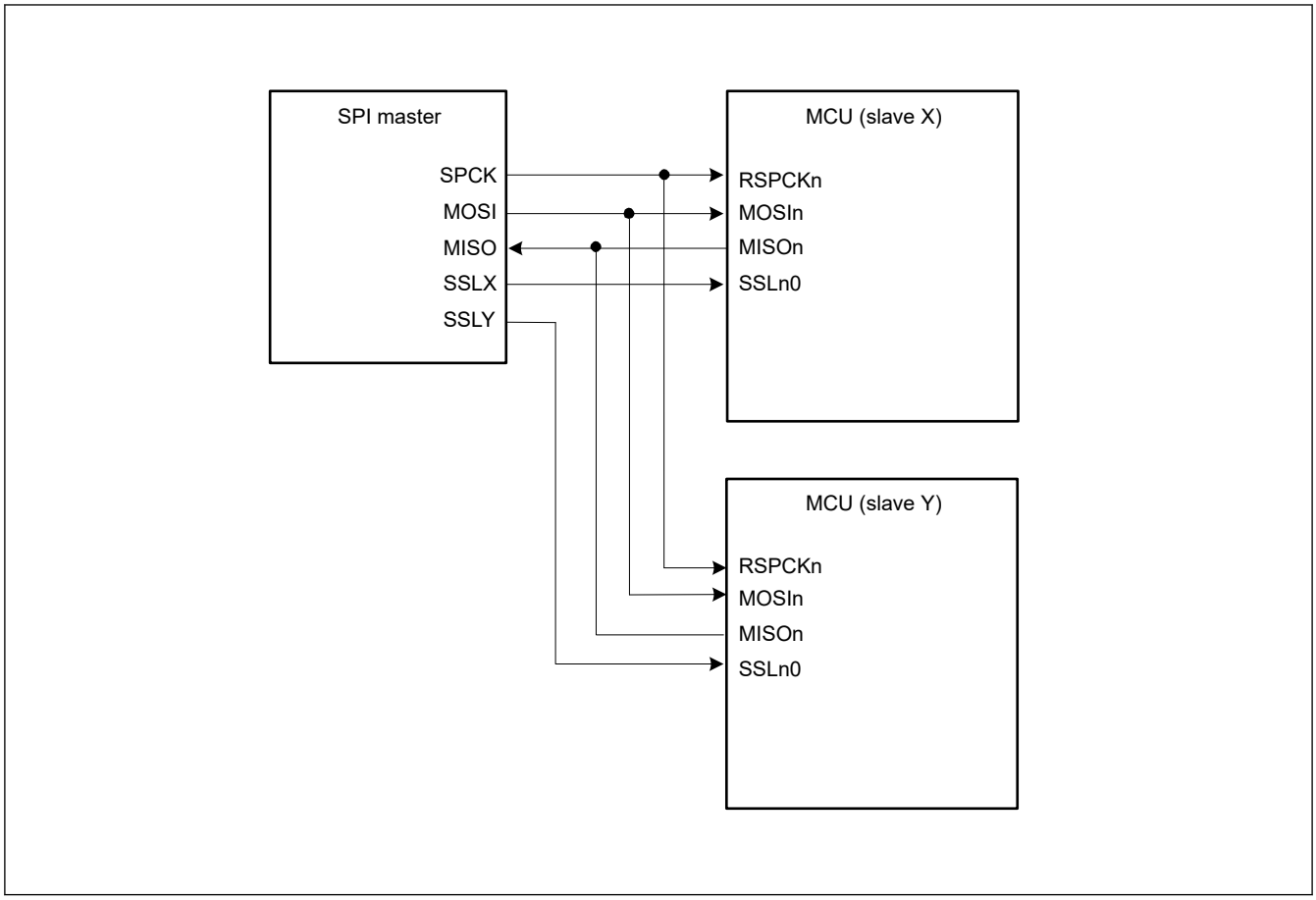


Figure 36.10 Single-master/multi-slave configuration example with the MCU as a slave

36.3.3.5 Multi-master/multi-slave with the MCU as a master

Figure 36.11 shows a multi-master/multi-slave SPI system configuration example where the MCU is used as a master. In this example, the SPI system includes two MCUs (masters X and Y) and two SPI slaves (SPI slaves 1 and 2).

The RSPCKn and MOSIn outputs of the MCUs (masters X and Y) are connected to the RSPCK and MOSI inputs of SPI slaves 1 and 2. The MISO outputs of SPI slaves 1 and 2 are connected to the MISOn inputs of the MCUs (masters X and Y). Any generic port Y output from the MCU (master X) is connected to the SSLn0 input of the MCU (master Y). Any generic port X output of the MCU (master Y) is connected to the SSLn0 input of the MCU (master X). The SSLn1 and SSLn2 outputs of the MCUs (masters X and Y) are connected to the SSL inputs of the SPI slaves 1 and 2. In this configuration example, because the system can be comprised solely of SSLn0 input, and SSLn1 and SSLn2 outputs for slave connections, the SSLn3 output of the MCU is not required.

The MCU drives the RSPCKn, MOSIn, SSLn1, and SSLn2 signals when the SSLn0 input level is high. When the SSLn0 input level is low, the MCU detects a mode fault error, sets RSPCKn, MOSIn, SSLn1, and SSLn2 to Hi-Z, and releases the SPI bus directly to the other master. Of the SPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives the MISO signal.

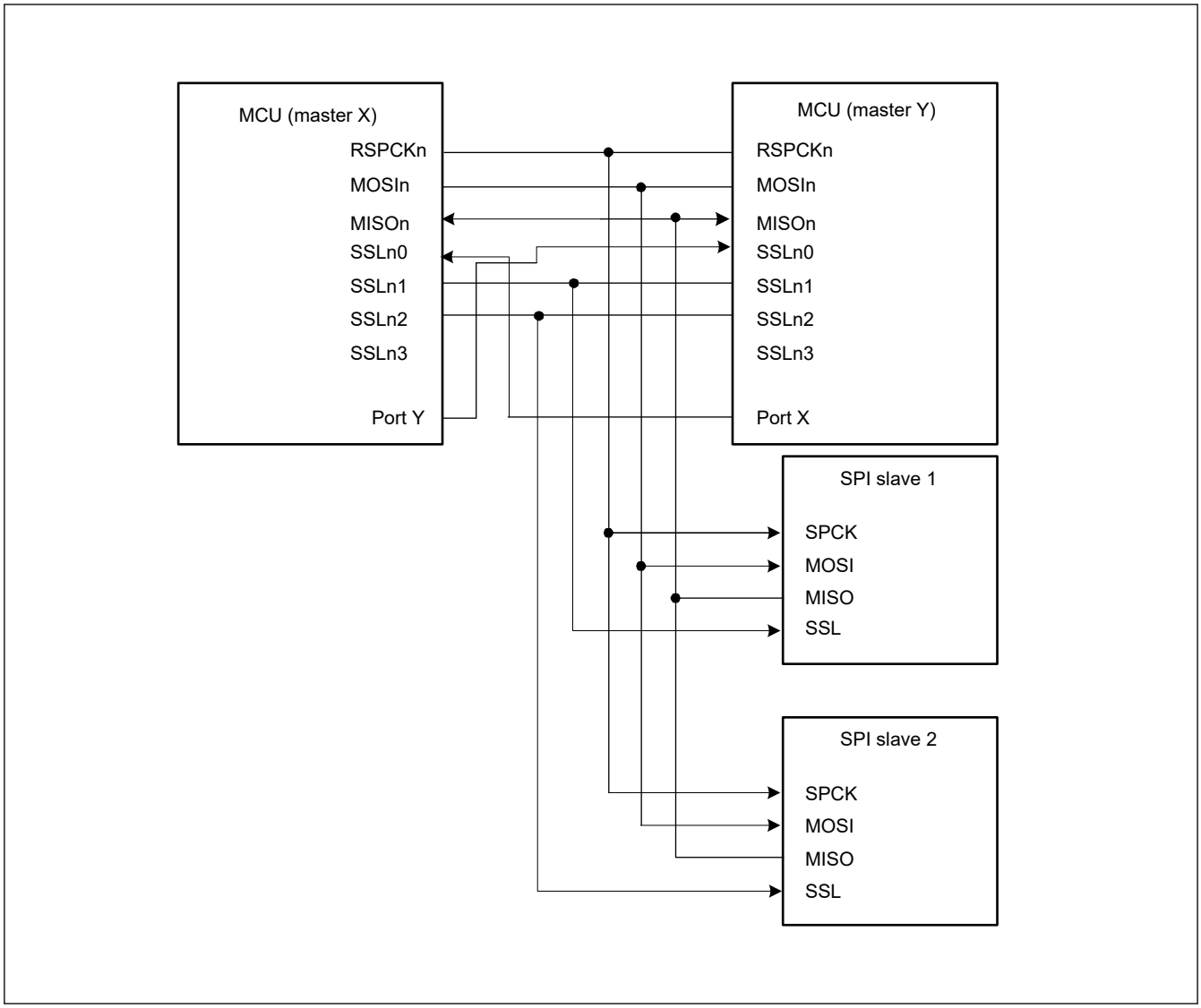


Figure 36.11 Multi-master/multi-slave configuration example with the MCU as a master

When setting TI SSP, enter the following levels for port X and port Y.

- Start of communication: the value of SPCR3.SSL0P of the other master.
- End of communication: the inverted value of SPCR3.SSL0P of the other master.

36.3.3.6 Master and slave in clock synchronous mode with the MCU configured as a master

Figure 36.12 shows a master and slave in clock synchronous mode configuration example where the MCU is used as a master. In this configuration, SSLni of the MCU (master) are not used.

The MCU (master) drives the RSPCKn and MOSIn signals. The SPI slave drives the MISO signal.

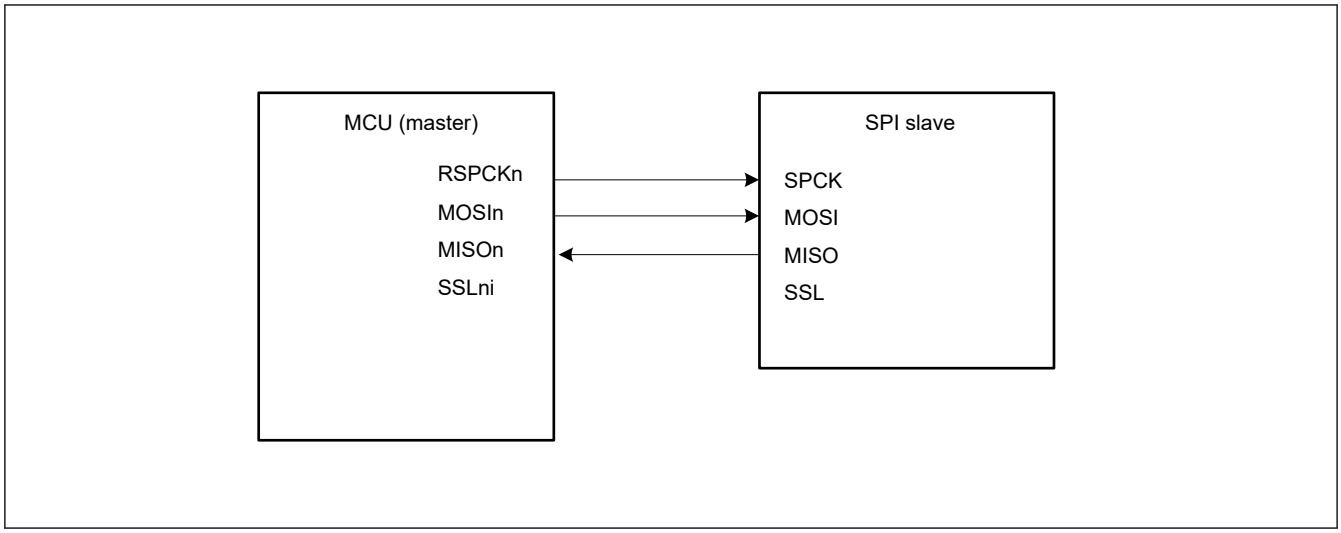


Figure 36.12 Clock synchronous master/slave configuration example with the MCU as a master

36.3.3.7 Master and slave in clock synchronous mode with the MCU as a slave

Figure 36.13 shows a master and slave in clock synchronous mode configuration example where the MCU is used as a slave. When the MCU operates as a slave (clock synchronous operation), the MCU (slave) drives the MISOn signal and the SPI master drives the SPCK and MOSI signals. In addition, SSLn0 to SSLn3 of the MCU (slave) are not used.

The MCU (slave) can only execute serial transfers in the single-slave configuration when the SPCMDm.CPHA bit is set to 1.

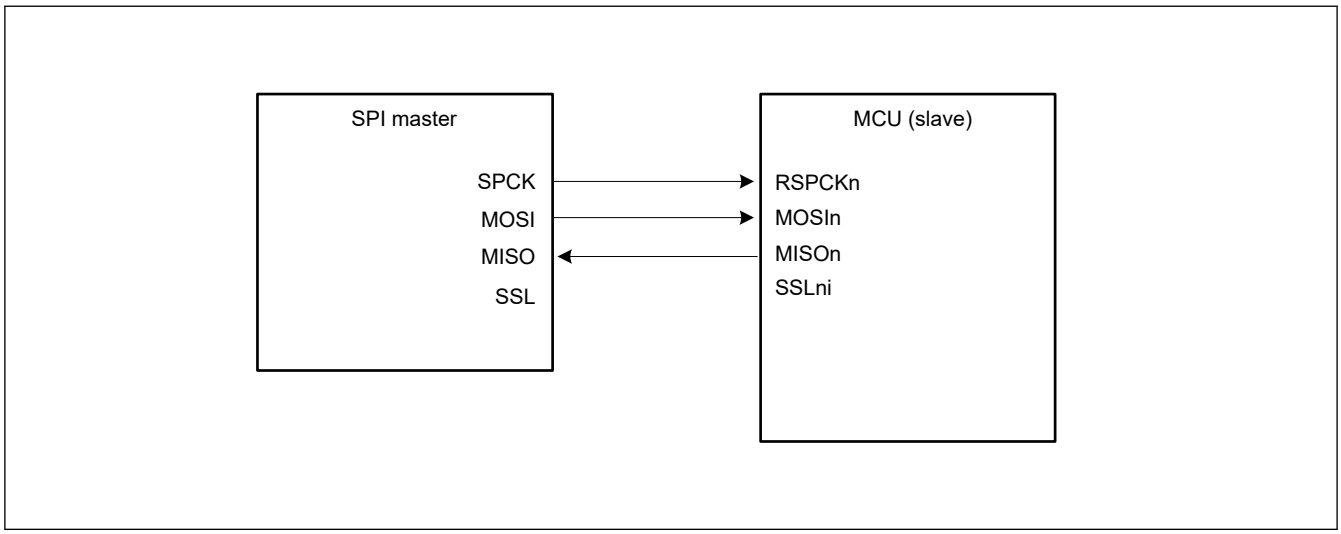


Figure 36.13 Clock synchronous master/slave configuration example with the MCU as a slave and CPHA = 1

36.3.4 Data Formats

The data format of the SPI depends on the settings in SPI Command Register m (SPCMDm) and the parity enable bit in SPI Control Register (SPCR.SPPE). Regardless of whether the MSB or LSB is first, the SPI treats the range from the LSB bit in the SPI Data Register (SPDR) to the bit associated with the selected data length, as transfer data.

This section shows the format of one frame of data before or after transfer.

Data format with parity disabled

When parity is disabled, transmission or reception of data proceeds with the length in bits selected in the SPI data length setting in SPI Command Register m (SPCMDm.SPB[4:0]).

Data format with parity enabled

When parity is enabled, transmission or reception of data proceeds with the length in bits selected in the SPI data length setting in SPI Command Register m (SPCMDm.SPB[4:0]). In this case, however, the last bit is a parity bit.

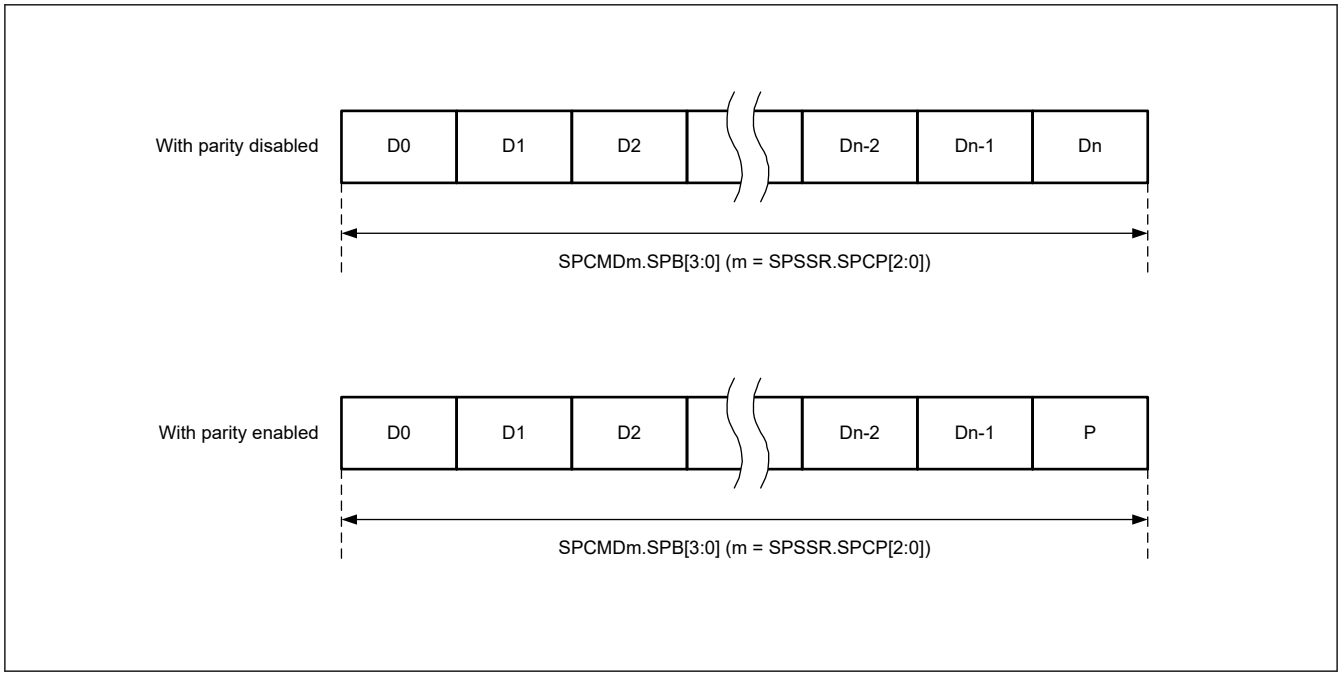


Figure 36.14 Data format with parity disabled and enabled

36.3.4.1 Operation when parity is disabled (SPCR.SPPE = 0)

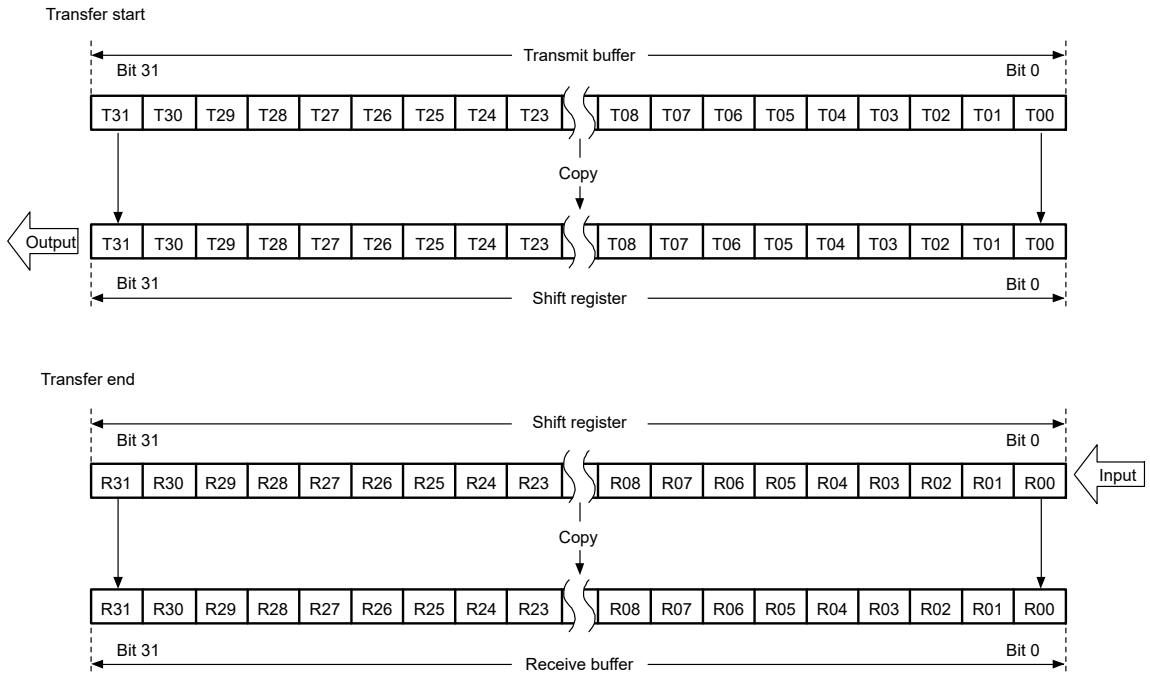
When parity is disabled, data for transmission is copied to the shift register with no pre-processing. This section describes the connection between the SPI Data Register (SPDR) and the shift register in terms of the combination of MSB- or LSB-first order and data length.

(1) MSB-first transfer with 32-bit data

Figure 36.15 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, a SPI data length of 32 bits, and MSB-first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are copied to the shift register. Data for transmission is shifted out from the shift register from T31 to T30, and continuing to T00.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R31 to R00 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 36.15 MSB-first transfer with 32-bit data and parity disabled

(2) MSB-first transfer with 24-bit data

Figure 36.16 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, an SPI data length of 24 bits for an example that is not 32 bits, and MSB-first selected.

In transmission, the lower 24 bits (T23 to T00) from the current stage of the transmit buffer are copied to the shift register. Data for transmission is shifted out from the shift register from T23 to T22, and continuing to T00.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R23 to R00 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.

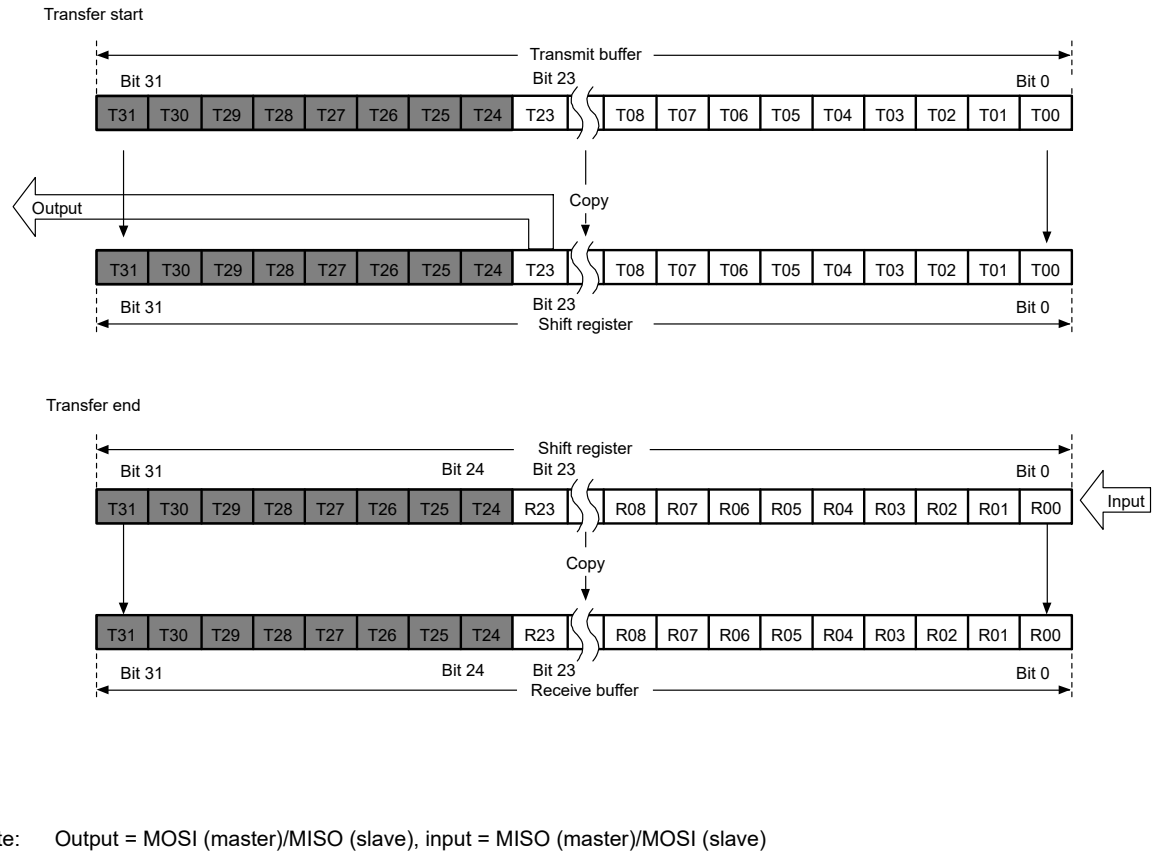


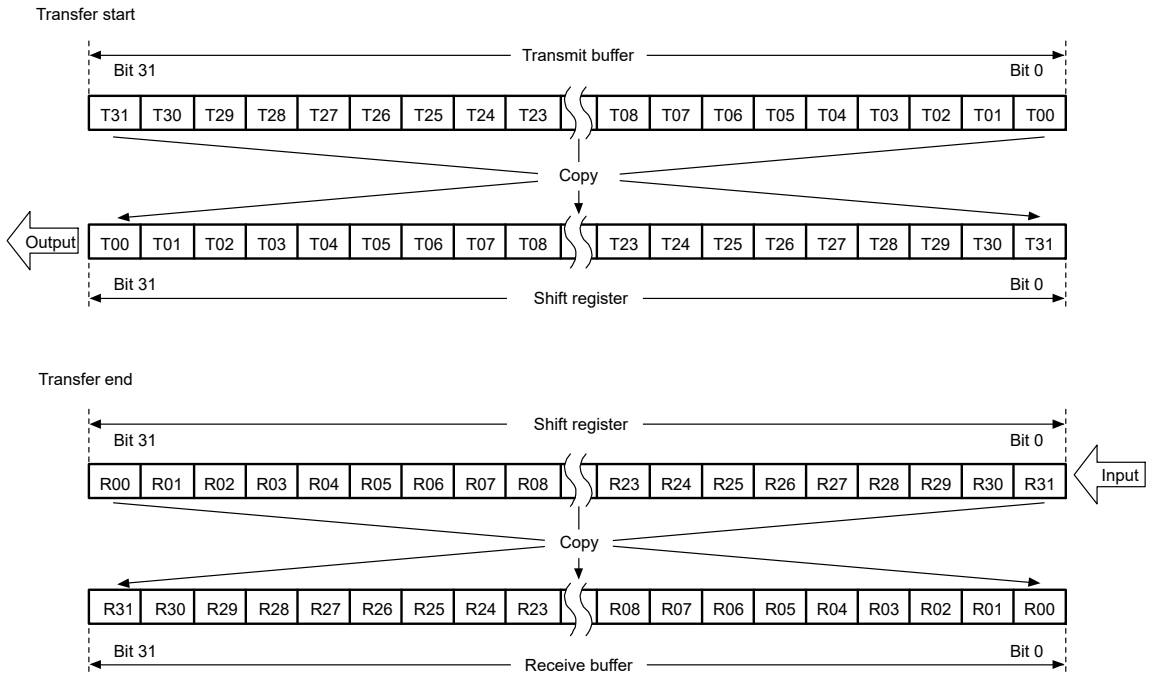
Figure 36.16 MSB-first transfer with 24-bit data and parity disabled

(3) LSB-first transfer with 32-bit data

Figure 36.17 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, an SPI data length of 32 bits, and LSB-first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are reordered bit-by-bit to obtain the order T00 to T31 for copying to the shift register. Data for transmission is shifted out from the shift register in order from T00 to T01, and continuing to T31.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R00 to R31 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

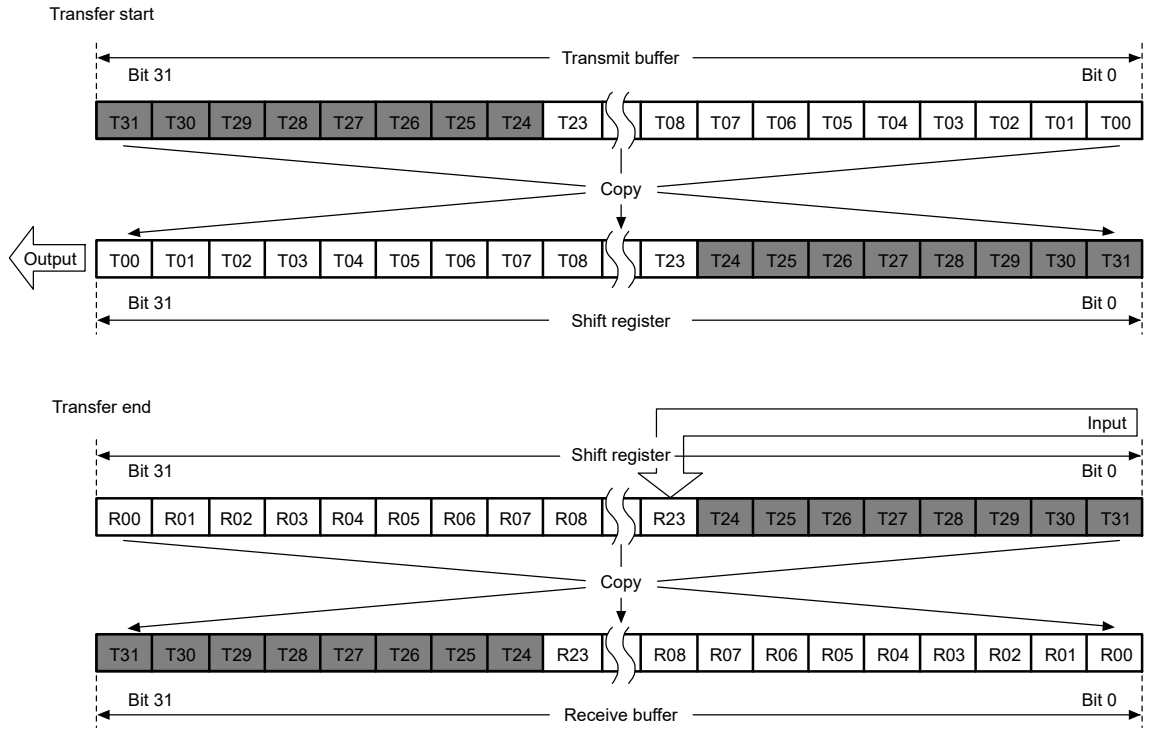
Figure 36.17 LSB-first transfer with 32-bit data and parity disabled

(4) LSB-first transfer with 24-bit data

Figure 36.18 shows the operation of the SPI Data Register (SPDR) and the shift register in transfers with parity disabled, an SPI data length of 24 bits for an example that is not 32, and LSB-first selected.

In transmission, the lower 24 bits (T23 to T00) from the current stage of the transmit buffer are reordered bit-by-bit to obtain the order T00 to T23 for copying to the shift register. Data for transmission is shifted out from the shift register from T00 to T01, and continuing to T23.

In reception, received data is shifted in bit-by-bit through bit[8] of the shift register. When the R00 to R23 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 36.18 LSB-first transfer with 24-bit data and parity disabled

36.3.4.2 Operation when parity is enabled (SPCR.SPPE = 1)

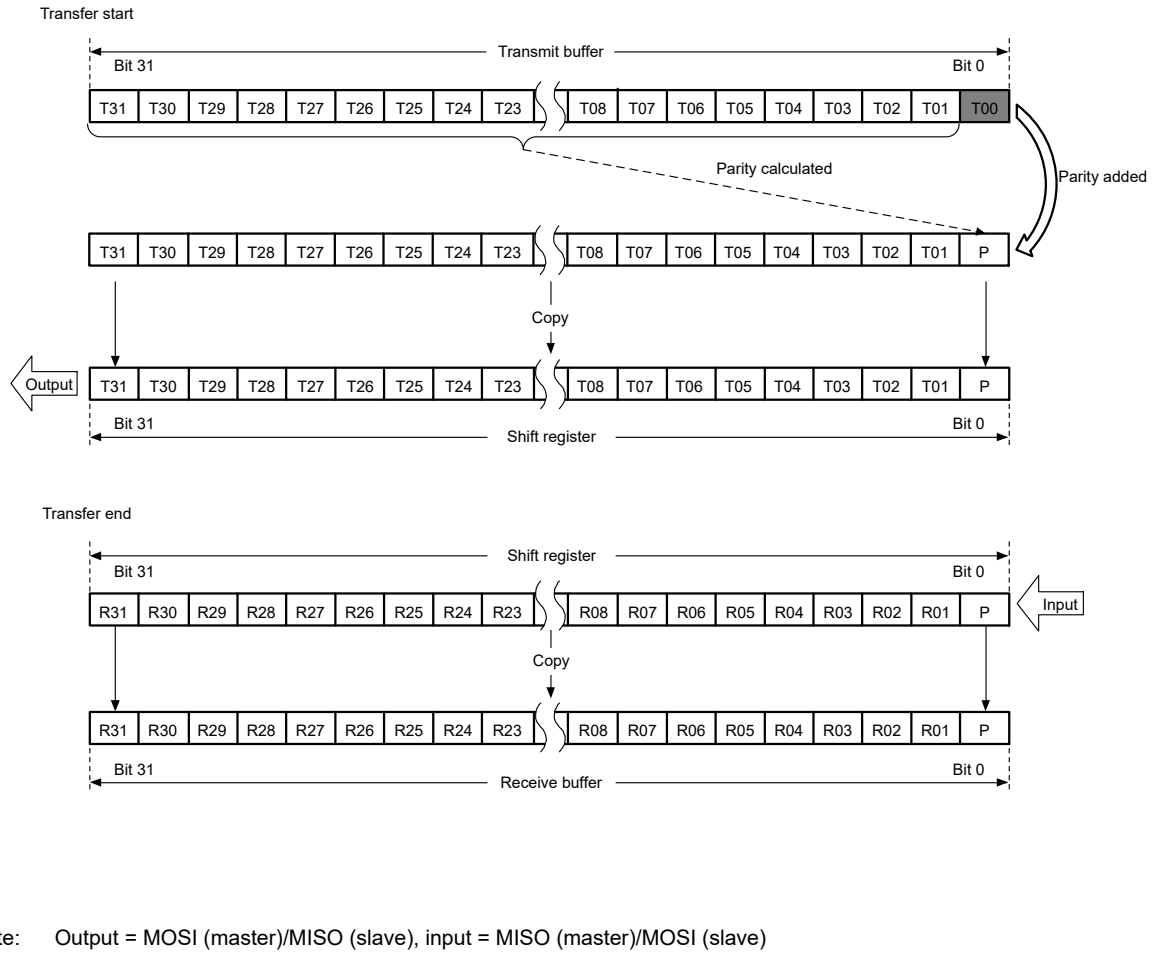
When parity is enabled, the lowest-order bit of the data for transmission becomes a parity bit. Hardware calculates the value of the parity bit.

(1) MSB-first transfer with 32-bit data

Figure 36.19 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, an SPI data length of 32 bits, and MSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T31 to T01. This replaces the final bit, T00, and the whole value is copied to the shift register. Data is transmitted in the order T31, T30, …, T01, and P.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R31 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R31 to P is checked for parity.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

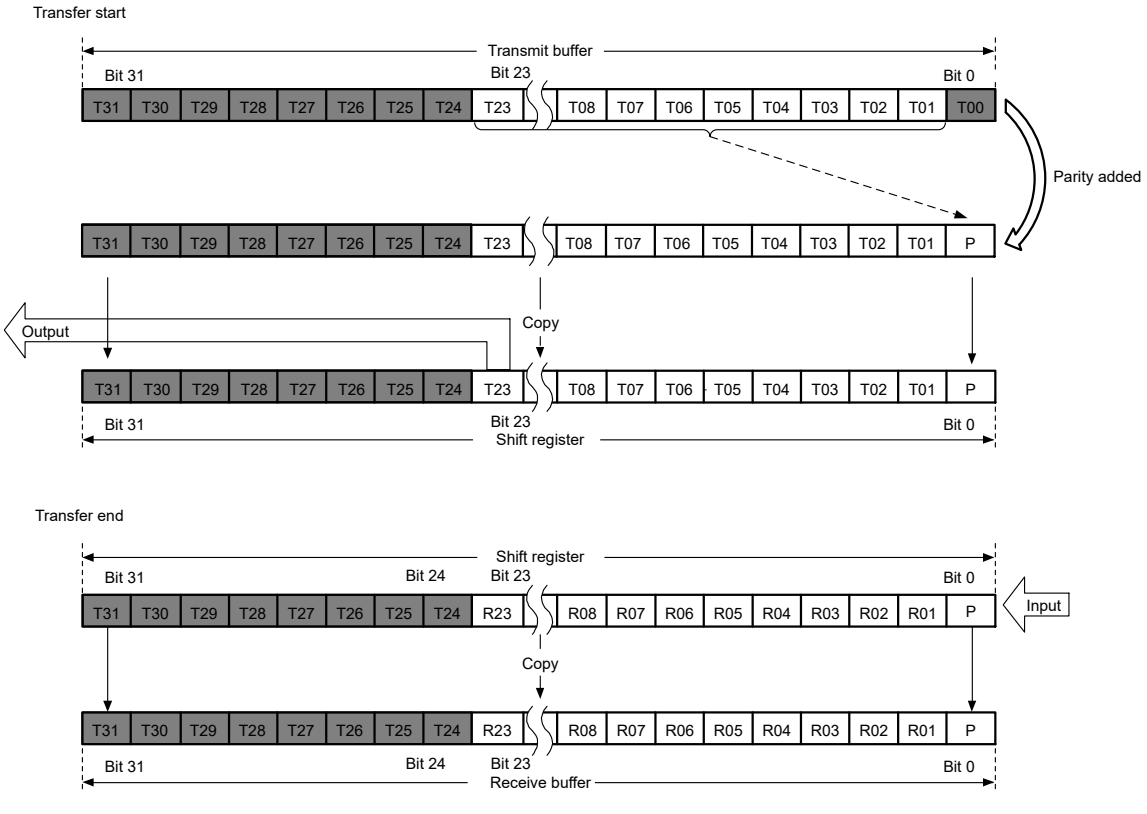
Figure 36.19 MSB-first transfer with 32-bit data and parity enabled

(2) MSB-first transfer with 24-bit data

Figure 36.20 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, a SPI data length of 24 bits, and MSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T23 to T01. This replaces the final bit, T00, and the whole value is copied to the shift register. Data is transmitted in the order T23, T22, …, T01, and P.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R23 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R23 to P is checked for parity.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

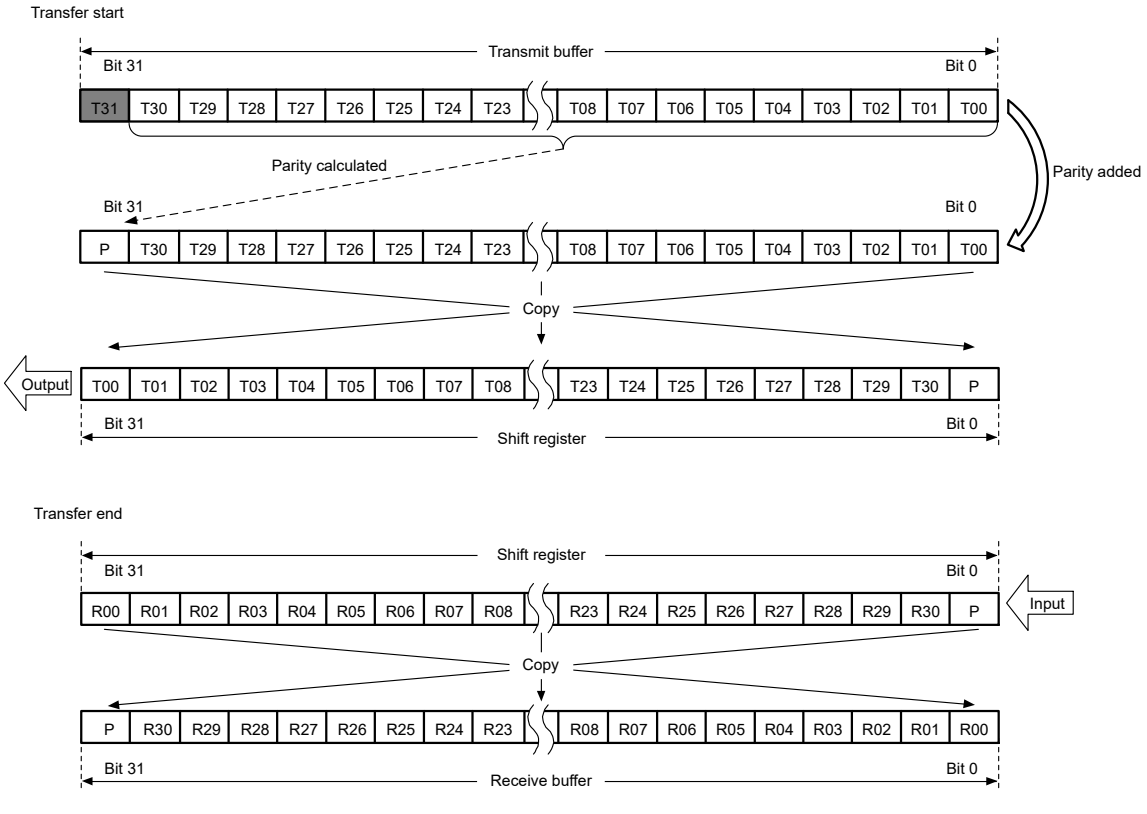
Figure 36.20 MSB-first transfer with 24-bit data and parity enabled

(3) LSB-first transfer with 32-bit data

Figure 36.21 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, an SPI data length of 32 bits, and LSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T30 to T00. This replaces the final bit, T31, and the whole value is copied to the shift register. Data is transmitted in the order T00, T01, …, T30, and P.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R00 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P is checked for parity.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

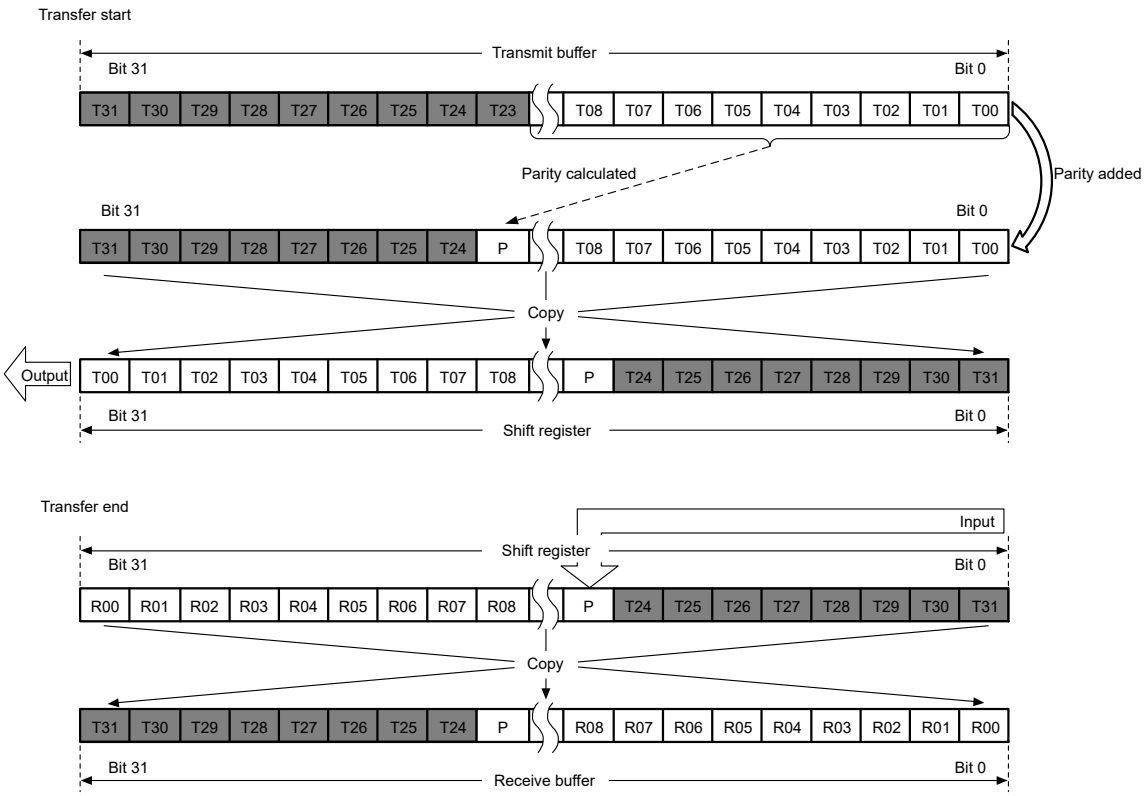
Figure 36.21 LSB-first transfer with 32-bit data and parity enabled

(4) LSB-first transfer with 24-bit data

Figure 36.22 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, a SPI data length of 24 bits, and LSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T22 to T00. This replaces the final bit, T23, and the whole value is copied to the shift register. Data is transmitted in the order T00, T01, …, T22, and P.

In reception, received data is shifted in bit-by-bit through bit[8] of the shift register. When the R00 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P is checked for parity.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 36.22 LSB-first transfer with 24-bit data and parity enabled

36.3.4.3 Byte Swap Transmission

When byte swapping is enabled, the data in the transmission buffer, swapped in 8-bit units, is copied to the shift register. Figure 36.23 shows the relationship between the SPDR (transmit buffer) and the shift register when transferring data with a 32-bit data length, using a combination of MSB / LSB first and with / without byte swap.

(1) MSB-first transfer. (When the byte swap is disabled.)

Data (Byte3 [T31 to T24] to Byte0 [T07 to T00]) in the transmit buffer are copied to the shift register.

Bit values in the shift register are shifted and transmitted in the order of T31 → T30 → … → T00 as transmit data.

(2) MSB-first transfer. (When the byte swap is enabled.)

Byte values of the transmit buffer (Byte3 [T31 to T24] to Byte0 [T07 to T00]) are reversed in byte units and are copied to the shift register in the order of Byte0 [T07 to T00] to Byte3 [T31 to T24].

Bit values in the shift register are shifted and transmitted in the order of T07 → T06 → … → T00 → T15 → T14 → … → T08 → T23 → T22 → … → T16 → T31 → T30 → … → T24 as transmit data.

(3) LSB-first transfer. (When the byte swap is disabled.)

Bit values of the transmit buffer (Byte3 [T31 to T24] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte0 [T00 to T07] to Byte3 [T24 to T31].

Bit values in the shift register are shifted and transmitted in the order of T00 → T01 → … → T31 as transmit data.

(4) LSB-first transfer. (When the byte swap is enabled.)

Bit values of each byte of the transmit buffer (Byte3 [T31 to T24] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte3 [T24 to T31] to Byte0 [T00 to T07].

Bit values in the shift register are shifted and transmitted in the order of T24 → T25 → … → T31 → T16 → T17 → … → T23 → T08 → T09 → … → T15 → T00 → T01 → … → T07 as transmit data.

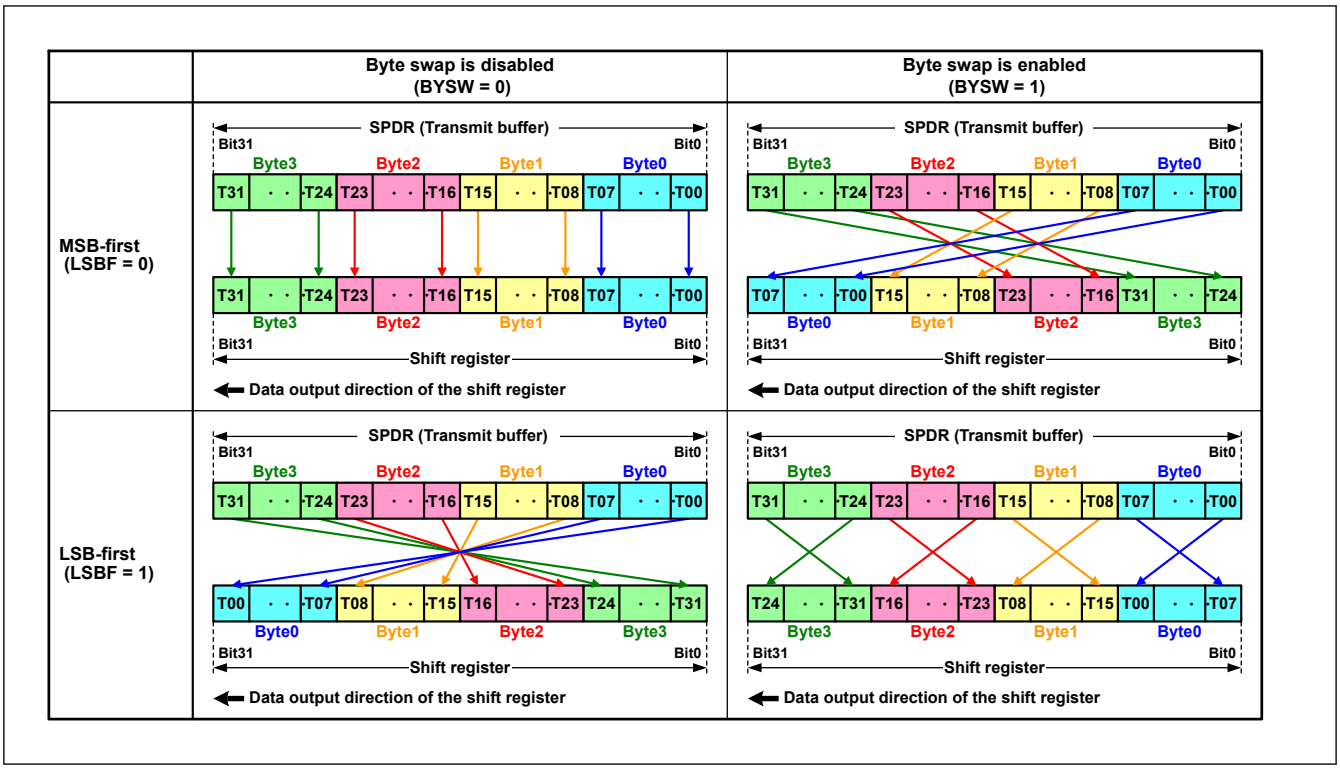


Figure 36.23 Byte swap with MSB/LSB transfer (32bit)

Figure 36.24 shows the relationship between the SPDR (transmit buffer) and the shift register when transferring data with a 16-bit data length, using a combination of MSB / LSB first and with / without byte swap.

1. MSB-first transfer. (When the byte swap is disabled.)
   Data (Byte1 [T15 to T08] to Byte0 [T07 to T00]) in the transmit buffer are copied to the shift register in the order of Byte1 [T15 to T08] to Byte0 [T07 to T00], Byte1 [T15 to T08] to Byte0 [T07 to T00]. Bit values in the shift register are shifted and transmitted in the order of T15 → T14 → … T00 as transmit data.
2. MSB-first transfer. (When the byte swap is enabled.)
   Byte values of the transmit buffer (Byte1 [T15 to T08] to Byte0 [T07 to T00]) are reversed in byte units and are copied to the shift register in the order of Byte0 [T07to T00] to Byte1 [T15 to T08], Byte0 [T07 to T00] to Byte1 [T15 to T08]. Bit values in the shift register are shifted and transmitted in the order of T07 → T06 → … T00 → T15 → T14 → … T08 as transmit data.
3. LSB-first transfer. (When the byte swap is disabled.)
   Bit values of the transmit buffer (Byte1 [T15 to T08] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte0 [T00 to T07] to Byte1 [T08 to T15], Byte0 [T00 to T07] to Byte1 [T08 to T15]. Bit values in the shift register are shifted and transmitted in the order of T00 → T01 → … T15 as transmit data.
4. LSB-first transfer. (When the byte swap is enabled.)
   Bit values of each byte of the transmit buffer (Byte1 [T15 to T08] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte1 [T08 to T15] to Byte0 [T00 to T07], Byte1 [T08 to T15] to Byte0 [T00 to T07]. Bit values in the shift register are shifted and transmitted in the order of T08 → T09 → … T15 → T00 → T01 → … T07 as transmit data.

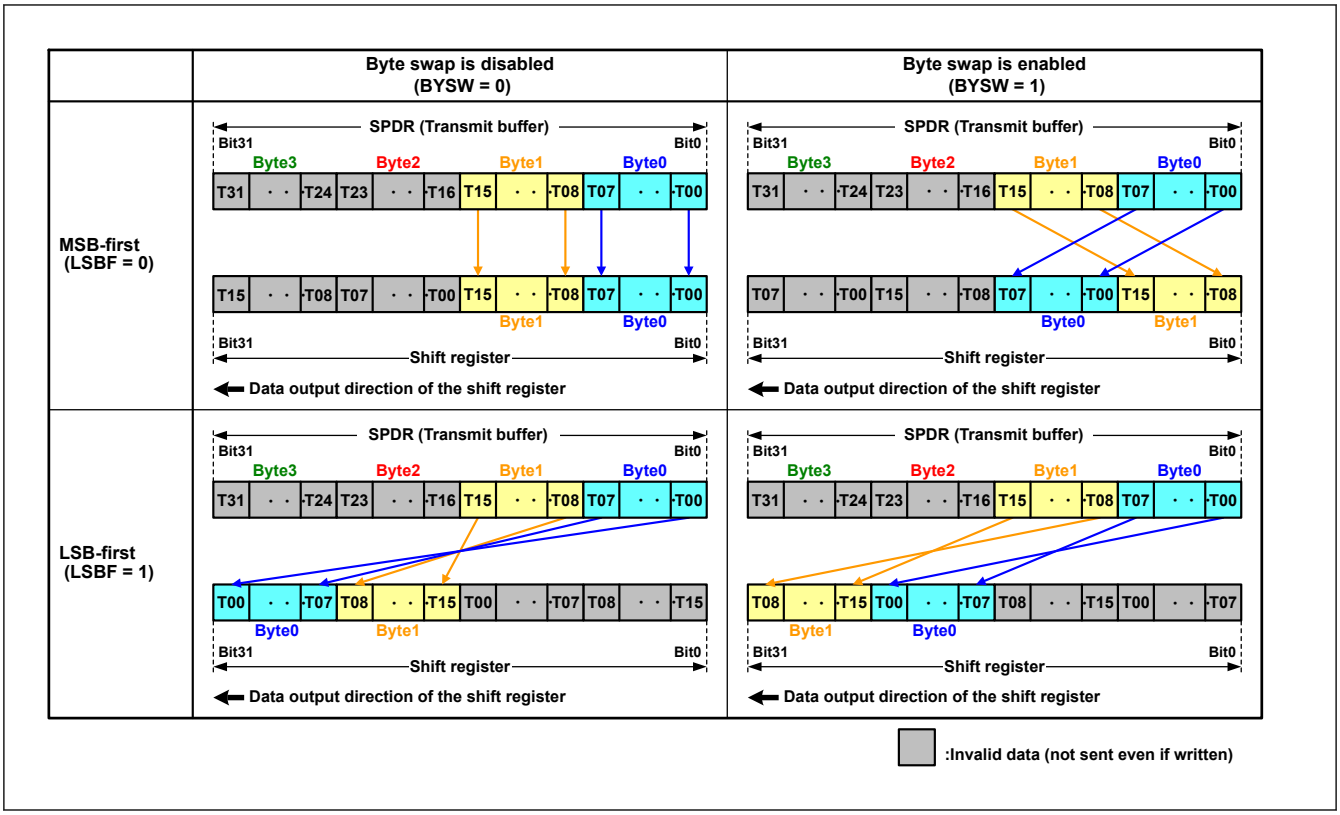


Figure 36.24 Byte swap with MSB/LSB transfer (16bit)

Note:
1. When using the byte swap, set 16 bits or 32 bits to the data length (SPCMDm.SPB[4:0] setting). If setting the other length, the behavior is not guaranteed.
2. When the byte swap is valid, set the parity function as invalid (SPCR.SPPE bit = 0). If setting the parity function as valid (SPPE bit = 1), the behavior is not guaranteed.
3. Set SPDCR.BYSW bit, when SPCR.SPE bit is 0. If rewriting BYSW bit, when SPE bit is 1, the behavior after it is not guaranteed.

36.3.4.4 Byte Swap Reception

When byte swap is enabled, the data in the shift register, swapped in 8-bit units, is copied to the receive buffer. Figure 36.25 shows the relationship between the shift register and SPDR (reception buffer) when transferring data with a 32-bit data length, using a combination of MSB / LSB first and with / without byte swap.

(1) MSB-first transfer (When the byte swap is disabled)

The first received data (R31) is stored in bit 0 of the shift register, and received data is shifted in the order of R31 → R30 → … → R00.

When necessary RSPCK cycles are input and data is stored from Byte3 [R31 to R24] to Byte0 [R07 to R00], the shift register value is copied to the receive buffer.

(2) MSB-first transfer (When the byte swap is enabled)

The first received data (R07) is stored in bit 0 of the shift register, and received data is shifted in the order of R07 → R06 → … → R00 → R15 → R14 → … → R08 → R23 → R22 → … → R16 → R31 → R30 → … → R24.

When necessary RSPCK cycles are input and data is stored from Byte0 [R07 to R00] to Byte3 [R31 to R24], byte values in the shift register are reversed in byte units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

(3) LSB-first transfer (When the byte swap is disabled)

The first received data (R00) is stored in bit 0 of the shift register, and received data is shifted in the order of R00 → R01 → … → R31.

When necessary RSPCK cycles are input and data is stored from Byte0 [R00 to R07] to Byte3 [R24 to R31], bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

(4) LSB-first transfer (When the byte swap is enabled)

The first received data (R24) is stored in bit 0 of the shift register, and received data is shifted in the order of R24 → R25 → … → R31 → R16 → R17 → … → R23 → R08 → R09 → … → R15 → R00 → R01 → … → R07.

When necessary RSPCK cycles are input and data is stored from Byte3 [R24 to R31] to Byte0 [R00 to R07], bit values of each byte in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

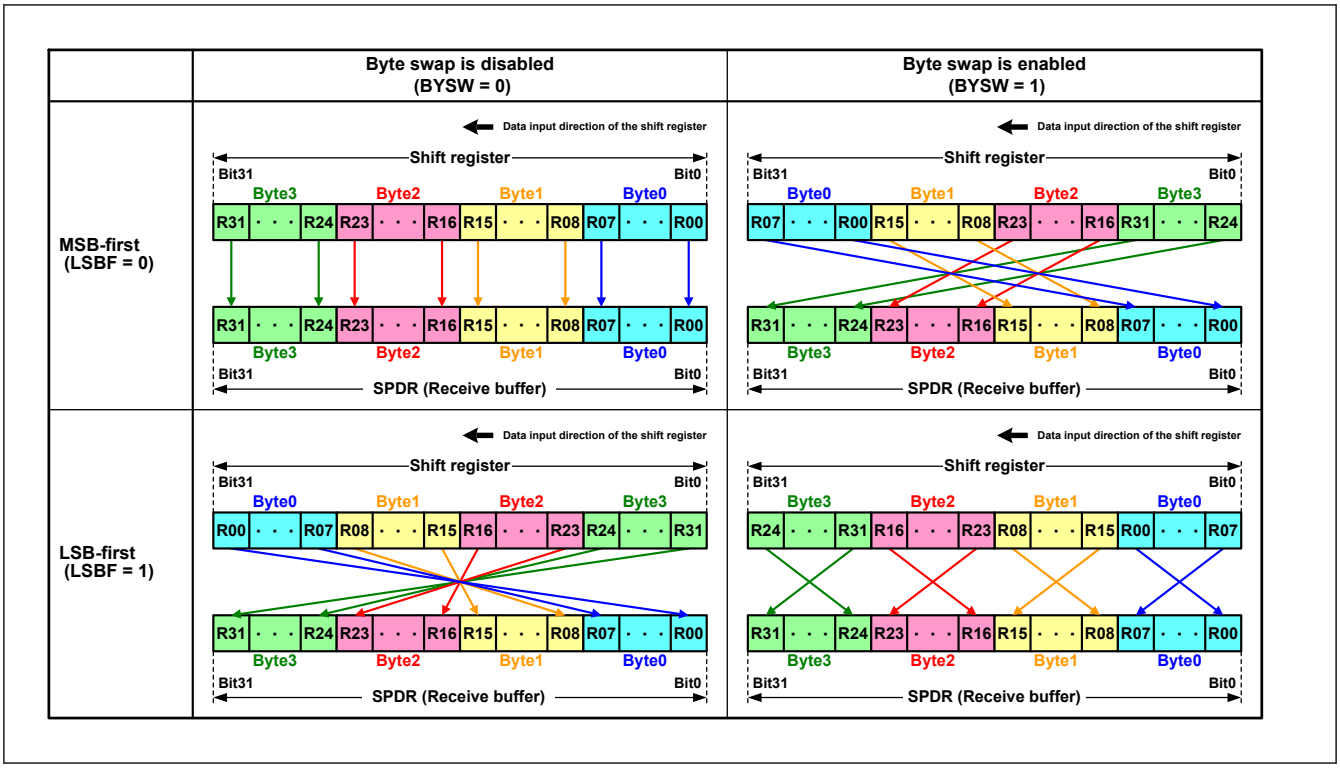


Figure 36.25 Byte swap with MSB/LSB transfer (32-bit)

Figure 36.26 shows the relationship between the shift register and SPDR (reception buffer) when transferring data with a 16-bit data length, using a combination of MSB / LSB first and with / without byte swap.

1. MSB-first transfer (when the byte swap is disabled)
   The first received data (R15) is stored in bit 0 of the shift register, and received data is shifted in the order of R15 → R14 → … R00. When necessary RSPCK cycles are input and data is stored from Byte3 [R31 to R24] to Byte0 [R07 to R00], the shift register value is copied to the receive buffer.
2. MSB-first transfer (when the byte swap is enabled)
   The first received data (R07) is stored in bit 0 of the shift register, and received data is shifted in the order of R07 → R06 → … R00 → R15 → R14 → … R08. When necessary RSPCK cycles are input and data is stored from Byte0 [R07 to R00] to Byte1 [R15 to R08], byte values in the shift register are reversed in byte units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].
3. LSB-first transfer (when the byte swap is disabled)
   The first received data (R00) is stored in bit 15 of the shift register, and received data is shifted in the order of R00 → R01 → … R07 → R08 → R09 → … R15. When necessary RSPCK cycles are input and data is stored from Byte0 [R00 to R07] to Byte1 [R08 to R15], bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07-R00].
4. LSB-first transfer (when the byte swap is enabled)
   The first received data (R08) is stored in bit 15 of the shift register, and received data is shifted in the order of R08 → R09 → … R15 → R00 → R01 → … R07. When necessary RSPCK cycles are input and data is stored from Byte1 [R08

to R15] to Byte0 [R00 to R07], bit values of each byte in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

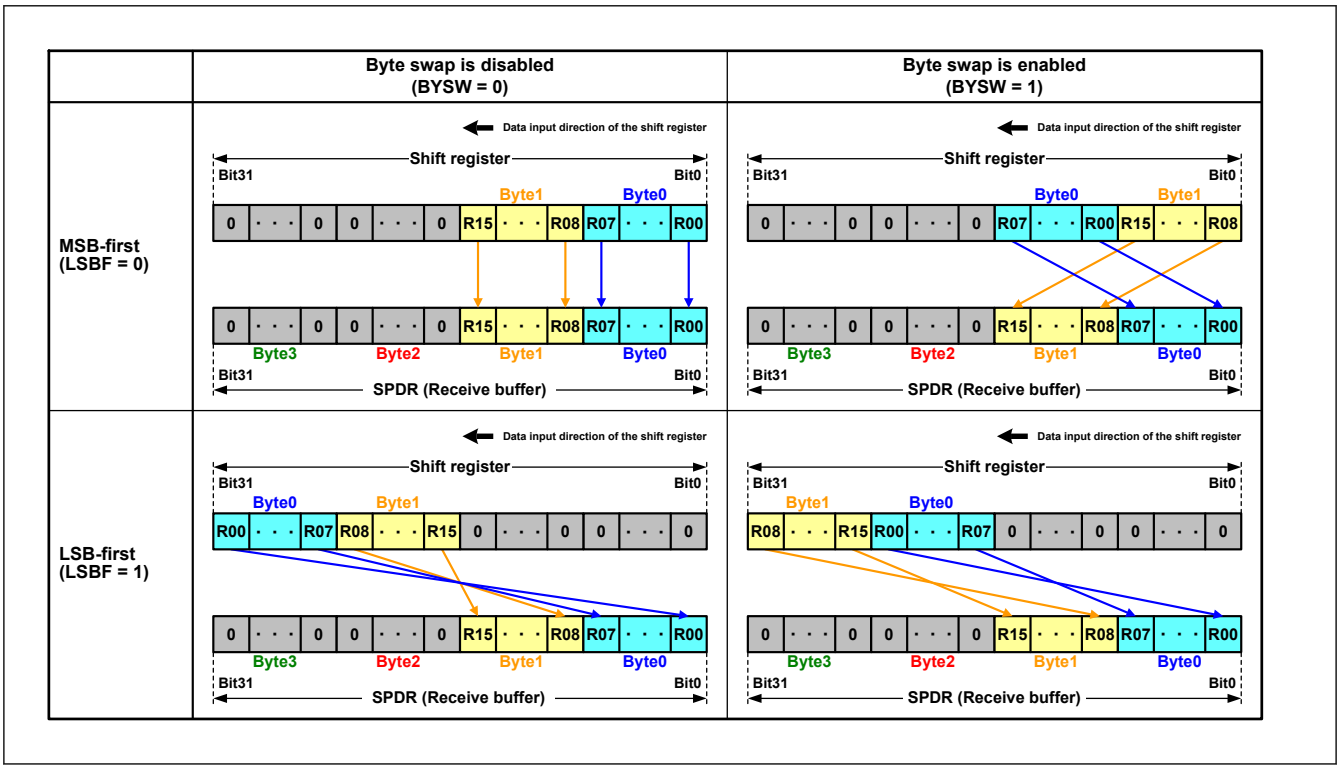


Figure 36.26 Byte swap with MSB/LSB transfer(16-bit)

Note: 1. When using the byte swap, set 16 bits or 32 bits to the data length (SPCMDm.SPB[4:0] setting). If setting the other length, the behavior is not guaranteed.
2. When the byte swap is valid, set the parity function as invalid (SPCR.SPPE bit = 0). If setting the parity function as valid (SPPE bit = 1), the behavior is not guaranteed.
3. Set SPDCR.BYSW bit, when SPCR.SPE bit is 0. If rewriting BYSW bit, when SPE bit is 1, the behavior after it is not guaranteed.

36.3.5 Transfer Formats

36.3.5.1 When CPHA = 0

Figure 36.27 shows an example transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 0. Do not perform clock synchronous operation (SPCR.SPMS = 1) when the SPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 36.27, RSPCKn (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0, and RSPCKn (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the SPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the SPI settings. For details, see section 36.3.2. Controlling the SPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIn and MISOn signals begins at an SSLni signal assertion. The first RSPCKn signal change that occurs after the SSLni signal assertion becomes the first transfer data fetch. After this, data is sampled every 1 RSPCKn cycle. The change timing for MOSIn and MISOn signals is 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCKn signal operation timing as it only affects the signal polarity.

t1 denotes the RSPCK delay, the period from an SSLni signal assertion to RSPCKn oscillation. t2 denotes the SSL negation delay, the period from the termination of RSPCKn oscillation to an SSLni signal negation. t3 denotes the next-access delay, the period in which SSLni signal assertion is suppressed for the next transfer after the end of serial transfer. t1, t2, and t3 are controlled by a master device running on the SPI system. For a description of t1, t2, and t3 when the SPI is in master mode, see section 36.3.12.1. Master mode operation.

[In the Motorola-SPI case]

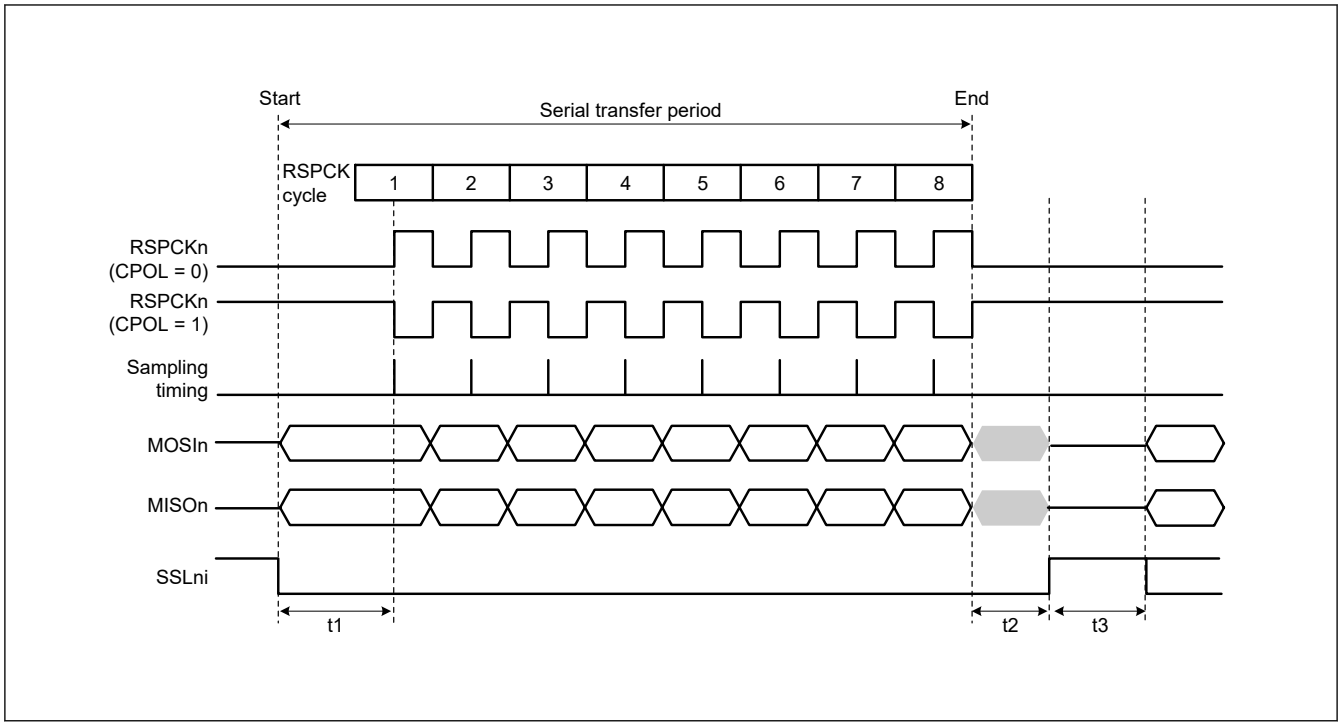


Figure 36.27 SPI transfer format when CPHA = 0, SPFRF = 0

[In TI-SSP case]

Not supported in CPHA = 0

36.3.5.2 When CPHA = 1

Figure 36.28 shows an example transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 1. However, when the SPCR.SPMS bit is 1, the SSLni signals are not used, and only the three signals RSPCKn, MOSIn, and MISOn handle communications. In Figure 36.28, RSPCKn (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0 and RSPCKn (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the SPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the SPI mode (master or slave mode). For details, see section 36.3.2. Controlling the SPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISOn signal begins at an SSLni signal assertion. The output of valid data to the MOSIn and MISOn signals begins at the first RSPCKn signal change that occurs after the SSLni signal assertion. After this, data is updated every 1 RSPCK cycle. The transfer data fetch timing is 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCKn signal operation timing. It only affects the signal polarity.

t1, t2, and t3 are the same as those when CPHA = 0. For a description of t1, t2, and t3 when the SPI of the MCU is in master mode, see section 36.3.12.1. Master mode operation.

[In the Motorola-SPI case]

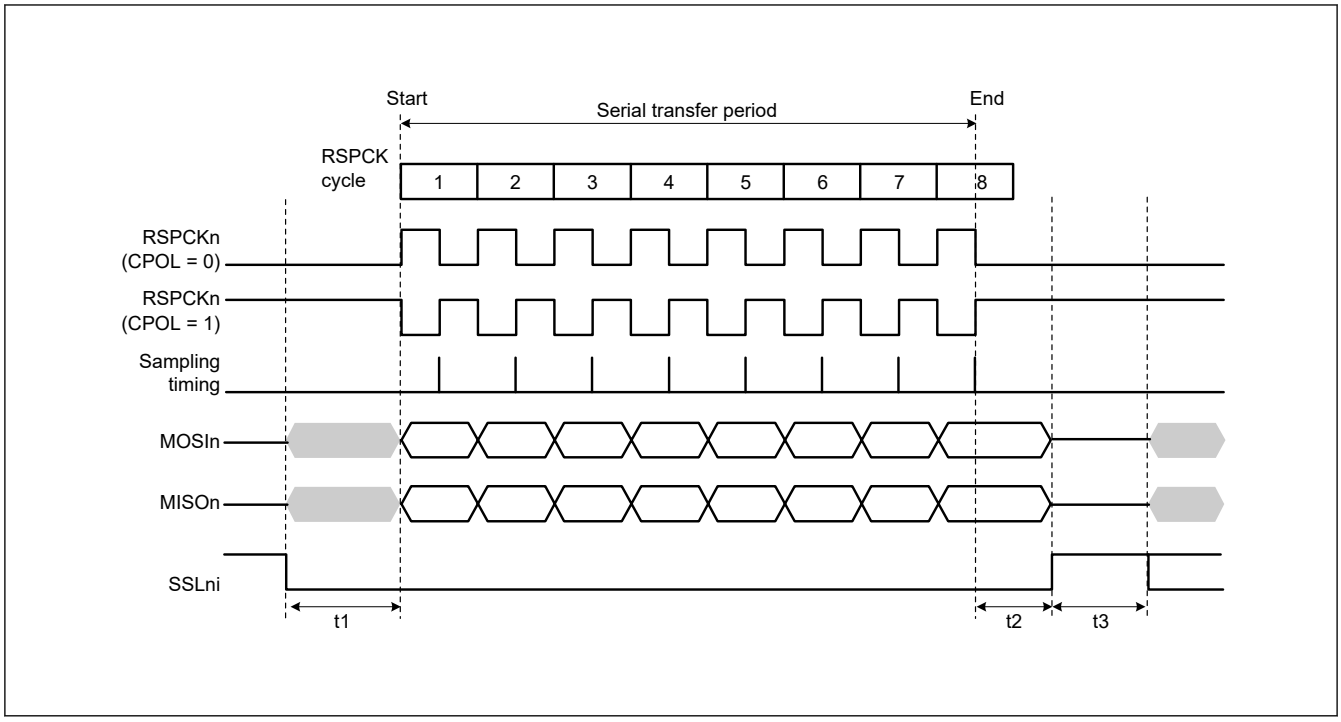


Figure 36.28 SPI transfer format when CPHA = 1, SPFRF = 0

[In the TI-SSP case]

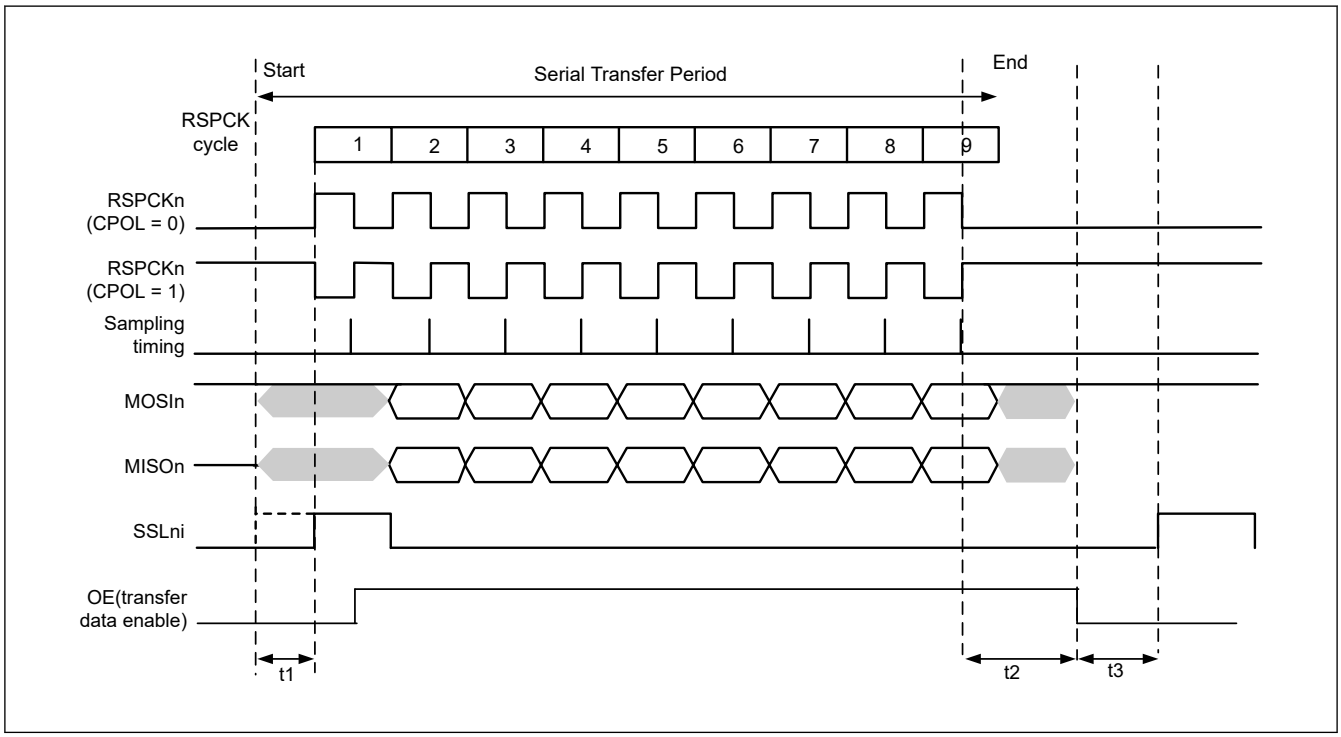


Figure 36.29 SPI transfer format when CPHA = 1, SPFRF = 1

36.3.6 Communication Operating Mode

Transmit-Receive serial communication, transmit-only operation, and Receive-only operation are selected by setting the Communication Mode Select bits (TXMD [1:0]) of the SPI control register (SPCR).

SPDR access described in Figure 36.30, Figure 36.31, Figure 36.32 shows an access to the SPI data register (SPDR). W shows a write cycle.

36.3.6.1 Transmit-Receive Serial Communication (TXMD[1:0] = 00b)

Figure 36.30 shows an example of operation when the communication mode select bit (TXMD[1:0]) in the SPI control register (SPCR) is set to 00b. In the example, the SPI performs an 8-bit serial transfer when the SPDCR2.TTRG is 0, the SPDCR2.RTRG is FIFO stage - 1, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

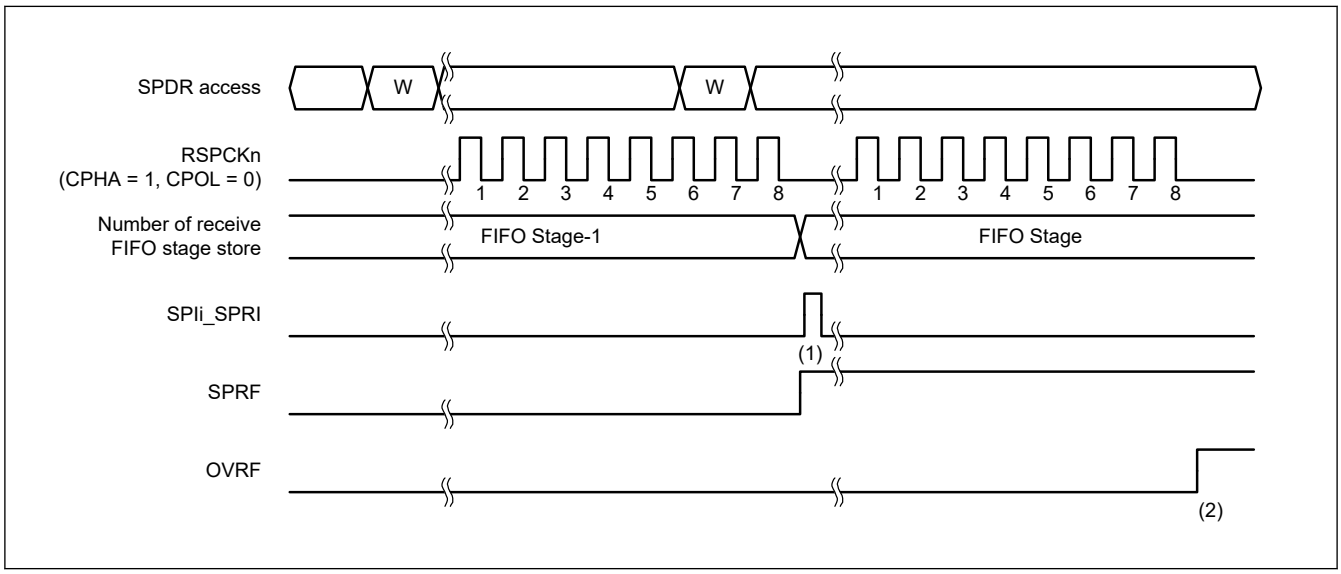


Figure 36.30 Operation example when SPCR.TXMD[1:0] = 00b

The operation of the flags at timings (1) and (2) in Figure 36.30 is as follows:

1. When serial transfer ends while the number of SPDR receive buffer store matches the number of frames set in SPDCR2.RTRG, the SPI generates a receive buffer full interrupt request (SPIi_SPRI), the SPI sets the SPSR.SPRF flag to 1, and copies the received data in the shift register to the receive buffer.
2. When serial transfer ends with data for the number of FIFO stages stored in the SPDR receive buffer, the SPI sets the SPSR.OVRF flag to 1, and discards the received data in the shift register. For details about the operation of the SPSR.OVRF flag, see section 36.3.10.1. Overrun errors.

In Transmit-Receive serial communication (TXMD[1:0] = 00b), transmit data is transmitted and receive data is received. Therefore, the SPRF flag and the OVRF flag are set to 1 at timings (1) and (2) respectively.

36.3.6.2 Transmit-Only Serial Communications (TXMD[1:0] = 01b)

Figure 36.31 shows an example of operation when the communication mode select bit (TXMD[1:0]) in the SPI control register (SPCR) is set to 01b. In this example, the SPI performs an 8-bit serial transfer when the SPDCR2.TTRG is 0, the SPDCR2.RTRG is 0, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

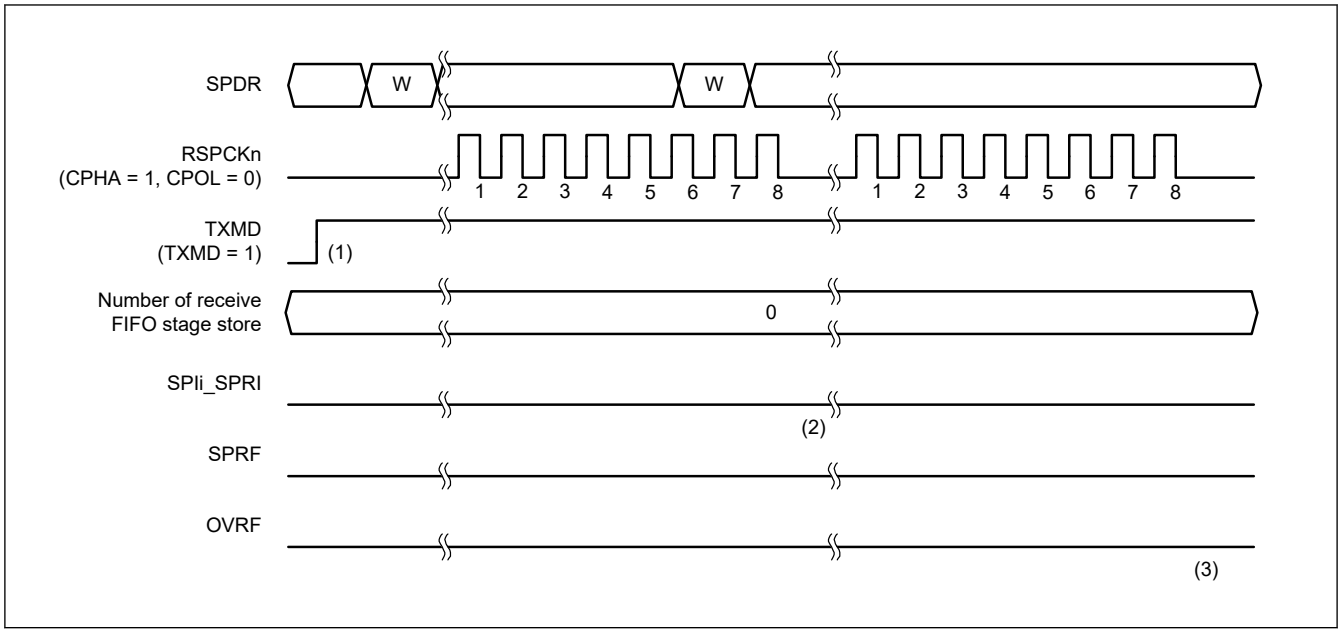


Figure 36.31 Operation example when SPCR.TXMD[1:0] = 01b

The operation of the flags at timings (1) to (3) in Figure 36.31 is as follows:

1. Make sure there is no data left in the receive buffer (the SPSR.SPRF flag is 0) and the SPSR.OVRF flag is 0 before entering transmit-only mode (SPCR.TXMD[1:0] = 01b).
2. When a serial transfer ends without receiving data in the receiving FIFO of SPDR, if the transmit-only mode is selected (SPCR.TXMD[1:0] = 01b), the SPSR.SPRF flag retains the value of 0, and the SPI does not copy the data in the shift register to the receive buffer.
3. Because the receive buffer of SPDR does not hold data that was received in the previous serial transfer, even when a serial transfer ends, the SPSR.OVRF flag retains the value of 0, and the data in the shift register is not copied to the receive buffer.

In transmit-only mode (SPCR.TXMD[1:0] = 01b), the SPI transmits data but does not receive data. Therefore, the SPSR.SPRF and SPSR.OVRF flags remain 0 at timings (1) to (3).

36.3.6.3 Receive-Only Serial Communication (TXMD[1:0] = 10b)

Figure 36.32 shows an example of operation when the communication mode select bit (TXMD[1]) in the SPI control register (SPCR) is set to 1. In this example, the SPI performs an 8-bit serial transfer when the SPDCR2.TTRG is FIFO stage - 1, the SPDCR2.RTRG is 0, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

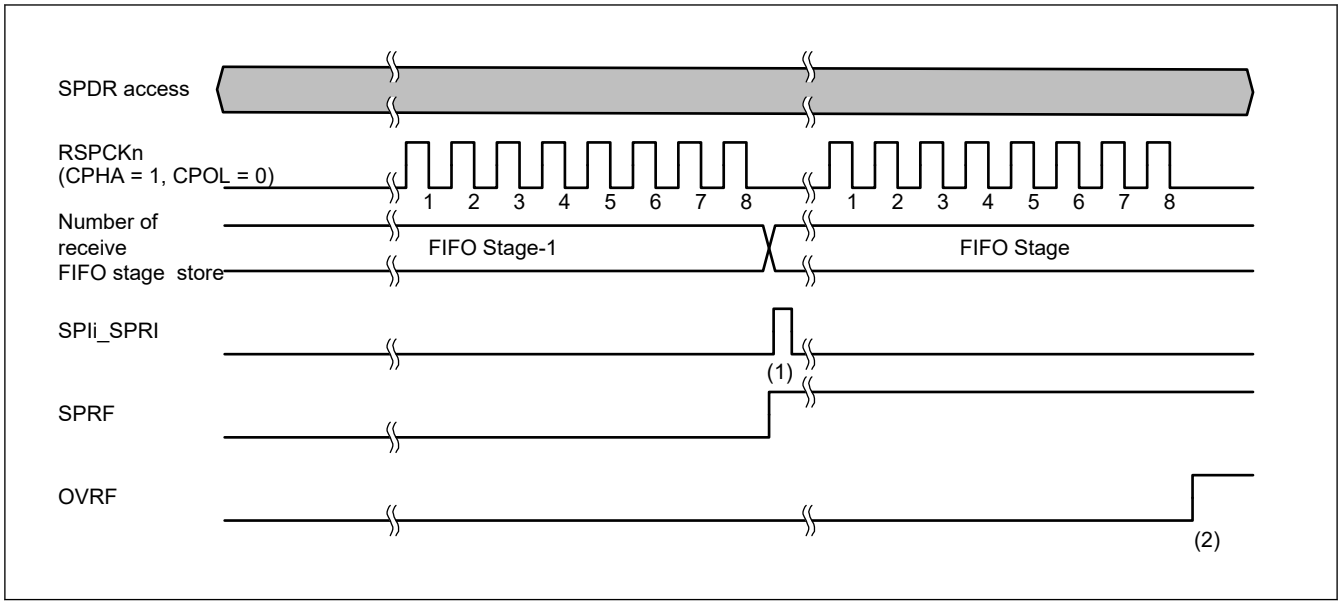


Figure 36.32 Example of Operation when SPCR.TXMD[1:0] = 10b

The following describes operation of flags at timings (1) and (2) in the figure above.

1. When serial transfer ends while the number of SPDR receive buffer store matches the number of frames set in SPDCR2.RTRG, the SPI generates a receive buffer full interrupt request (SPIi_SPRI), the SPI sets the SPSR.SPRF flag to 1, and copies the received data in the shift register to the receive buffer
2. When serial transfer ends with data for the number of FIFO stages stored in the SPDR receive buffer, the SPI sets the SPSR.OVRF flag to 1 and discards the received data in the shift register.

36.3.7 Transmit Buffer Empty and Receive Buffer Full Interrupts

Figure 36.33 show examples of operation of the transmit buffer empty interrupt (SPIi_SPTI (i = 0, 1)) and the receive buffer full interrupt (SPIi_SPRI). The SPDR register accesses shown in these figures indicate the conditions of access to the register, where W denotes a write cycle and R a read cycle. In Figure 36.33, the SPI performs an 8-bit serial transfer when SPCR.TXMD[1:0] bits are 00b, the SPDCR2.TTRG bit is 0, the SPDCR2.RTRG bit is 0, the SPCMDm.CPHA bit is 0, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

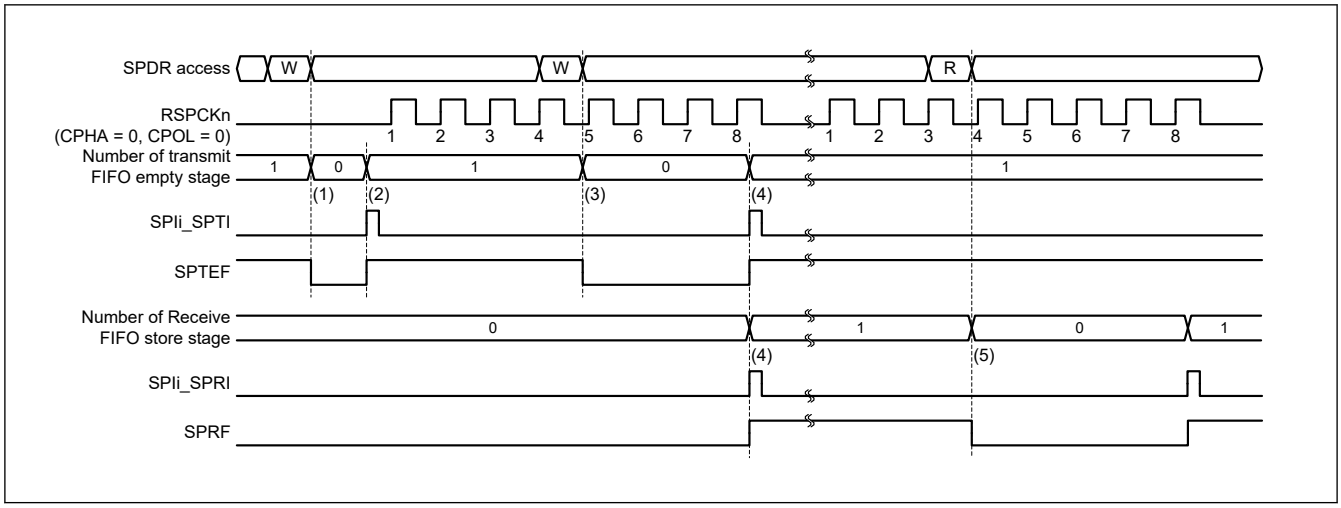


Figure 36.33 Operation example of the SPIi_SPTI and SPIi_SPRI interrupts when CPHA = 0 and CPOL = 0 in master mode

The operation of the SPI at timings (1) to (5) in Figure 36.33 is as follows:

1. When transmit data is written to SPDR with the transmit buffer of SPDR is before the next transfer data is set, the SPI writes data to the transmit buffer. When transmit data is written to SPDR in one processing routine using DTC / DMAC, the SPSR.SPTEF flag is cleared to 0 at the last access.
2. If the shift register is empty, the SPI copies the data in the transmit buffer to the shift register. At this time, if transmit FIFO empty stage number > TTRG value, then the SPI generates a transmit buffer empty interrupt request (SPIi_SPTI), and sets the SPSR.SPTEF flag to 1. How a serial transfer is started depends on the SPI mode. For details, see section 36.3. Operation, and section 36.3.13. Clock Synchronous Operation.
3. When transmit data is written to SPDR either by the transmit buffer empty interrupt routine, or by the processing of the transmit buffer empty using the SPTEF flag, the SPI writes data to the transmit buffer. When the transmit data is written to SPDR in one processing routine using DTC / DMAC, the SPTEF flag is cleared to 0 at the last access. Because the data being transferred serially is stored in the shift register, the SPI does not copy the data in the transmit buffer to the shift register.
4. When the serial transfer ends with the receive buffer of SPDR > FIFO stage number, the SPI copies the receive data in the shift register to the receive buffer, generates a receive buffer full interrupt request (SPIi_SPRI), and sets the SPRF flag to 1. Because the shift register becomes empty on completion of the serial transfer, if the next transfer data is set in the transmit FIFO before the serial transfer ended, the SPI sets the SPTEF flag to 1 and copies data in the transmit buffer to the shift register. Even when received data is not copied from the shift register to the receive buffer in an overrun error status, on completion of the serial transfer, the SPI determines that the shift register is empty, so data transfer from the transmit buffer to the shift register is enabled.
5. When SPDR is read either by the receive buffer full interrupt routine or processing of the receive buffer full interrupt using the SPRF flag, the receive data can be read. If the received data is read from SPDR in one processing routine using DTC / DMAC, the SPRF flag is cleared to 0 at the last access.

When transmit data is written to the SPDR register while no empty stages in the transmit FIFO, the SPI does not update data in the transmit buffer. When writing to SPDR, always use either a transmit buffer empty interrupt request or check the empty or processing of the transmit buffer empty interrupt using the SPTEF flag. To use a transmit buffer empty interrupt, set the SPTIE bit in SPCR to 1. If the SPI function is disabled (the SPCR.SPE bit is 0), set the SPTIE bit to 0.

When serial transfer ends while data is stored in the receive FIFO for the number of FIFO stages, the SPI does not copy data from the shift register to the receive buffer, and it detects an overrun error (see section 36.3.10. Error Detection). To prevent a receive data overrun error, read the received data using a receive buffer full interrupt request before the next serial transfer ends. To use an SPI receive buffer full interrupt, set the SPCR.SPRIE bit to 1.

Transmission and reception interrupts or the associated IELSRn.IR flags (where n is the interrupt vector number) in the ICU can be used to confirm the states of the transmit and receive buffers.

Similarly, the SPTEF and SPRF flags can be used to confirm the states of the transmit and receive buffers. See section 13, Interrupt Controller Unit (ICU) for the interrupt vector numbers.

36.3.8 Idle Interrupt

When the SPCP[2:0] of the SPI status register (SPSR) becomes 000b (start of sequence control), the IDLNF flag in the SPI status register (SPSR) is set to 1 and an idle interrupt request is made during master mode operation. An interrupt request is also made by clearing the SPCR.SPE bit to 0.

[In the Motorola-SPI case]

Figure 36.34 shows an example of idle interrupt operation during normal operation.

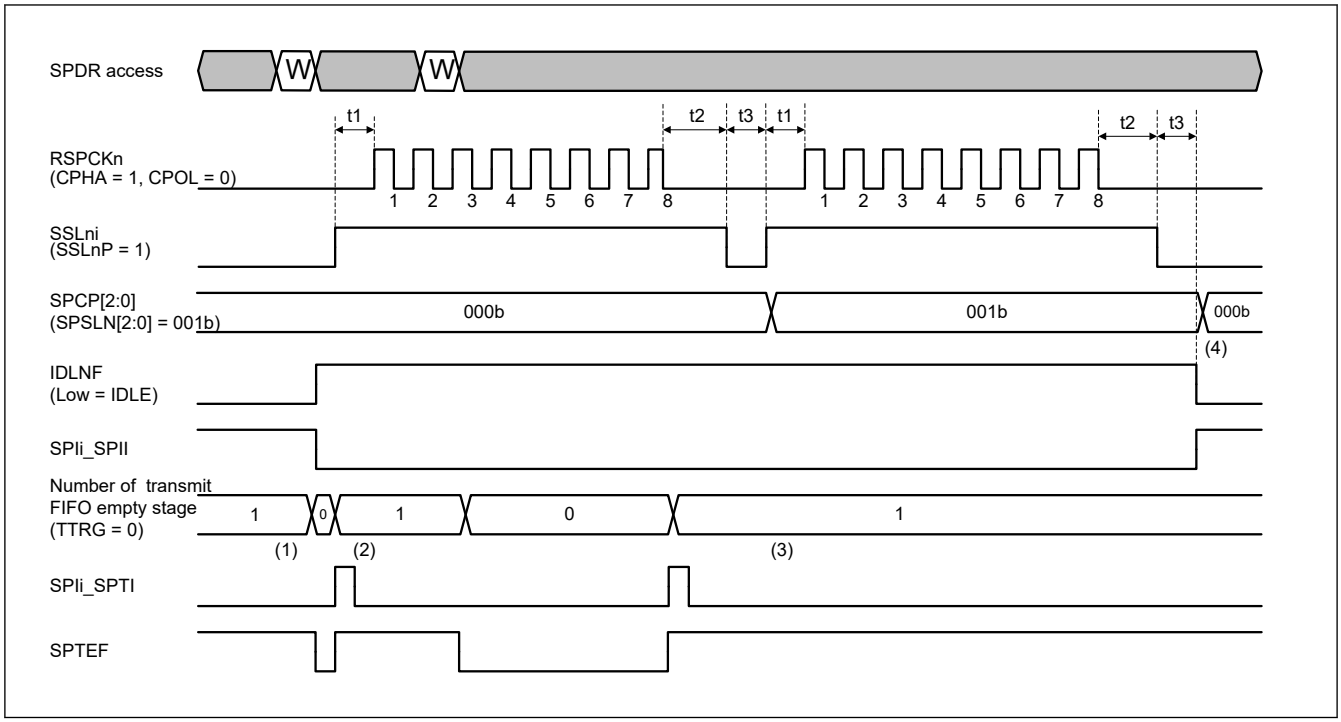


Figure 36.34 Example of Idle Interrupt Operation (Master mode / Motorola-SPI)

1. At the start of transmission, if the next transfer data is not set in the transmission buffer, the IDLNF flag is 0 (IDLE). Writing transmit data sets the IDLNF flag to 1 (BUSY). When the SPIIE bit in the SPI control register (SPCR) is set to 1 before transmit data is written, interrupt processing is required before transmission start. For this reason, set the SPIIE bit to 0 before starting transmission.
2. After transmission has started, the IDLNF flag remains 1 (BUSY) regardless of the transmit buffer state.
3. The SPCP[2:0] bits change the command to the next command at the end of t3 cycle. When the next command is not 000b, the IDLNF flag remains unchanged even when the next transmit data has not been written.
4. The IDLNF flag is cleared to 0 (IDLE) at the end of t3 cycle because the next command is 000b and the next transmit data is not present. When the SPIIE bit is 1 currently, an SPIi_SPII (i = 0, 1) interrupt is output.

[In the TI-SSP case]

Figure 36.35 shows an example of idle interrupt operation during normal operation.

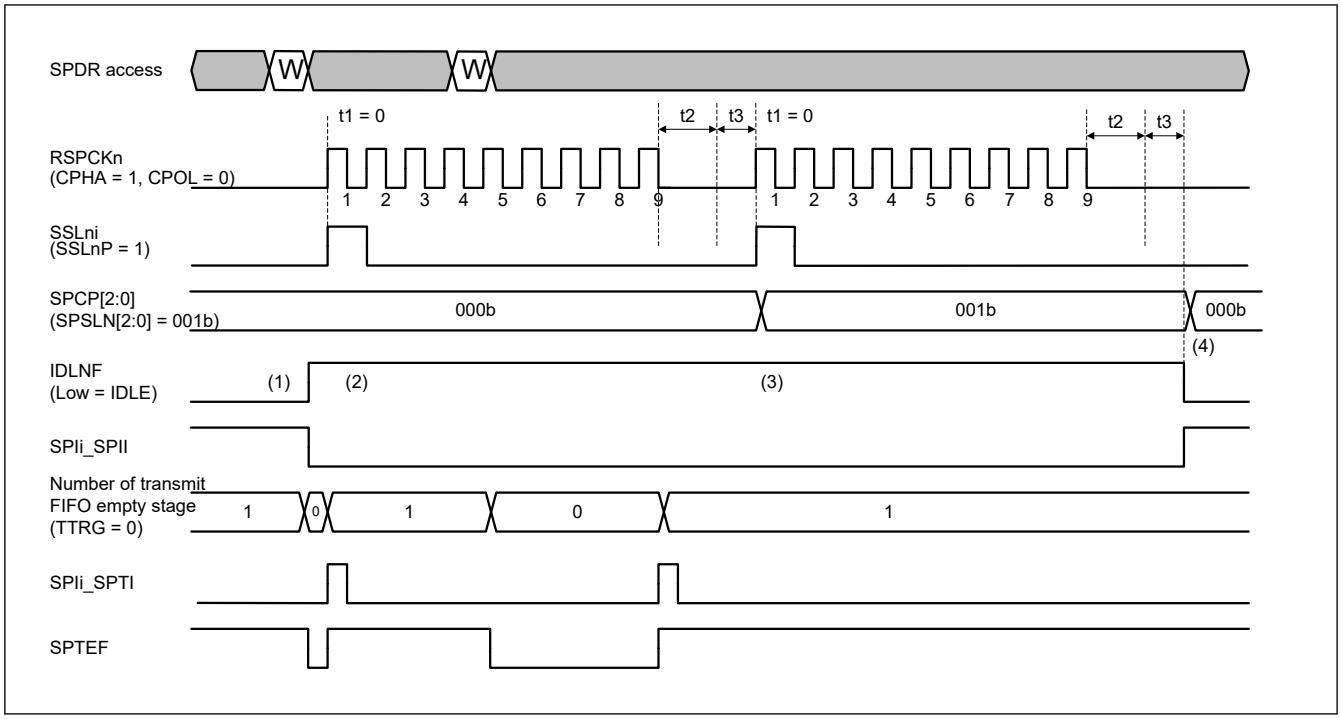


Figure 36.35 Example of Idle Interrupt Operation (Master mode / TI-SSP)

1. At the start of transmission, if the next transfer data is not set in the transmission buffer, the IDLNF flag is 0 (IDLE). Writing transmit data makes sets the IDLNF flag to 1 (BUSY). When the SPIIE bit in the SPI control register (SPCR) is set to 1 before transmit data is written, interrupt processing is required before transmission start. For this reason, set the SPIIE bit to 0 before starting transmission.
2. After transmission has started, the IDLNF flag remains 1 (BUSY) regardless of the transmit buffer state.
3. The SPCP[2:0] bits change the command to the next command at the end of t3 cycle. When the next command is not 000b, the IDLNF flag remains unchanged even when the next transmit data has not been written.
4. The IDLNF flag is cleared to 0 (IDLE) at the end of t3 cycle because the next command is 000b and the next transmit data is not present. When the SPIIE bit is 1 currently, an SPIi_SPII interrupt is output.

36.3.9 Communication End Interrupt

36.3.9.1 Transmit-Receive/Transmit in Master Mode

See the description of the CENDF bit in section 36.2.9. SPSR : SPI Status Register for the setting / clearing conditions of the communication completion flag during Transmit-Receive/Transmit-only in Master Mode.

[In the Motorola-SPI case]

Figure 36.36 shows an example of communication end interrupt operation during transmit-receive/transmit master mode.

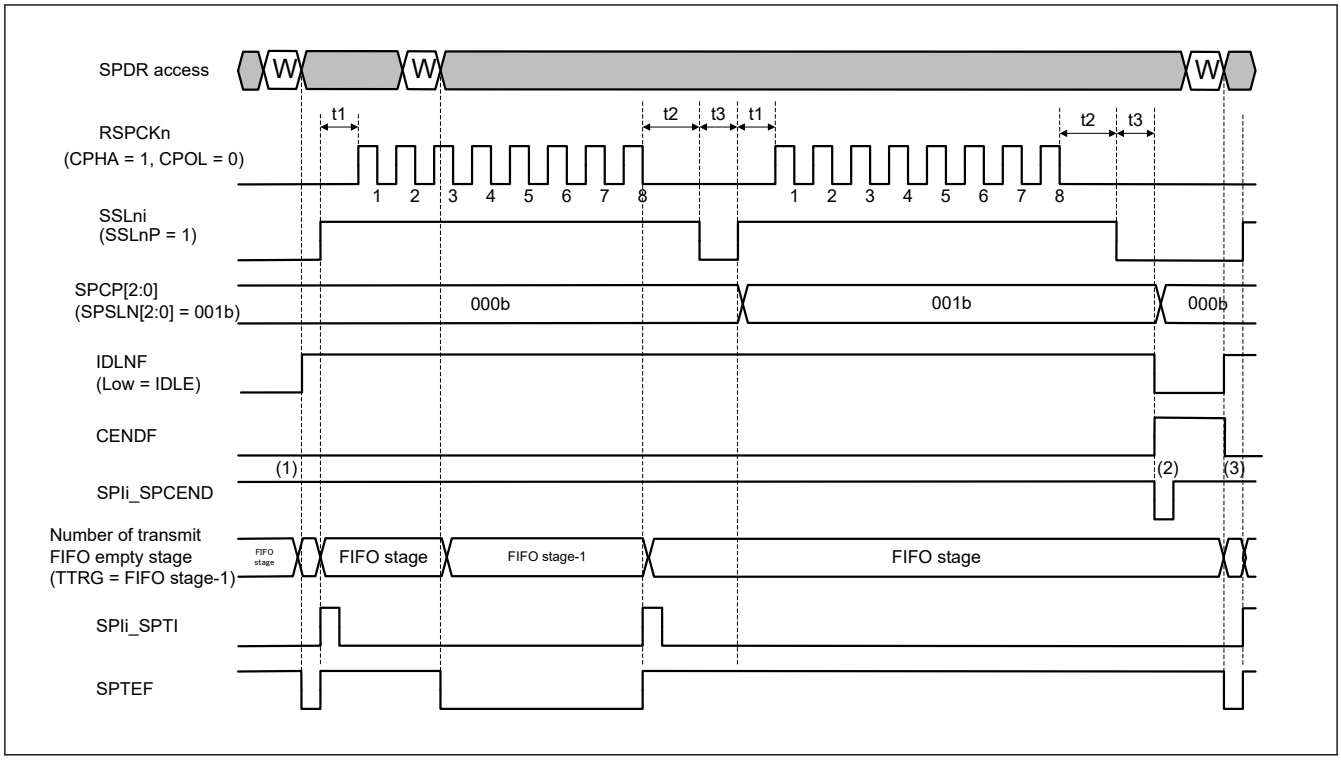


Figure 36.36 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit Master mode/Motorola-SPI)

1. The CENDF flag is 0 and the level of SPIi_SPCEND (i = 0, 1) is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the end of t3 cycle because the next command is 000b and there is no next transmit data, and then the SPIi_SPCEND interrupt output when the CENDIE bit is 1.
3. The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

[In the TI-SSP case]

Figure 36.37 shows an example of communication end interrupt operation during transmit-receive/transmit-only master mode.

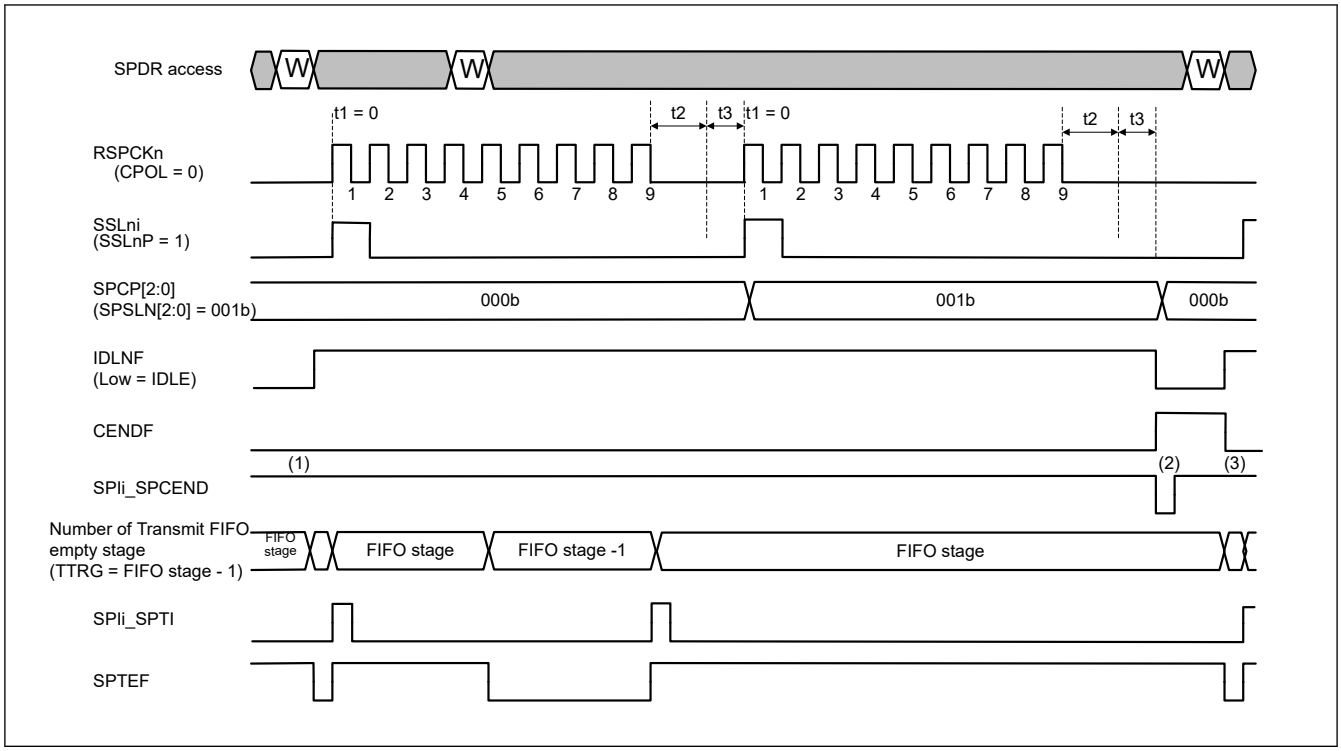


Figure 36.37 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit-only Master mode/TI-SSP)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the end of t3 cycle because the next command is 000b and there is no next transmit data, and then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

In slave mode operation, the output timing of the communication end interrupt is deferent due to the value of the SPCR.SPMS bit (SPI mode select bit), and the clear timing of the communication end interrupt is deferent due to the communication mode (transmit-receive or transmit-only or receive-only).

36.3.9.2 Receive-only in Master Mode

See the description of the CENDF bit in section 36.2.9. SPSR : SPI Status Register for the setting / clearing conditions of the communication completion flag during Receive-only in Master Mode.

Figure 36.38 shows an example of communication end interrupt operation during receive-only master mode at RMFM[4:0] = 0.

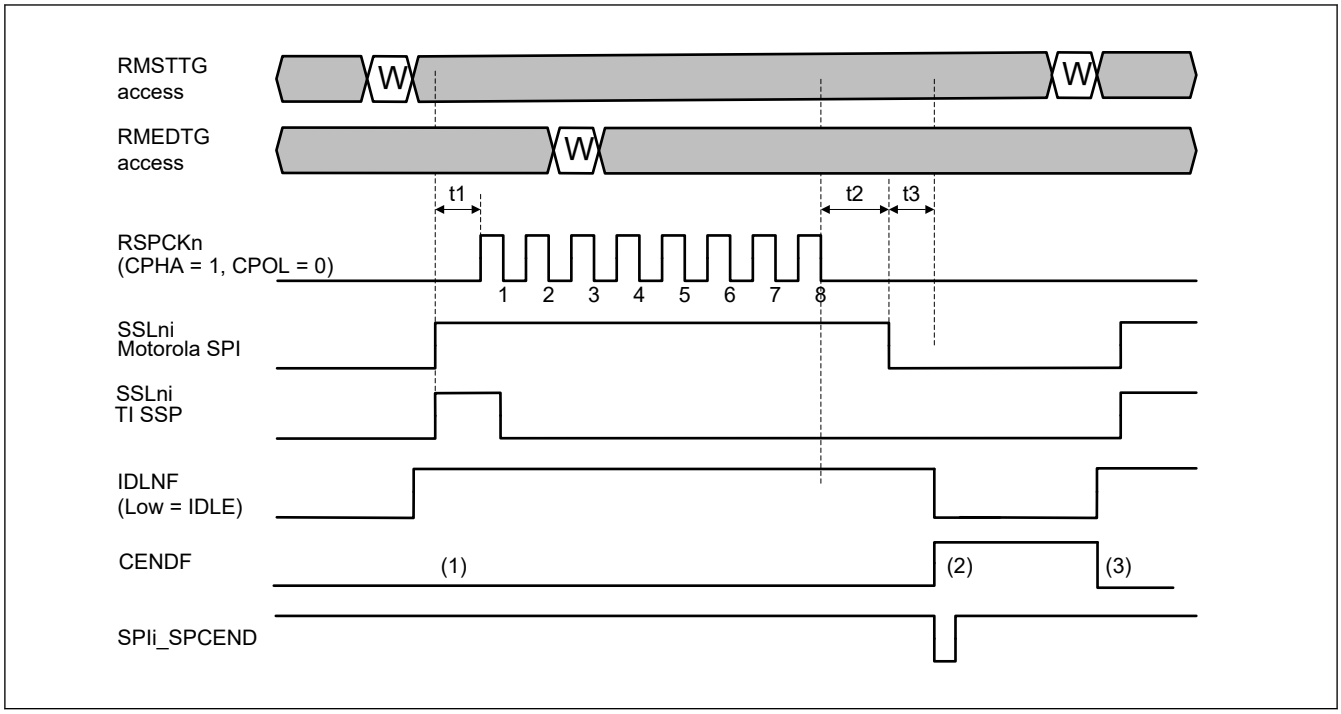


Figure 36.38 Example of Communication End Interrupt Operation (Receive-only Master mode / Motorola-SPI) at RMFM [4:0] = 0

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start. These have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the end of t3 cycle by writing 1 to RMEDTG during the communication frame. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when writing 1 to RMSTTG. Also when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

Figure 36.39 shows an example of communication end interrupt operation during receive-only master mode at RMFM[4:0] ≠ 0.

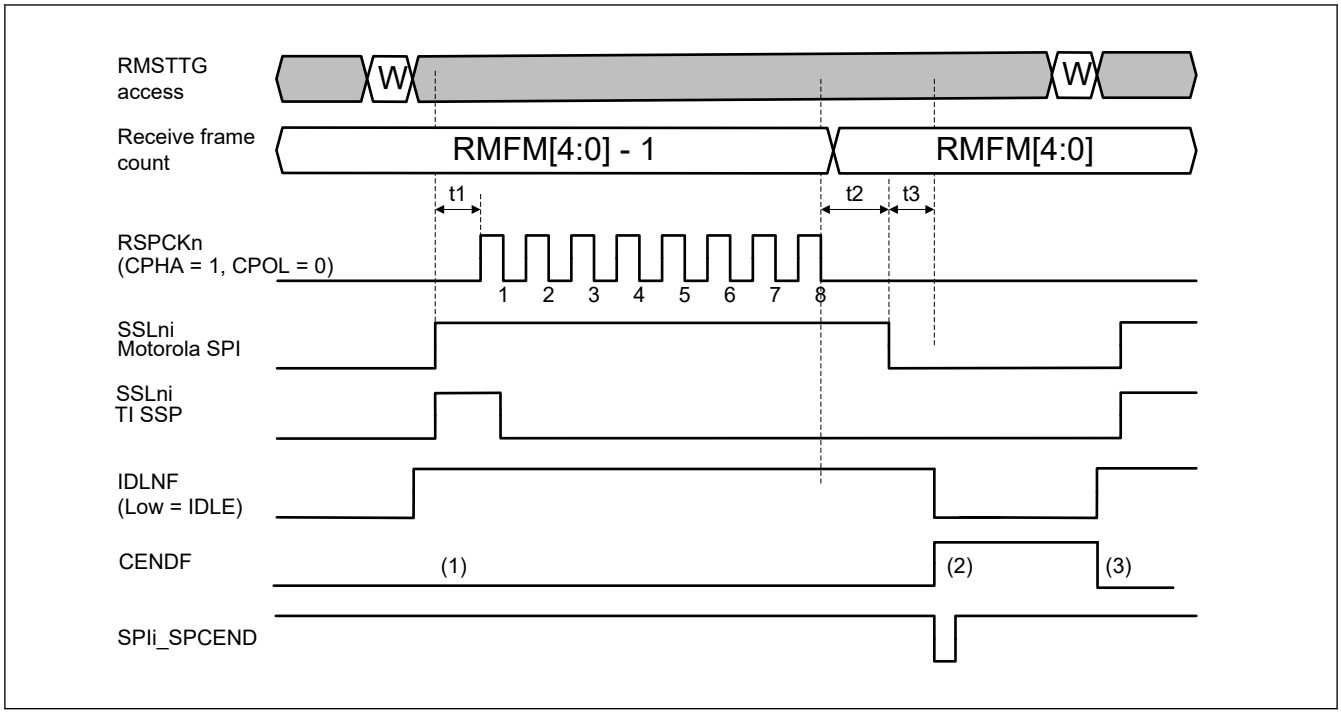


Figure 36.39 Example of Communication End Interrupt Operation (Receive-only Master mode / Motorola-SPI) at RMFM [4:0] ≠ 0

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start. These have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the end of t3 cycle after receiving the number of frames set by RMFM[4:0]. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when writing 1 to RMSTTG. Also when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

36.3.9.3 Transmit-Receive/Transmit in Slave Mode on SPI Operation (4-wire)

See the description of the CENDF bit in section 36.2.9. SPSR : SPI Status Register for the setting / clearing conditions of the communication completion flag during Transmit-Receive/Transmit-only in Slave Mode (4-wire).

[In the Motorola-SPI case]

Figure 36.40 shows an example of communication end interrupt operation during transmit-receive/transmit slave mode on SPI operation.

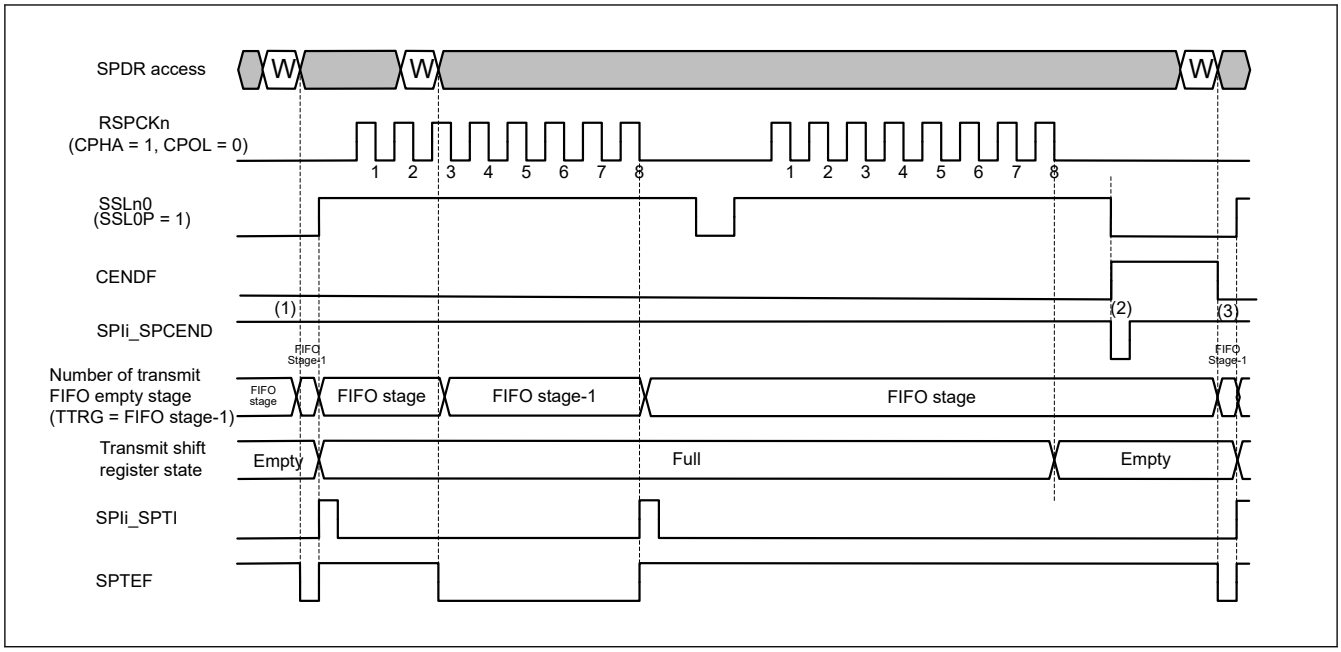


Figure 36.40 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit Slave mode on SPI Operation/Motorola-SPI)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the timing of SSLn0 negate when the next transfer data is not set in the transmit FIFO and the transmit shift register is empty. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

[In the TI-SSP case]

Figure 36.41 shows an example of communication end interrupt operation during transmit-receive/transmit-only slave mode on SPI operation.

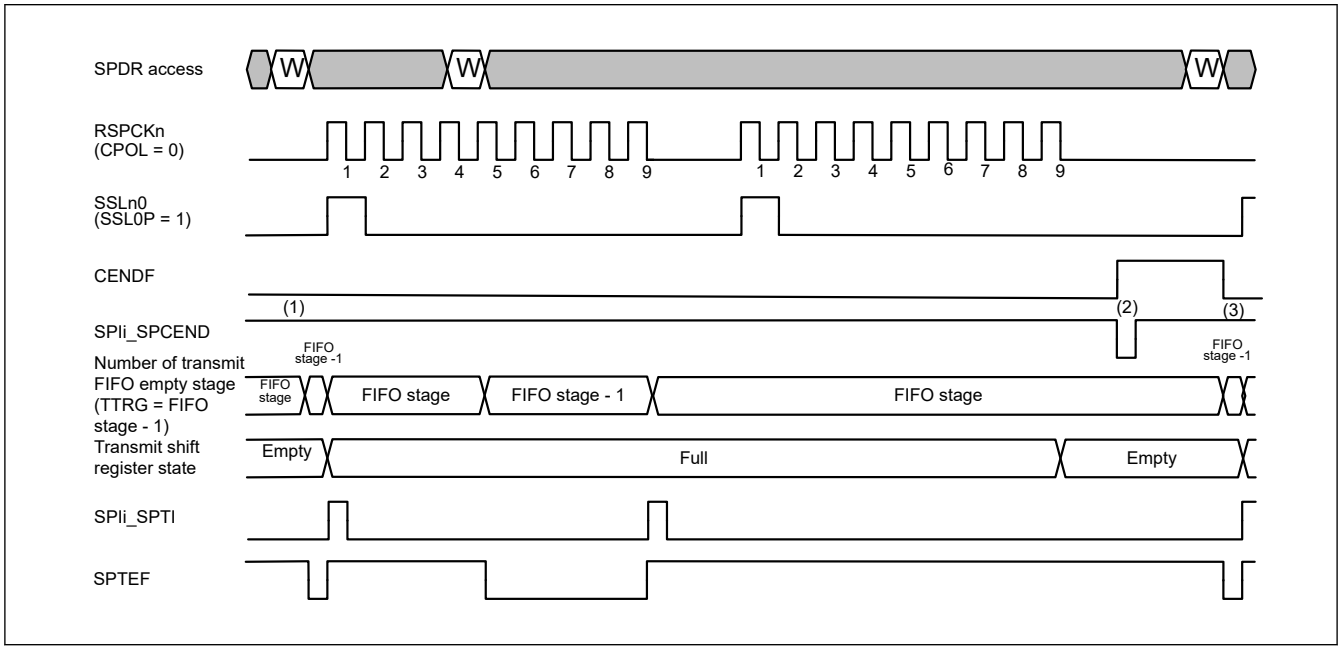


Figure 36.41 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit-only Slave mode on SPI Operation / TI-SSP)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the RSPCKn last data bit sampling when the next transfer data is not set in the transmit FIFO and the transmit shift register is empty. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

36.3.9.4 Receive Only in Slave Mode on SPI Operation (4-wire)

See the description of the CENDF bit in section 36.2.9. SPSR : SPI Status Register for the setting / clearing conditions of the communication completion flag during Receive-only in Slave Mode (4-wire).

[In the Motorola-SPI case]

Figure 36.42 shows an example of communication end interrupt operation during receive only slave mode on SPI operation (4-wire).

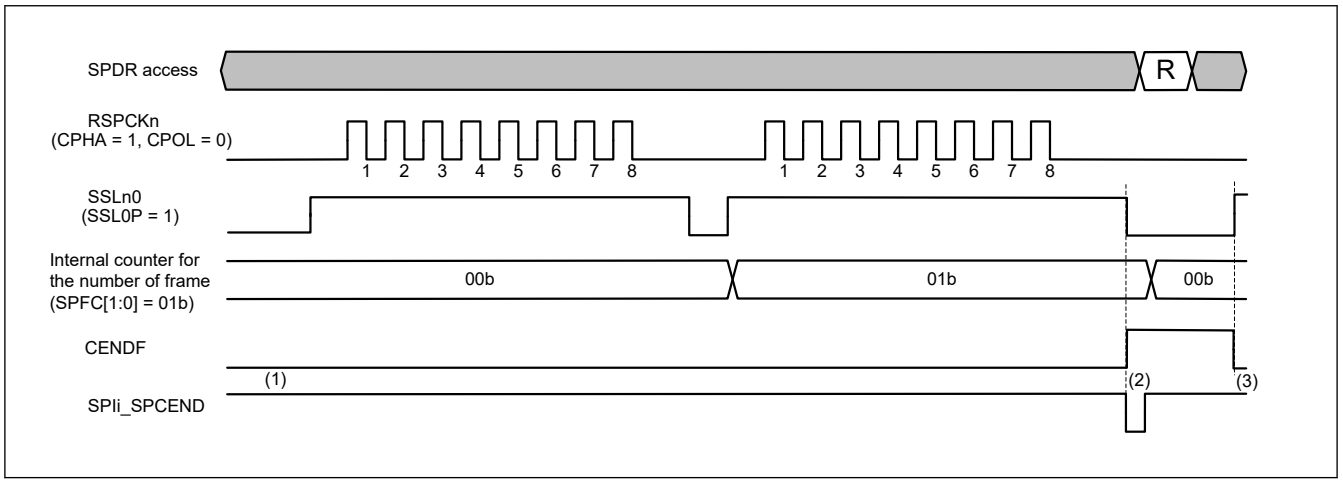


Figure 36.42 Example of Communication End Interrupt Operation (Receive only Slave mode on SPI Operation / Motorola-SPI)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.
2. After the frames for SPFC set value in the SPI data control register (SPDCR) are stored in the receive buffer, the CENDF flag becomes 1 (communication completed) at the timing of SSLn0 negation. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared at the SSLn0 assert when the next transmission start. Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

[In the TI-SSP case]

Figure 36.43 shows an example of communication end interrupt operation during receive only slave mode on SPI operation (4-wire).

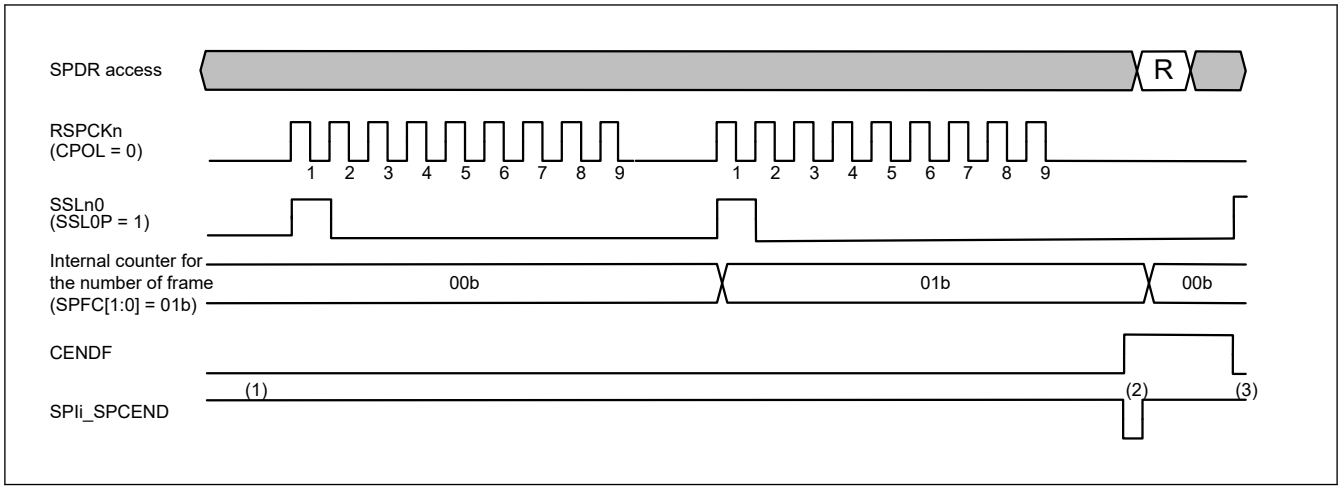


Figure 36.43 Example of Communication End Interrupt Operation (Receive-only Slave mode on SPI Operation / TI-SSP)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the RSPCK last data bit sampling when the last frame transmission ends. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared at the SSLn0 assert when the next transmission start. Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

36.3.9.5 Transmit-Receive/Transmit in Slave Mode on Clock Synchronous Operation (3-wire)

See the description of the CENDF bit in section 36.2.9. SPSR : SPI Status Register for the setting / clearing conditions of the communication completion flag during Transmit-Receive/Transmit-only in Slave Mode on Clock Synchronous (3-wire).

Figure 36.44 shows an example of communication end interrupt operation during transmit-receive/transmit slave mode on clock synchronous operation (3-wire).

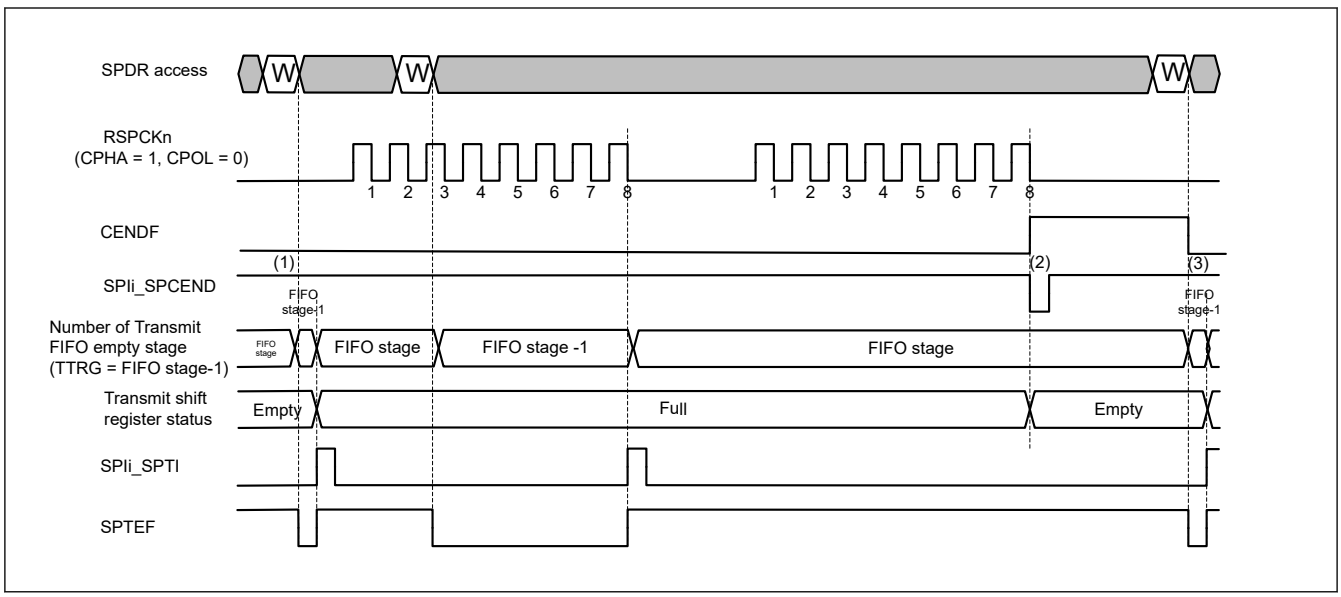


Figure 36.44 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit Slave mode on Clock Synchronous Operation)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.

2. When the next transfer data is not set in the transmit FIFO and the transmit shift register is empty, then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

36.3.9.6 Receive Only in Slave Mode on Clock Synchronous Operation (3-wire)

See the description of the CENDF bit in section 36.2.9. SPSR : SPI Status Register for the setting / clearing conditions of the communication completion flag during Receive -only in Slave Mode on Clock Synchronous (3-wire).

Figure 36.45 shows an example of communication end interrupt operation during receive only slave mode on clock synchronous operation.

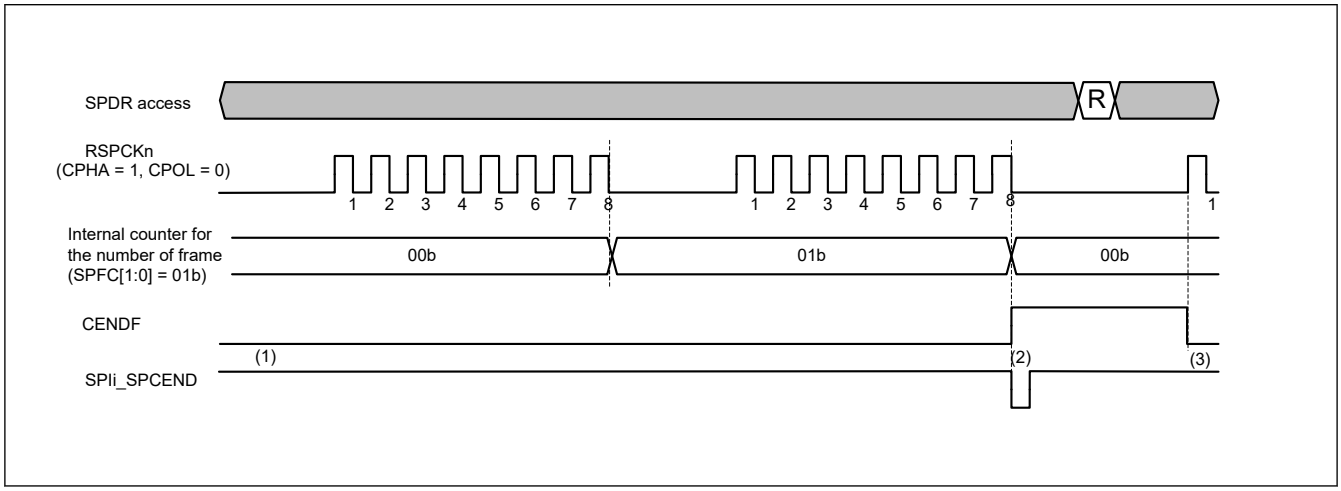


Figure 36.45 Example of Communication End Interrupt Operation (Receive-only Slave mode on Clock Synchronous Operation)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag is set to 1 (communication completed) at the timing of the last data bit sampling of RSPCKn in the last frame communication when the last frame of the SPI data control register (SPDCR) SPFC set value is received. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared at the first edge of RSPCKn for the next transmission. Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

36.3.9.7 Common Operation

In this chapter, the operation common to each mode / area option communication in section 36.3.9.1. Transmit-Receive/ Transmit in Master Mode to section 36.3.9.6. Receive Only in Slave Mode on Clock Synchronous Operation (3-wire) is explained. When the enable of SPI communication end interrupt (CENDIE) is 0, at the time of communication completion, a flag of communication end (CENDF) is set and an event of communication end is output, but no interrupt is output. However, if the enable of communication end interrupt (CENDIE) is set to 1 before clearing the flag of communication end (CENDF) while the enable of SPI function (SPE) is 1, the communication end interrupt (SPIi_SPCEND) is output.

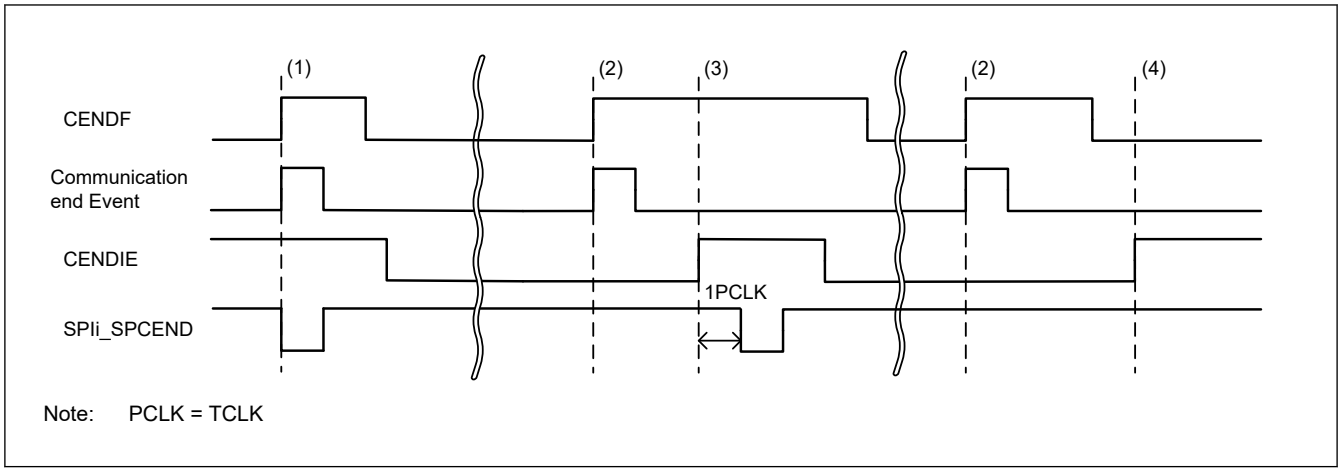


Figure 36.46 Example of Communication End Interrupt Operation (Enable control)

1. When the enable of SPI communication end interrupt (CENDIE) is 1, at the time of communication completion, the following three are the same timing.
   - A flag of communication end (CENDF)
   - An event of communication end
   - The communication end interrupt (SPIi_SPCEND)
2. When the enable of SPI communication end interrupt (CENDIE) is 0, at the time of communication completion, the following two are the same timing, but no interrupt.
   - A flag of communication end (CENDF)
   - An event of communication end
3. After (2), if the enable of communication end interrupt (CENDIE) is set when the enable of SPI function (SPE) and the flag of communication end (CENDF) are 1, the communication end interrupt (SPIi_SPCEND) is output after 1 TCLK.
4. After (2), even if the enable of communication end interrupt (CENDIE) is set when the enable of SPI function (SPE) or the flag of communication end (CENDF) is 0, the communication end interrupt (SPIi_SPCEND) is not output.

36.3.10 Error Detection

In normal SPI serial transfers, data written to the transmit buffer of SPDR is transmitted, and received data can be read from the receive buffer of SPDR. If access is made to SPDR, an abnormal transfer might occur, depending on the status of the transmit or receive buffer or the status of the SPI at the beginning or end of serial transfer.

If an abnormal transfer occurs, the SPI detects the event as an underrun error, overrun error, parity error, or mode fault error. Table 36.9 lists the relationship between non-normal transfer operations and the SPI error detection function.

Table 36.9 Relationship between non-normal transfer operations and SPI error detection (1 of 2)

Operation	Occurrence condition	SPI operation	Error detection
1	SPDR is written while no empty stages in the transmit FIFO.	• The contents of the transmit buffer are kept • Write data is missing	None
2	SPDR is read while no data stored in receive FIFO.	The contents of the receive buffer and previously received data are output.	None
3	Serial transfer is started in slave mode when the SPI is not able to transmit data.	• Serial transfer is suspended • Transmit or receive data is missing • Driving of the MISOn output signal is stopped • SPI function is disabled	Underrun error
4	Serial transfer ends when data is stored in the receive FIFO for the number of FIFO stages.	• Keeps the contents of the receive FIFO • Missing receive data	Overrun error

Table 36.9 Relationship between non-normal transfer operations and SPI error detection (2 of 2)

Operation	Occurrence condition	SPI operation	Error detection
5	An incorrect parity bit is received during full-duplex synchronous serial communication with the parity function enabled in following mode: • Transmit-receive master mode • Receive-only master mode • Transmit-receive slave mode • Receive-only slave mode	The parity error flag is asserted	Parity error
6	The SSLn0 input signal is asserted when the serial transfer is idle in multi-master mode.	• Driving of the RSPCKn, MOSIn, SSLn1 to SSLn3 output signals is stopped • SPI function is disabled	Mode fault error
7	The SSLn0 input signal is asserted during serial transfer in multi-master mode.	• Serial transfer is suspended • Transmit or receive data is missing • Driving of the RSPCKn, MOSIn, SSLn1 to SSLn3 output signals is stopped • SPI function is disabled	Mode fault error
8	[In the Motorola-SPI case] The SSLn0 input signal is negated during serial transfer in slave mode.	• Serial transfer is suspended • Transmit or receive data is missing • Driving of the MISOn output signal is stopped • SPI function is disabled	Mode fault error
9	[In the TI-SSP case] The SSLn0 input signal is asserted during serial transfer in slave mode.	• Serial transfer is suspended • Transmit or receive data is missing • Driving of the MISOn output signal is stopped • SPI function is disabled	Mode fault error
10	After data is stored in the receive FIFO with SPDRES = 1, the number of stored data is less than the threshold value and no receive data is written for the set value of SPDRC [7:0]	Assert the receive data ready flag	Receive data ready

In operation 1 described in Table 36.9, the SPI does not detect an error. To prevent data omission during writes to SPDR, the writes to SPDR must be executed using a transmit buffer empty interrupt request (when the SPSR.SPTEF flag is 1).

Similarly, the SPI does not detect an error in operation 2. To prevent extraneous data from being read, SPDR read must be executed with an SPI receive buffer full interrupt request (when the SPSR.SPRF flag is 1).

For information on the other errors, see the following sections:

- Underrun error, indicated in operation 3, see section 36.3.10.4. Underrun errors
- Overrun error, indicated in operation 4, see section 36.3.10.1. Overrun errors
- Parity error, indicated in operation 5, see section 36.3.10.2. Parity errors
- Mode fault error, indicated in operations 6 to 9, see section 36.3.10.3. Mode fault errors
- For the transmit and receive interrupts, see section 36.3.7. Transmit Buffer Empty and Receive Buffer Full Interrupts.
- For the reception data ready in operations 10, see section 36.3.10.5. Received data ready.

36.3.10.1 Overrun errors

If a serial transfer ends when the receive buffer of SPDR is full, the SPI detects an overrun error and sets the SPSR.OVRF flag to 1. When the OVRF flag is 1, the SPI does not copy data from the shift register to the receive buffer, so the data prior to the error occurrence is retained in the receive buffer. To set the OVRF flag to 0, issue a system reset or 1 is written to the SPSRC.OVRFC bit.

Figure 36.47 shows an example of operation of the OVRF and SPRF flags. The SPSRC and SPDR accesses shown in Figure 36.47 indicate the condition of accesses to the SPSRC and SPDR register, where W denotes a write cycle, and R a read cycle. In this example, the SPI performs an 8-bit serial transfer when SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

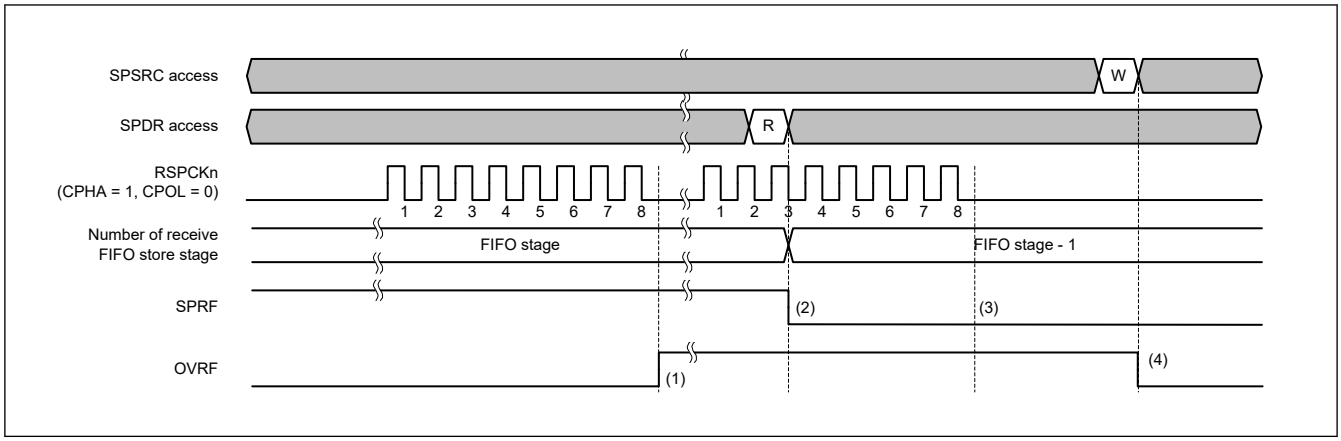


Figure 36.47 Operation example of the OVRF and SPRF flags

The operation of the flags at timings (1) to (4) in Figure 36.47 is as follows:

1. When serial transfer ends while data is stored for the number of FIFO stages, the SPI detects an overrun error and sets the OVRF flag to 1. The SPI does not copy shift register data to the receive buffer. The SPI does not detect a parity error even when SPPE = 1. In master mode, the SPI copies the value of pointer to the SPI command register (SPCMDm) to the SPECM[2:0] bits in the SPI status register (SPSR).
2. When SPDR is read, the SPI outputs the data in the receive buffer. At this time, the SPRF flag is cleared to 0 at the last access when the received data is read from SPDR in one processing routine using DTC / DMAC.
3. If the serial transfer ends with the OVRF flag set to 1 (overrun error occurred), the SPI does not copy data in the shift register to the receive buffer (the SPRF flag does not set to 1). A receive buffer full interrupt is not generated. Even when the SPPE bit is 1, parity errors are not detected. In an overrun error state when the SPI does not copy the received data from the shift register to the receive buffer, on termination of the serial transfer, the SPI determines that the shift register is empty. This enables data transfer from the transmit buffer to the shift register.
4. When 1 is written to the SPSRC.OVRFC bit, the SPSR.OVRF flag is cleared.

The occurrence of an overrun can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. When executing a serial transfer, you must ensure that overrun errors are detected early, for example by reading SPSR immediately after SPDR is read.

If an overrun error occurs and the OVRF flag sets to 1, normal reception operations cannot be performed until the OVRF flag is set to 0.

When the RSPCK auto-stop function is enabled (SPCR.SCKASE = 1) in master mode, an overrun error does not occur. Figure 36.48 and Figure 36.49 show the clock stop waveform when a serial transfer continues while the receive buffer is full in master mode.

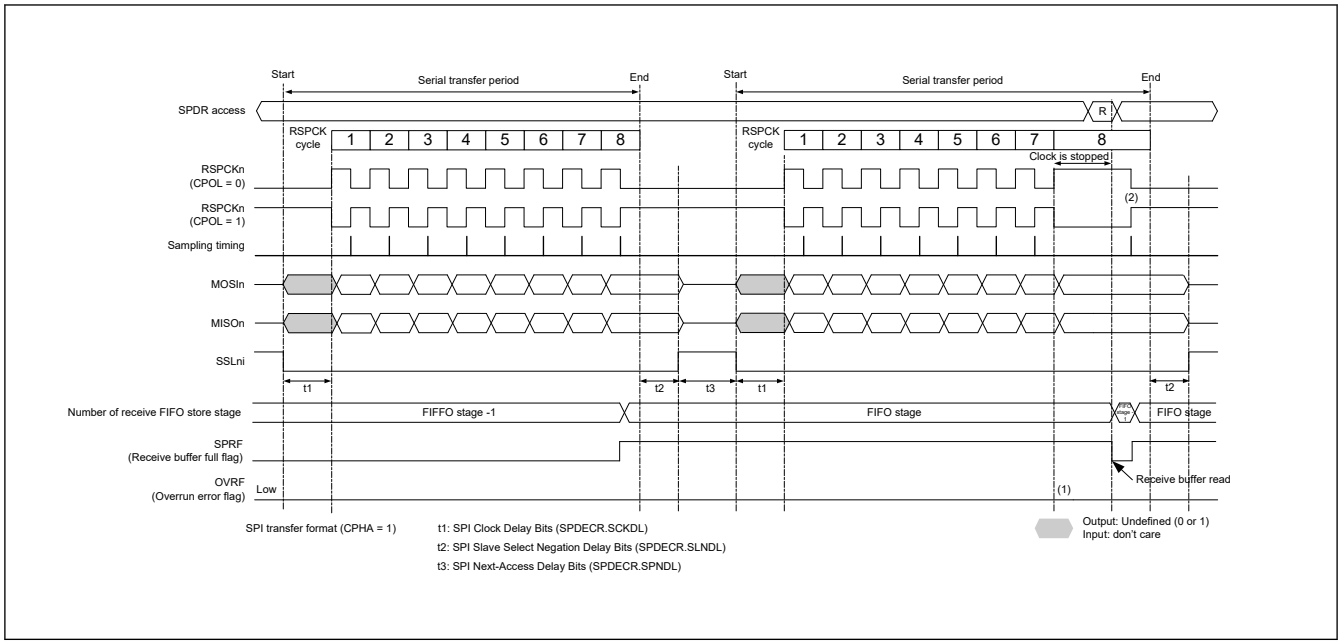


Figure 36.48 Clock stop waveform when serial transfer continues with data is stored for the number of FIFO stages in master mode (CPHA = 1)

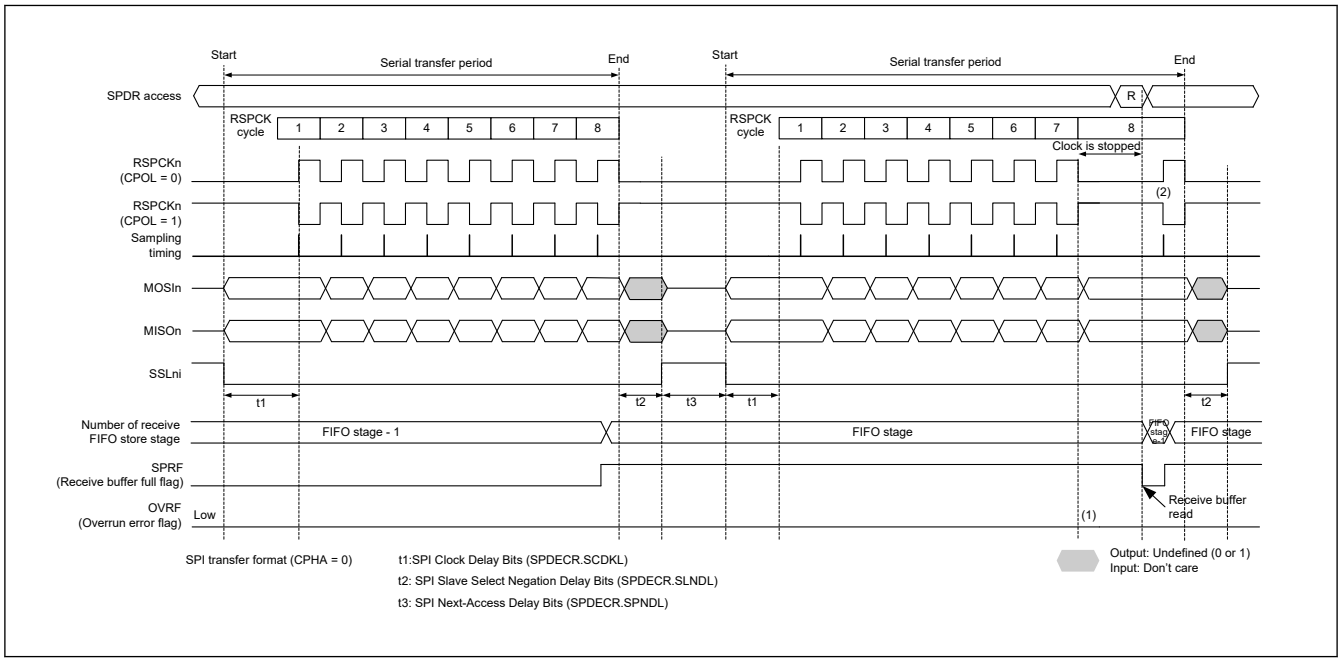


Figure 36.49 Clock stop waveform when serial transfer continues with data is stored for the number of FIFO stages in master mode (CPHA = 0)

The operation of the flags at timings (1) and (2) in Figure 36.48 and Figure 36.49 is as follows:

1. While data is stored in the receive FIFO for the number of FIFO stages, the RSPCK clock is deactivated and no overrun error occurs.
2. If SPDR is read while the clock is stopped, data in the receive buffer can be read. The RSPCK clock restarts.

Overrun error does not occur when RSPCK automatic stop function is enabled for transfer with no delay of between frames during burst transfer in master mode. Figure 36.50 and Figure 36.51 show the clock stop waveform, when there is no delay between frames at burst transfer and the serial transfer continues in the data is stored in the receive FIFO for the number of FIFO stages.

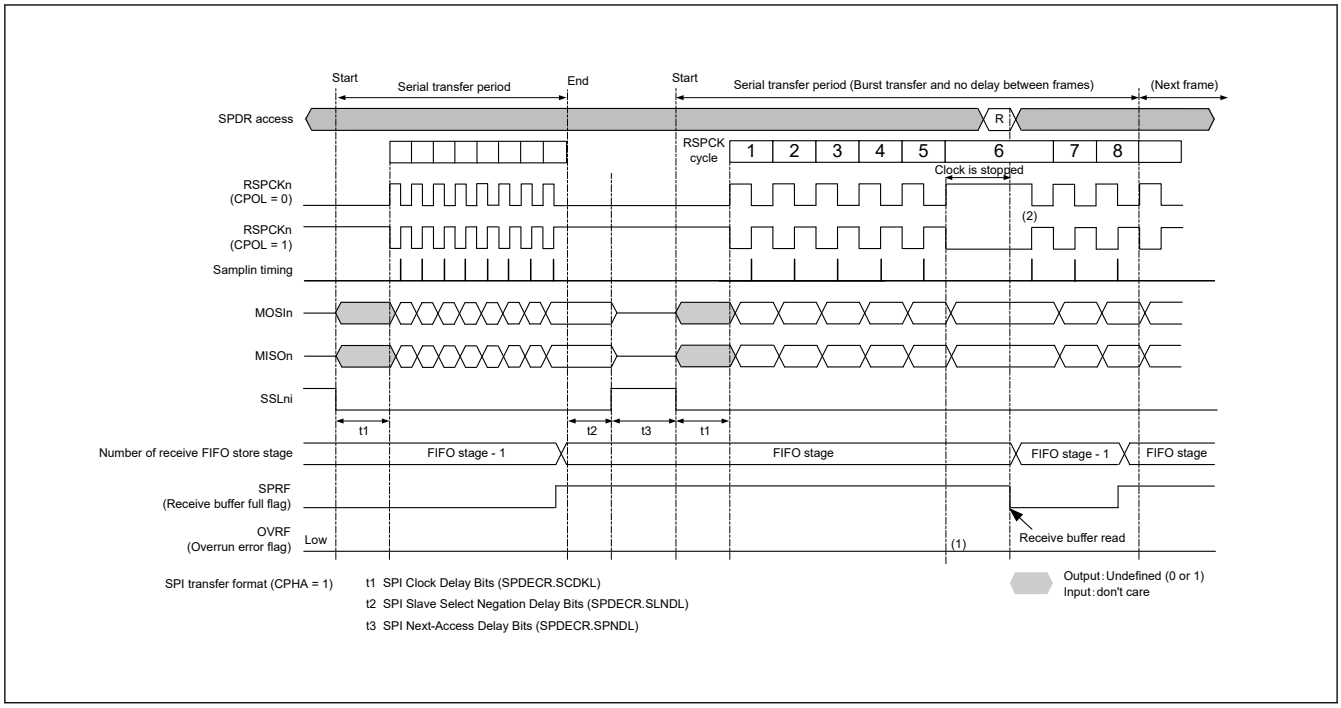


Figure 36.50 Clock Stop Waveform when Serial Transfer Continues in the Receive Buffer Full with data is stored for the number of FIFO stages (at burst transfer and no delay between frames CPHA = 1)

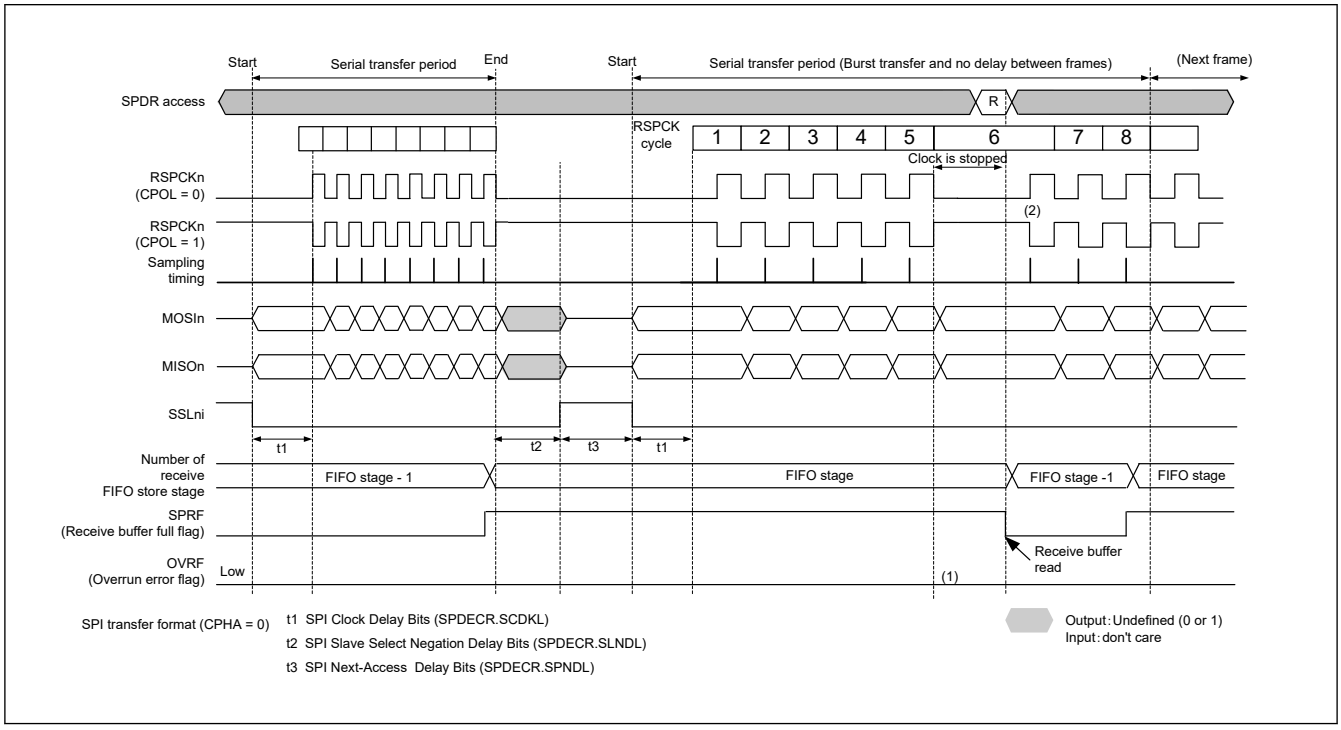


Figure 36.51 Clock Stop Waveform when Serial Transfer Continues with data is stored for the number of FIFO stages in Master Mode (at burst transfer and no delay between frames CPHA = 0)

The following describes operation of flags at timings (1) and (2) in the figure above.

1. While the data is stored for the number of FIFO stages, the RSPCK clock is deactivated and no overrun error occurs.
2. Receive buffer data can be read by reading SPDR during clock stop. After the receive buffer data has been read the RSPCK clock restarts.

36.3.10.2 Parity errors

After transfer in transmit-receive or receive-only master mode, transmit-receive slave mode or receive only slave mode while the SPPE bit in the SPI control register (SPCR) is 1, the SPI checks occurrence of a parity error. When the SPI detects a parity error in received data, the PERF flag in the SPI status register (SPSR) is set to 1. While the OVRF flag is 1, the SPI does not copy shift register data to the receive buffer. Therefore, parity error in received data is not detected. To clear the PERF flag in SPSR to 0, issue a system reset or 1 is written to the SPSRC.PERFC bit.

Figure 36.52 shows an example of operation of the OVRF and PERF flags. The SPSR access shown in Figure 36.52 indicates the condition of access to the register, where W denotes a write cycle, and R a read cycle. In this example, full-duplex serial communication is performed while the SPCR.SPPE bit is 1. The SPI performs an 8-bit serial transfer when SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

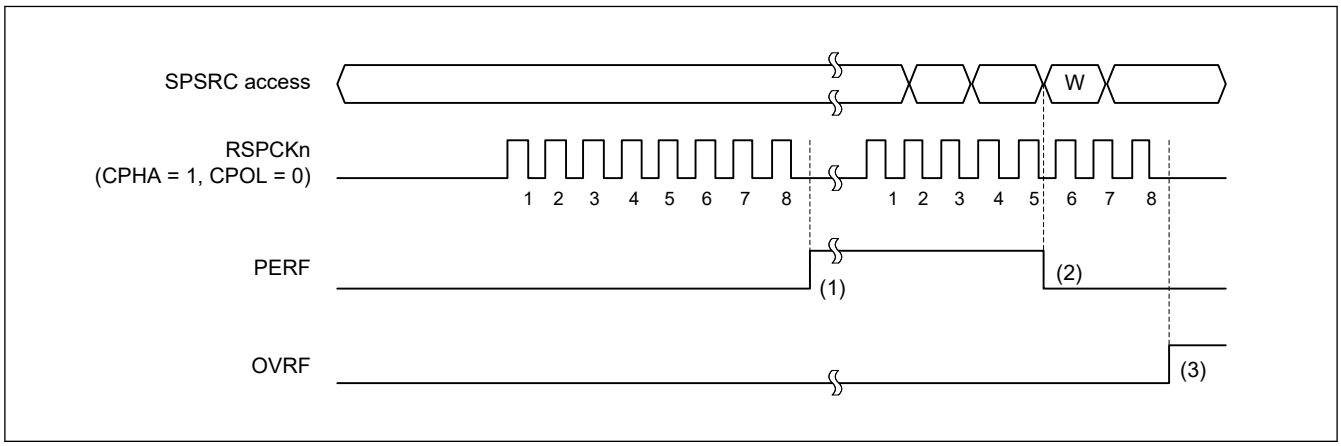


Figure 36.52 Operation example of the OVRF and PERF flags

The operation of the flags at timings (1) to (3) in Figure 36.52 is as follows:

1. When the SPI does not detect an overrun error and terminates the serial transfer, the SPI copies shift register data to the receive buffer. When the SPI checks the received data and detects a parity error at this time, the PERF flag is set to 1. In master mode, the SPI copies the value of pointer to the SPI command register (SPCMDm) to the SPECM[2:0] bits in the SPI Data control register 2 (SPDCR2).
2. When 1 is written to the SPSRC.PERFC bit, then clear the PERF flag.
3. When the SPI detects an overrun error and serial transfer is terminated, the data in the shift register is not copied to the receive buffer. The SPI does not perform parity error detection at this time.

Parity errors can be checked for by either reading the SPSR register or using an SPI error interrupt and reading the SPSR register. When executing a serial transfer, such checks are required to ensure early detection of parity errors. When the SPI is used in master mode, the pointer value to the SPCMDm register at the occurrence of the error can be checked by reading the SPDCR2.SPECM[2:0] bits.

36.3.10.3 Mode fault errors

The SPI operates in multi-master mode when the SPCR.MSTR bit is 1, the SPCR.SPMS bit is 0, and the SPCR.MODFEN bit is 1.

If the active level is input for the SSLn0 input signal of the SPI in multi-master mode, the SPI detects a mode fault error regardless of the status of the serial transfer, and sets the SPSR.MODF flag to 1.

On detecting the mode fault error, the SPI copies the value of the pointer to SPCMD to the SPECM[2:0] bits.

The active level of the SSLn0 signal is determined by the SPCR3.SSL0P bit.

When the MSTR bit is 0, the SPI operates in slave mode.

When the SPCR.MODFEN bit = 1 and the SPMS bit = 0 in slave mode, if the SSLn0 input signal is negated during the serial transfer period (from valid data drive start to final valid data latch), the SPI detects a mode fault error, while any of the following 2 conditions is met.

[In the Motorola-SPI case]

When the SSLn0 input signal is negated while serial data transfer.

[In the TI-SSP case]

When the SSLn0 input signal is asserted while serial data transfer. However, during burst transfer, no error is detected even if the SSLn0 input signal is asserted during the last bit of frame.

When the SPI detects a mode fault error, it stops driving output signals and clears the SPE bit in the SPCR register. When the SPE bit is cleared, the SPI function is disabled (as described in section 36.3.12. SPI Operation). In a multi-master configuration, the mastership can be released by stopping driving output signals and disabling the SPI function by using a mode fault error.

Whether a mode fault error is present can be checked by reading SPSR or by reading an SPI error interrupt and SPSR. To detect a mode fault error without using an SPI error interrupt, poll SPSR. When the SPI is used in master mode, the pointer value to SPCMD when an error is present can be checked by reading the SPECM[2:0] bits in SPSR.

While the MODF flag = 1, the SPI ignores writing 1 to the SPE bit. To enable the SPI function after a mode fault error is detected, clear the MODF flag to 0 without fail.

36.3.10.4 Underrun errors

While the SPI is operating in slave mode (SPCR.MSTR bit = 0) and the communication mode select bit (TXMD[1:0]) in the SPI control register (SPCR) is set to 00b or 01b, if serial transfer is started before transmit data output is ready with the SPCR.SPE bit set to 1 (SPI function enabled), the SPI detects an underrun error and sets the SPSR.MODF and SPSR.UDRF flags to 1.

On detecting an underrun error, the SPI stops the driving of output signals and clears the SPCR.SPE bit to 0 (see section 36.3.11. Initializing the SPI).

The occurrence of underrun errors can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. Detecting underrun errors without using the SPI error interrupt requires polling of SPSR.

When the MODF flag is 1, writing 1 to the SPE bit is ignored by the SPI. To enable the SPI function after the detection of an underrun error, the MODF flag must be set to 0.

36.3.10.5 Received data ready

When SPCR.TXMD[1:0] = 00b, 01b, or 11b, and SPCR2.SPDRC[7:0] ≠ 0x00, after receiving data in the receive FIFO during communication (SPE = 1), SPSR.SPDRF flag is set to 1 when the received data is not stored even after the number of received FIFOs is equal to or less than ≤ the threshold value and the value set in SPDRC[7:0] has elapsed.

When the receive data ready is detected, the interrupt and event link output can be selected as SPIi_SPRI or SPIi_SPEI with the SPCR.SPDRES bit.

Figure 36.53 shows an example of reception data ready detection operation.

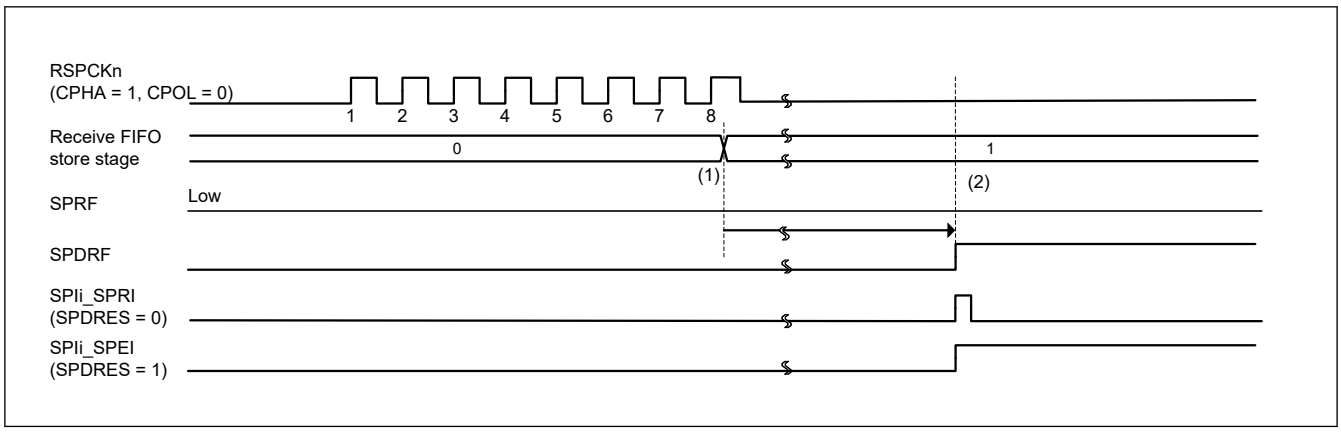


Figure 36.53 Received data ready

The following describes the operation at the timings indicated by (1) and (2) in the figure.

(1) Store the received data in the receive FIFO. SPRF is 0, because receive FIFO store stage ≤ number of frames set by SPDCR2.RTRG[1:0] .

(2) Set SPDRF and assert SPIi_SPRI or SPIi_SPEI because there is no writing to the receive FIFO for the amount of SPDRFC [7:0] set from above (1).

36.3.11 Initializing the SPI

If 0 is written to the SPCR.SPE bit or if the SPI sets the SPE bit to 0 because it detected a mode fault error or an underrun error, the SPI disables the SPI function and initializes some of the module functions. When a system reset is generated, the SPI initializes all of the module functions. This section describes initialization by clearing of the SPCR.SPE bit and by a system reset.

36.3.11.1 Initialization by clearing of the SPCR.SPE bit

When the SPCR.SPE bit is set to 0, the SPI initializes by:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals (Hi-Z) in slave mode
- Initializing the internal state of the SPI
- Initializing the transmit buffer of the SPI (the SPSR.SPTEF flag sets to 1)

Initialization by clearing of the SPE bit does not initialize the control bits of the SPI. For this reason, the SPI can be started in the same transfer mode in use prior to initialization when the SPE bit is set to 1 again.

The SPSR.CENDF, SPSR.SPRF, SPSR.OVRF, SPSR.MODF, SPSR.PERF, and SPSR.UDRF flags are not initialized, and the value of the SPDCR2.SPECM[2:0] and SPDCR2.SPCP[2:0] bits are not initialized. Therefore, even after the SPI is initialized, data from the receive buffer can be read to check the communication completion status and the error status during an SPI transfer.

The transmit buffer is initialized to an empty state (the SPSR.SPTEF flag sets to 1). Therefore, if the SPCR.SPTIE bit is set to 1 after SPI initialization, a transmit buffer empty interrupt is generated. To disable any transmit buffer empty interrupts when the SPI is initialized, write 0 to the SPTIE bit simultaneously while writing 0 to the SPE bit.

36.3.11.2 Initialization by system reset

A system reset completely initializes the SPI by initializing all SPI control bits, status bits, and data registers, in addition to the requirements described in section 36.3.11.1. Initialization by clearing of the SPCR.SPE bit.

36.3.12 SPI Operation

36.3.12.1 Master mode operation

The only difference between single- and multi-master mode operation is the use of mode fault error detection (see section 36.3.10. Error Detection). In single-master mode, the SPI does not detect mode fault errors whereas in multi-master mode, it does. This section explains operations that are common to both modes.

(1) Starting a serial transfer

When data is written to the SPI data register (SPDR) while the next transfer data is not set in the transmit FIFO, the SPI updates the transmit buffer (SPTXn, n = 0 to 3) data in SPDR. When the shift register is empty, the SPI copies data from the transmit buffer to the shift register and starts serial transfer. On copying transmit data to the shift register, the SPI changes the status of the shift register to full. On termination of the serial transfer, it changes the status of the shift register to empty. The status of the shift register cannot be referenced.

The polarity of the SSLni output pins depends on the SPCR3.SSLnP (n = 0 to 3) bits settings. For details on the SPI transfer format, see section 36.3.5. Transfer Formats.

(2) Terminating a serial transfer

[Except Receive-only in Master Mode]

After the SPI detects the RSPCKn edge corresponding to the final sampling timing regardless of the CPHA bit value in the SPI command register (SPCMD), the SPI terminates serial transfer. When the number of data stored in the receive FIFO is less than the number of FIFO stages, data is copied from the shift register to the receive buffer in the SPI data register (SPDR) after serial transfer.

The final sampling timing varies depending on the bit length of transfer data. In master mode, the SPI data length depends on the SPCMDm.SPB[4:0] bit settings. The polarity of the SSLni output pin depends on the SPCR3.SSLnP (n = 0 to 3) bits settings. For details on the SPI transfer format, see section 36.3.5. Transfer Formats.

[Receive-only in Master Mode]

When any of the following 2 conditions is met, then SPI terminating the serial transfer.

- After the SPI detects the RSPCKn edge corresponding to the final sampling timing regardless of the CPHA bit value in the SPI command register (SPCMD), the SPI terminates serial transfer.
- When writing SPCR2.RMEDTG = 1 during the serial transfer period, SPI terminating the serial transfer.

When the number of data stored in the receive FIFO is less than the number of FIFO stages, data is copied from the shift register to the receive buffer in the SPI data register (SPDR) after serial transfer.

The final sampling timing varies depending on the transfer data bit length. The data length of the SPI in master mode depends on the set value of the SPB[4:0] bits in the SPI command register (SPCMD). The SSLni output signal polarity depends on the set SPI SSLi signal polarity bit (SPCR3.SSLiP) (i = 0 to 3) value. For details about the SPI transfer format, see section 36.3.5. Transfer Formats.

(3) Sequence control

The transfer format in master mode is determined as follows.

The transfer format used in master mode is determined by the SPCR3, SPCMDm, and SPDECR registers.

The SPCR3.SPSLN[2:0] bits determine the sequence configuration for serial transfers that are executed by the SPI in master mode. The following items are set in the SPCMDm register:

- SSLni pin output signal value
- MSB- or LSB-first
- Data length
- Some of the bit rate settings
- RSPCKn polarity and phase
- Whether SPDECR.SCKDL is to be referenced
- Whether SPDECR.SLNDL is to be referenced
- Whether SPDECR.SPNDL is to be referenced

SPCR3.SPBR holds some of the bit rate settings, including SPDECR.SCKDL (SPI clock delay), SPDECR.SLNDL (SSL negation delay), and SPDECR.SPNDL (next-access delay).

Based on the sequence length assigned in SPCR3.SPSLN, the SPI makes up a sequence comprised of a part or all of the SPCMDm register. The SPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPDCR2.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the SPI function is enabled, the SPI loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. The SPI increments the pointer each time the next-access delay period for a data transfer ends. On completion of the serial transfer that corresponds to the final command in the sequence, the SPI sets the pointer to SPCMD0, and in this way the sequence is executed repeatedly.

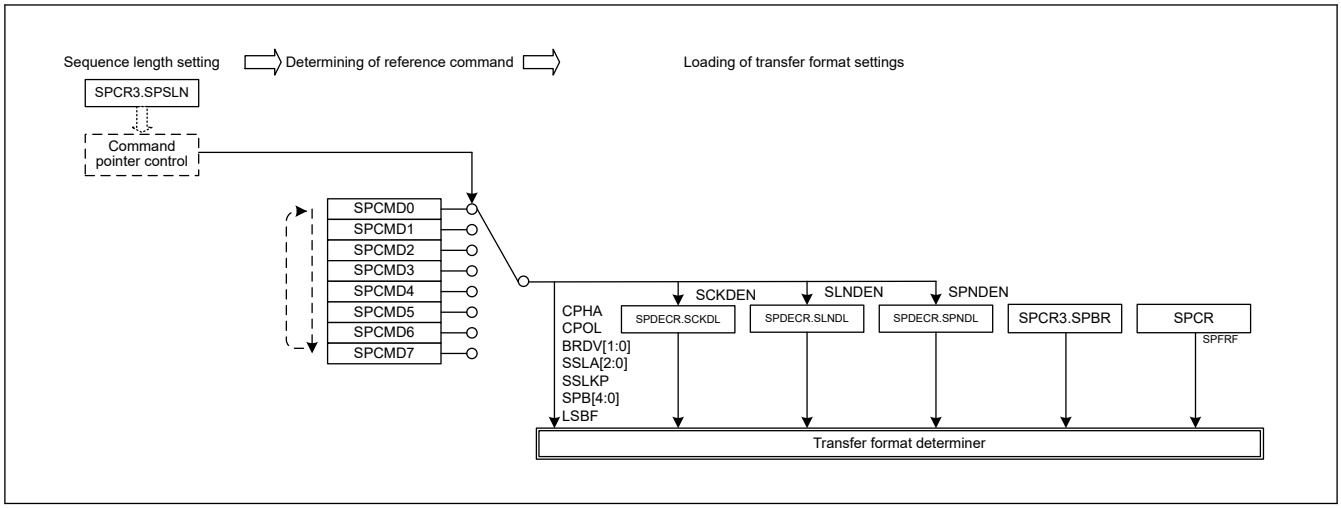


Figure 36.54 Procedure for determining the form of a serial transfer in master mode

In this section, a frame is the combination of the data in SPDR and the settings in SPCMDm.

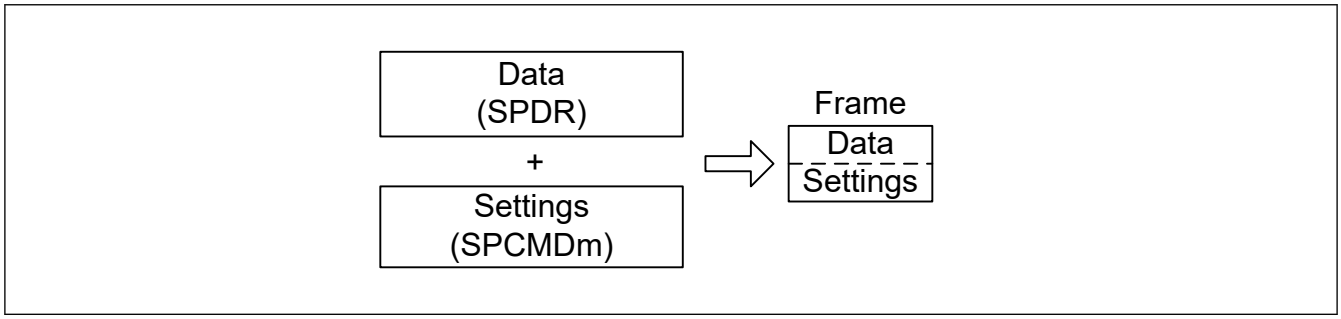


Figure 36.55 Conceptual diagram of frames

Figure 36.56 shows the correspondence between the commands and the transmit and receive buffers in the sequence of operations specified by the settings.

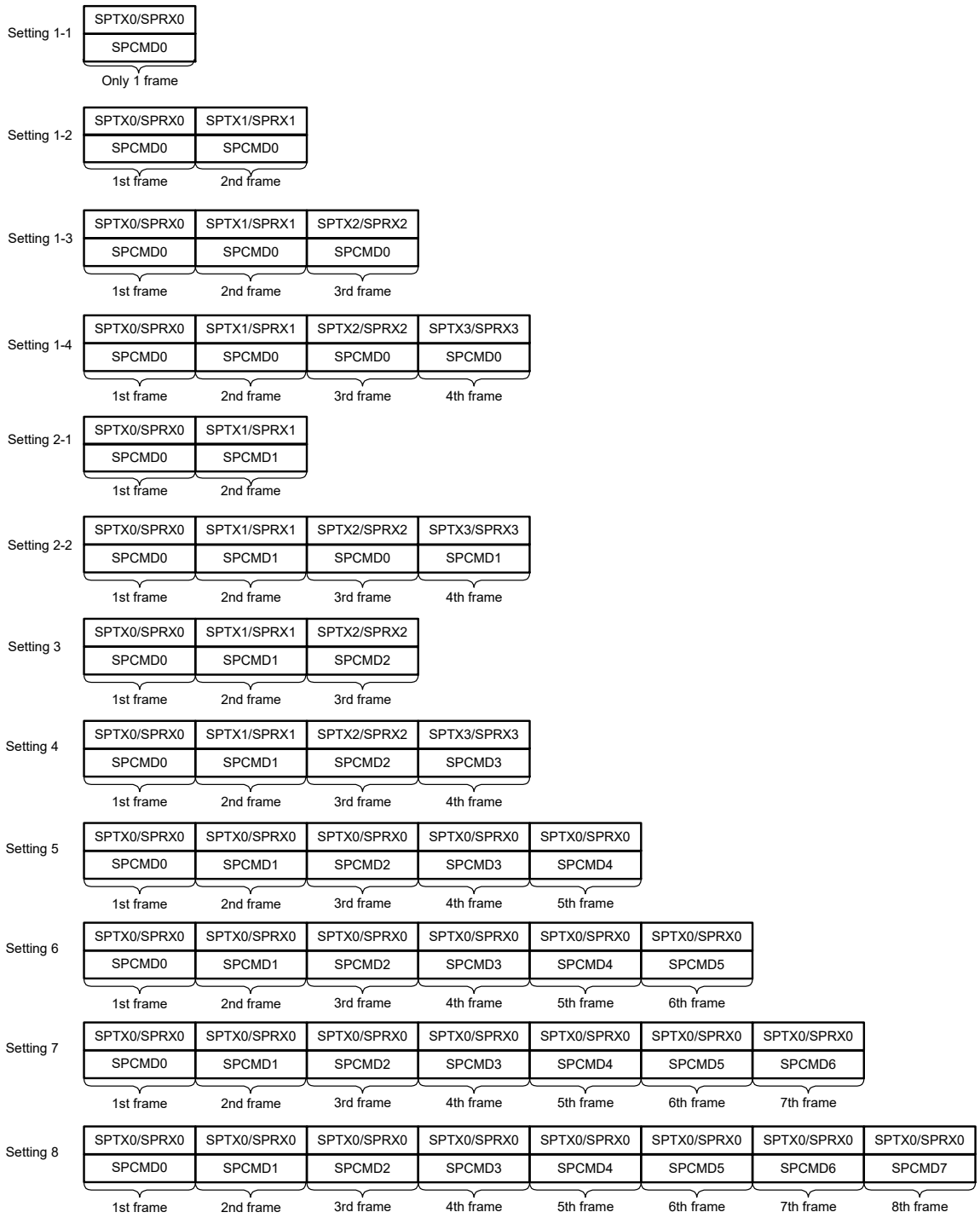


Figure 36.56 Correspondence between SPI Command Register and transmit and receive buffers in sequence operations

(4) Burst transfers

This section describes burst transfer during transmit-receive/transmit-only operation.

[In the Motorola-SPI case]

If the SPCMDm.SSLKP bit that the SPI references during the current serial transfer is 1, the SPI maintains the SSLni signal level during the serial transfer until the beginning of the SSLni signal assertion for the next serial transfer. If the SSLni signal level for the next serial transfer is the same as the SSLni signal level for the current serial transfer, the SPI can execute continuous serial transfers while keeping the SSLni signal assertion status (burst transfer).

- When Between Burst Transfer Frames Delay Select bit (BFDS) of SPI control register (SPCR) is 0.

Figure 36.57 shows an example of an SSLni signal operation for a burst transfer that is implemented using the SPCMD0 and SPCMD1 register settings. This section describes SPI operations (1) to (8) shown in Figure 36.57.

Note: The polarity of the SSLni output signal depends on the SPCR3.SSLnP (n = 0 to 3) bits settings.

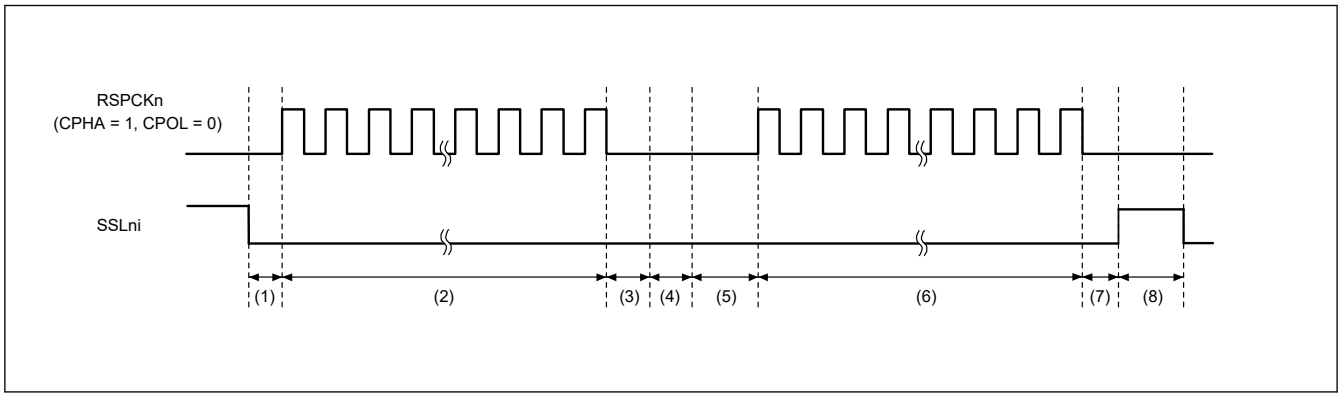


Figure 36.57 Example of burst transfer operation using the SSLKP bit (BFDS = 0, SPFRF = 0)

The SPI operation at times (1) to (8) in the figure is as follows:

1. Based on the SPCMD0 settings, the SPI asserts the SSLni signal and inserts RSPCK delays.
2. The SPI executes serial transfers in accordance with the SPCMD0 settings.
3. The SPI inserts an SSL negation delay.
4. Because the SPCMD0.SSLKP bit is 1, the SPI keeps the SSLni signal value specified in SPCMD0. This period additionally continues for 5 TCLK cycles (at minimum) that is the same as the next-access delay time of SPCMD0. If the shift register is empty after the passage of the minimum period, this period is sustained until the transmit data is stored in the shift register for the next transfer.
5. Based on the SPCMD1 settings, the SPI asserts the SSLni signal and inserts RSPCK delays.
6. The SPI executes serial transfers in accordance with the SPCMD1 settings.
7. Insert SSL negate delay.
8. Because the SPCMD1.SSLKP bit is 0, the SPI negates the SSLni signal. In addition, a next-access delay is inserted in accordance with SPCMD1.

If the SSLni signal output settings in the SPCMDm register where 1 is assigned to the SSLKP bit are different from the SSLni signal output settings in the SPCMDm register to be used in the next transfer, the SPI switches the SSLni signal status to SSLni signal assertion as shown in (5) in Figure 36.57. This corresponds to the command for the next transfer.

Note: If such an SSLni signal switching occurs, the slaves that drive the MISOn signal compete, and collision of signal levels might occur.

The SPI in master mode references the SSLni signal operation within the module when the SSLKP bit is not used. When the SPCMDm.CPHA bit is 0, the SPI can accurately start serial transfers by using the SSLni signal assertion for the next transfer that is detected internally.

- When Between Burst Transfer Frames Delay Select bit (BFDS) of SPI control register (SPCR) is 1.

Figure 36.58 shows an example of SSLni signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (6) shown in Figure 36.58. The SSLni output signal polarity depends on the set SPCR3.SSLnP (n = 0 to 3) value.

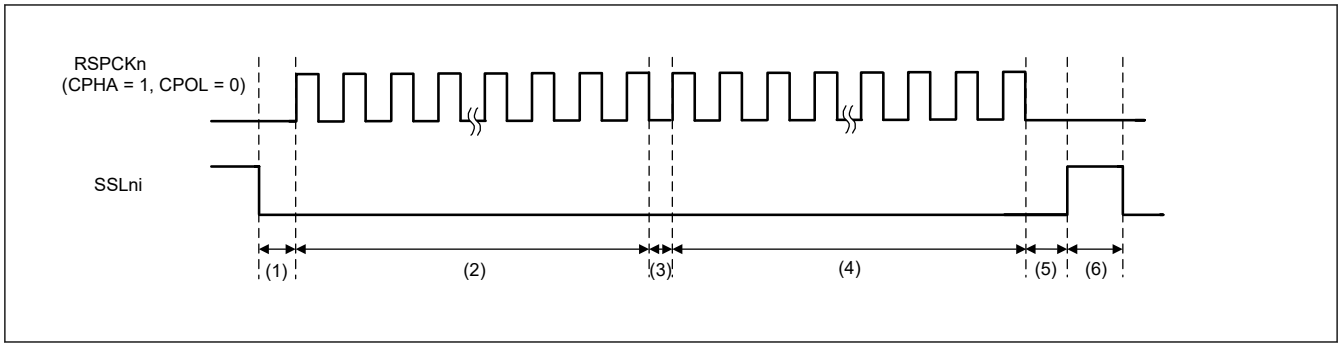


Figure 36.58 Example of Burst Transfer Operation Using SSLKP Bit (BFDS = 1, SPFRF = 0)

1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0. The RSPCK delay is inserted only the first frame of burst transmission.
2. Perform serial transfer according to SPCMD0. Wait last clock until the next transmit data is stored in the shift register, if the shift register is empty during RSPCK negate period between frames.
3. The value of SSLni signal according to SPCMD0 was hold, because the SPCMD0.SSLKP bit is 1. RSPCK negate period between frames is 0.5RSPCK, if the shift register is not empty.
4. Perform serial transfer according to SPCMD1.
5. Insert SSLni negate delay for the last frame.
6. The SSLni signal is negated because the SSLKP bit in SPCMD1 is 0. Furthermore, the next-access delay is inserted according to SPCMD1.

[In the TI-SSP case]

SPI asserts the SSLni signal for one cycle at the start of serial transfer.

Serial transfer can be executed continuously by asserting the SSLni signal for one cycle at the start of the next serial transfer (burst transfer).

- When the SSLni signal level holding bit (SSLKP) of the SPI command register (SPCMD) is 1 and the burst transfer frame delay selection bit (BFDS) of the SPI control register (SPCR) is 1, SPCMD0 to SPCMD1 are shown in Figure 36.59. The following shows an example of SSLni signal operation and serial data MISOn/MOSIn when burst transfer is realized using the settings. The SSLni output signal polarity depends on the set SPI SSLi signal polarity bit (SPCR3.SSLiP) (i = 0 to 3) value.

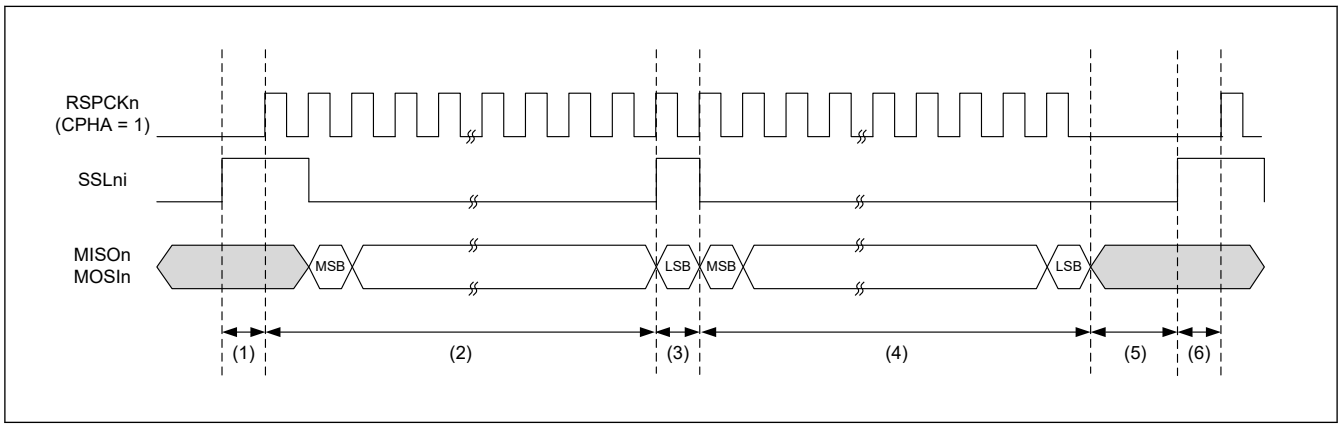


Figure 36.59 Example of Burst Transfer Operation (SPFRF = 1)

1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0.
   The RSPCK delay is inserted only the first frame of burst transmission.
2. Perform serial transfer according to SPCMD0.
3. Final data transfer and SSLni assertion are performed simultaneously. If the shift register is empty during the RSPCK negation period between frames, wait for the output of the last clock until the transmission data for the next transfer is stored in the shift register.

4. Perform serial transfer according to SPCMD1.
5. Insert OE negate delay for the last frame.
6. Insert the next access delay according to SPCMD1.

If the SSLni signal output setting in SPCMD with the SSLKP bit set to 1 differs from the SSLni signal output setting in SPCMD to be used for the next transfer, the SPI changes the SSLni signal state when the SSLni signal corresponding to the next-transfer command is asserted ((5)). Note that, if an SSLni signal change like this takes place, slaves that drive the MISOn signal may conflict with each other, which may cause collision of signal level.

This section describes burst transfer during receive-only operation.

[In the Motorola-SPI case]

When the SSLKP bit in the SPI command register (SPCMD), which the SPI references in the current serial transfer, is 1, the SPI retains the SSLni signal level during serial transfer until the SSLni signal assertion of the next serial transfer starts. When the SSLni signal level in the next serial transfer is the same as the SSLni signal level in the current serial transfer, the SPI can continuously perform serial transfer while holding the SSLni signal assertion status (burst transfer).

- When Between Burst Transfer Frames Delay Select bit (BFDS) of SPI control register (SPCR) is 0.

Figure 36.60 shows an example of SSLni signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (8) shown in Figure 36.60. The SSLni output signal polarity depends on the set SPI SSLni signal polarity bit (SPCR3.SSLiP) (i = 0 to 3) value.

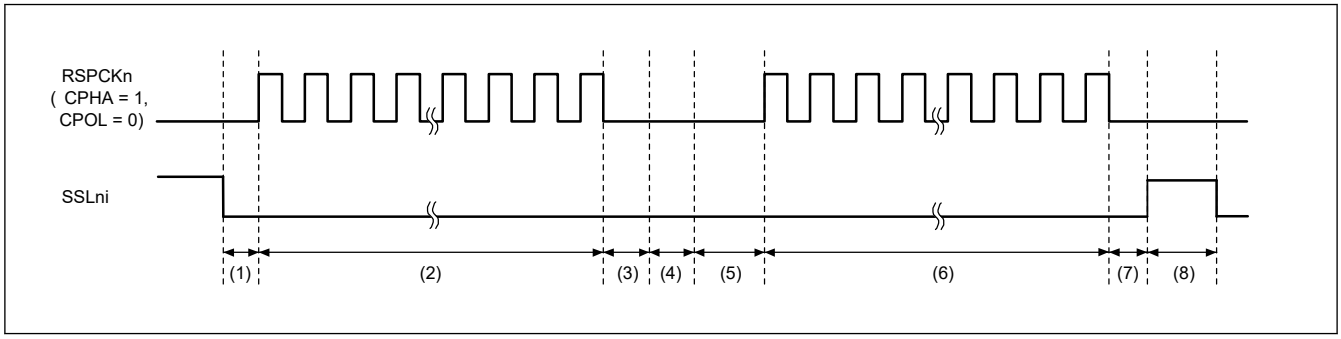


Figure 36.60 Example of Burst Transfer Operation Using SSLKP Bit (BFDS = 0, SPFRF = 0)

1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0.
2. Perform serial transfer according to SPCMD0.
3. Insert an SSLni negation delay.
4. The SSLni signal value in SPCMD0 is retained because the SSLKP bit in SPCMD0 is 1. This period additionally continues for 5 TCLK cycles (at minimum) that is the same as the next-access delay time of SPCMD0.
5. Assert the SSLni signal and insert an RSPCK delay according to SPCMD1.
6. Perform serial transfer according to SPCMD1.
7. Insert SSLni negate delay.
8. The SSLni signal is negated because the SSLKP bit in SPCMD1 is 0. Furthermore, the next-access delay is inserted according to SPCMD1.

If the SSLni signal output setting and the SSLni signal output setting between SPCMDs used for burst transfer are different, SPI switches the SSLni signal state when the SSLni signal corresponding to the next transfer command is asserted (5). Note that, if an SSLni signal change like this takes place, slaves that drive the MISOn signal may conflict with each other, which may cause collision of signal level.

The SPI in master mode references the SSLni signal operation in the module when SSLKP is not used.

Even when the CPHA bit in SPCMD is 0, the SPI can accurately start serial transfer by using the next transfer SSLni signal assertion detected internally. For this reason, burst transfer in master mode is enabled regardless of the set CPHA bit value. (See section 36.3.11. Initializing the SPI.)

- When Between Burst Transfer Frames Delay Select bit (BFDS) of SPI control register (SPCR) is 1.

Figure 36.61 shows an example of SSLni signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (6) shown in Figure 36.61. The SSLni output signal polarity depends on the set SPI SSLi signal polarity bit (SPCR3.SSLiP) (i = 0 to 3) value.

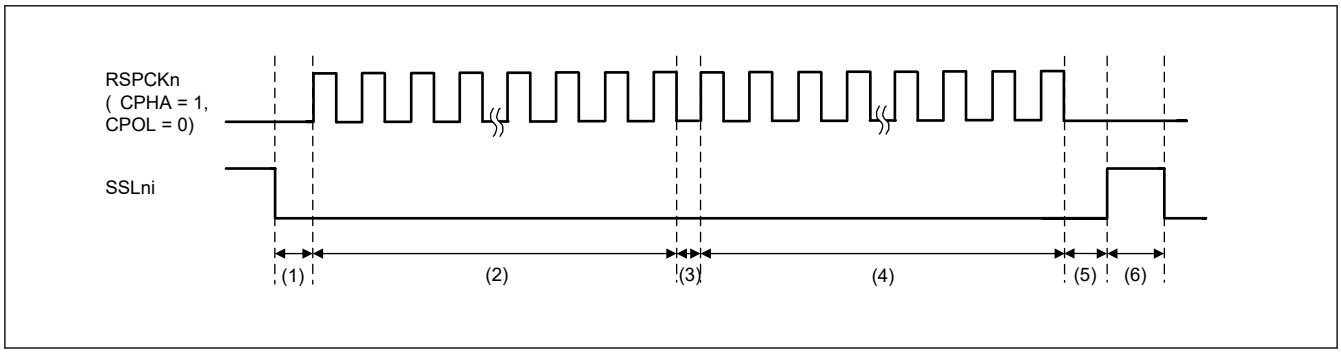


Figure 36.61 Example of Burst Transfer Operation Using SSLKP Bit (BFDS = 1, SPFRF = 0)

1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0.
   The RSPCK delay is inserted only the first frame of burst transmission.
2. Perform serial transfer according to SPCMD0.
3. Since it is not the last frame, the SSLni signal value at SPCMD0 is retained. RSPCKn negation between frames is 0.5 RSPCKn for the next frame.
4. Perform serial transfer according to SPCMD1.
5. Insert SSLni negate delay for the last frame.
6. The SSLni signal is negated. Furthermore, the next-access delay is inserted according to SPCMD1.

Note: Last frame: Frame set by RMFM[4:0] bits when SPCR2.RMFM[4:0] ≠ 0x00
Or, a frame in which SPCR2.RMEDTG = 1 has been accepted.

[In the TI-SSP case]

SPI asserts the SSLni signal for one cycle at the start of serial transfer.

Serial transfer can be executed continuously by asserting the SSLni signal for one cycle at the start of the next serial transfer (burst transfer).

- When the SSLni signal level holding bit (SSLKP) of the SPI command register (SPCMD) is 1 and the burst transfer frame delay selection bit (BFDS) of the SPI control register (SPCR) is 1, SPCMD0 to SPCMD1 are shown in Figure 36.62. The following shows an example of SSLni signal operation and serial data MISOn / MOSIn when burst transfer is realized using the settings. The SSLni output signal polarity depends on the set SPI SSLi signal polarity bit (SPCR3.SSLiP) (i = 0 to 3) value.

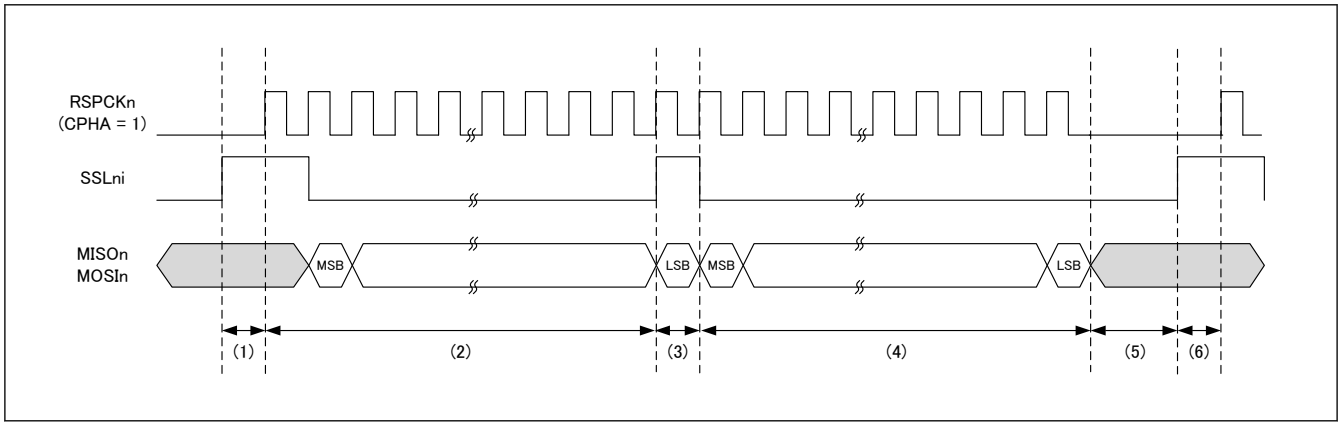


Figure 36.62 Example of Burst Transfer Operation (SPFRF = 1)

1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0.
   The RSPCK delay is inserted only the first frame of burst transmission.

2. Perform serial transfer according to SPCMD0.
3. Final data transfer and SSLni assertion are performed simultaneously.
4. Perform serial transfer according to SPCMD1.
5. Insert OE negate delay for the last frame.
6. Insert the next access delay according to SPCMD1.

Note: Last frame: Frame set by RMFM[4:0] bits when SPCR2.RMFM[4:0] ≠ 0x00
Or, a frame in which SPCR2.RMEDTG = 1 has been accepted.

If the SSLni signal output setting between the SPCMDs used for burst transfer differs from the SSLni signal output setting, SPI switches the SSLni signal state when the SSLni signal corresponding to the next transfer command is asserted (5). Note that, if an SSLni signal change like this takes place, slaves that drive the MISOn signal may conflict with each other, which may cause collision of signal level.

(5) RSPCK delay (t1)

The RSPCK delay value of the SPI in master mode depends on the SPCMDm.SCKDEN bit setting and the SPDECR.SCKDL[2:0] bits setting. The SPI determines the SPCMDm register to be referenced during a serial transfer by pointer control, and determines an RSPCK delay using the SPCMDm.SCKDEN bit and SPDECR.SCKDL[2:0] bits, as listed in Table 36.10. For a definition of RSPCK delay, see section 36.3.5. Transfer Formats.

RSPCK delay insert to only the first frame of burst transmission, when transmit without “Between Burst Transfer Frames Delay”. (The SPCMD.SSLKP bit is 1 and the SPCR.BFDS bit is 1.)

Table 36.10 Relationship between the SPCMDm.SCKDEN bit, SPDECR.SCKDL[2:0] bits, and RSPCK delay

SPCMDm.SCKDEN bit	SPDECR.SCKDL[2:0] bits	RSPCK delay	
		Motorola-SPI	TI-SSP
0	000b to 111b	1 RSPCK	0 RSPCK
1	000b	1 RSPCK	1 RSPCK
	001b	2 RSPCK	2 RSPCK
	010b	3 RSPCK	3 RSPCK
	011b	4 RSPCK	4 RSPCK
	100b	5 RSPCK	5 RSPCK
	101b	6 RSPCK	6 RSPCK
	110b	7 RSPCK	7 RSPCK
	111b	8 RSPCK	8 RSPCK

(6) SSL negation delay (t2)

The SSL negation delay value of the SPI in master mode depends on the SPCMDm.SLNDEN bit setting and the SPDECR.SLNDL[2:0] bits setting. The SPI determines the SPCMDm register to be referenced by pointer control during a serial transfer, and determines an SSL negation delay using the SPCMDm.SLNDEN bit and SPDECR.SLNDL[2:0] bits, as listed in Table 36.11. For a definition of SSL negation delay, see section 36.3.5. Transfer Formats.

An SSL negation delay is inserted to only the last frame of the burst transmission, that is, transmit without between burst transfer frames delay. (SPCMD.SSLKP bit is 1 and SPCR.BFDS bit is 1).

Table 36.11 Relationship between the SPCMDm.SLNDEN bit, SPDECR.SLNDL[2:0] bits, and SSL negation delay (1 of 2)

SPCMDm.SLNDEN bit	SPDECR.SLNDL[2:0] bits	SSL negation delay
0	000b to 111b	1 RSPCK

Table 36.11 Relationship between the SPCMDm.SLNDEN bit, SPDECR.SLNDL[2:0] bits, and SSL negation delay (2 of 2)

SPCMDm.SLNDEN bit	SPDECR.SLNDL[2:0] bits	SSL negation delay
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(7) Next-access delay (t3)

The next-access delay value of the SPI in master mode depends on the SPCMDm.SPNDEN bit setting and the SPDECR.SPNDL[2:0] bits setting. The SPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and then determines a next-access delay during serial transfer using the SPCMDm.SPNDEN bit and SPDECR.SPNDL[2:0] bits, as listed in Table 36.12. For a definition of next-access delay, see section 36.3.5. Transfer Formats.

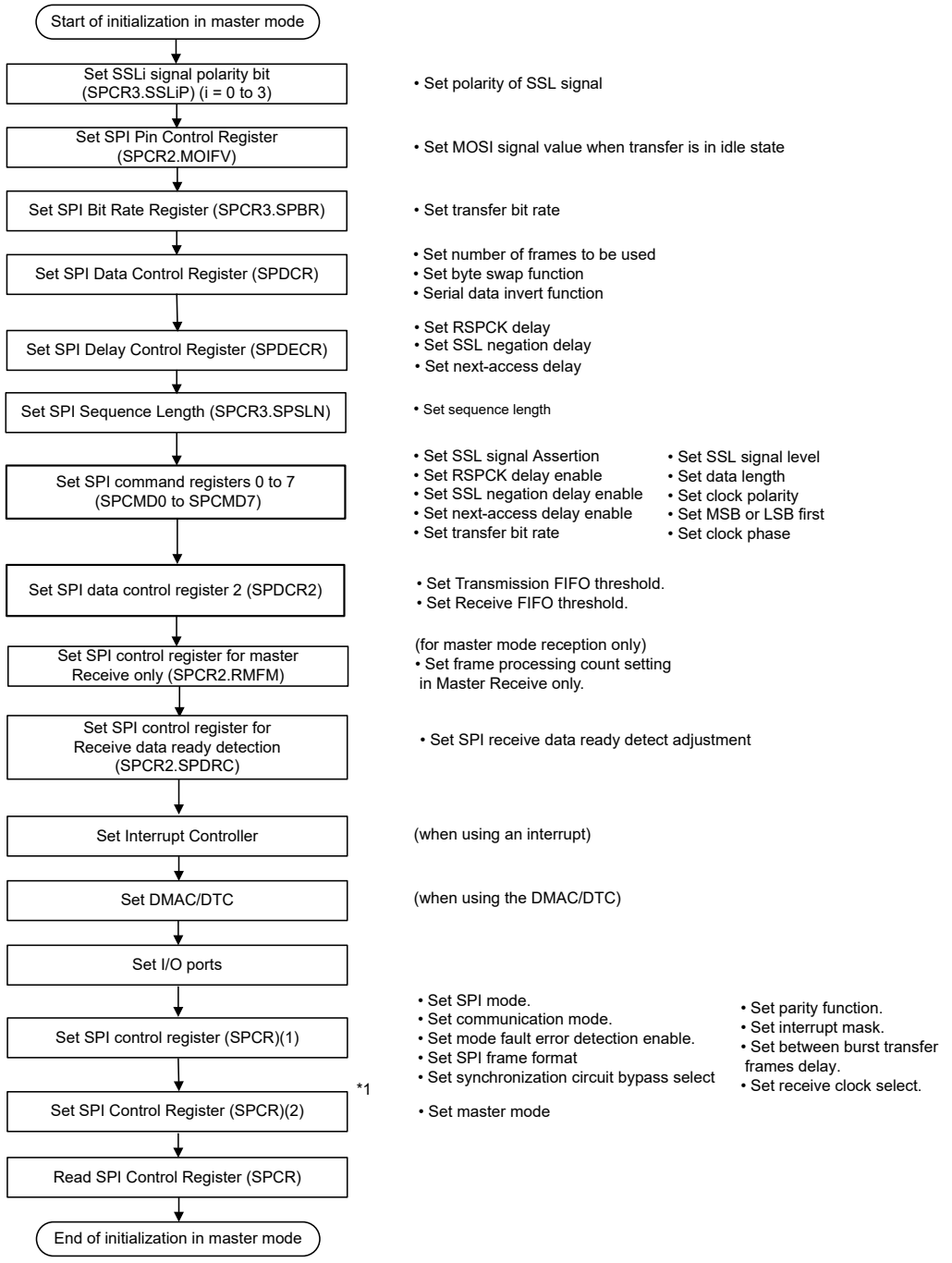
A next-Access delay is inserted to only the last frame of the burst transmission, that is, transmit without between burst transfer frames delay. (SPCMD.SSLKP bit is 1 and SPCR.BFDS bit is 1).

Table 36.12 Relationship between the SPCMDm.SPNDEN bit, SPDECR.SPNDL[2:0] bits, and next-access delay

SPCMDm.SPNDEN bit	SPDECR.SPNDL[2:0] bits	Next-access delay
0	000b to 111b	1 RSPCK + 5 TCLK
1	000b	1 RSPCK + 5 TCLK
	001b	2 RSPCK + 5 TCLK
	010b	3 RSPCK + 5 TCLK
	011b	4 RSPCK + 5 TCLK
	100b	5 RSPCK + 5 TCLK
	101b	6 RSPCK + 5 TCLK
	110b	7 RSPCK + 5 TCLK
	111b	8 RSPCK + 5 TCLK

(8) Initialization flow

Figure 36.63 shows an example of SPI initialization flow when the SPI is in master mode. For information on how to set up the Interrupt Controller Unit (ICU), DMAC or DTC, and I/O ports, see the descriptions given in the individual blocks.



Note 1. Regarding the setting of the SPI control register (SPCR), leave an interval of 1TCLK between setting (1) (initial setting other than MSTR bit) and setting (2) (MSTR setting).

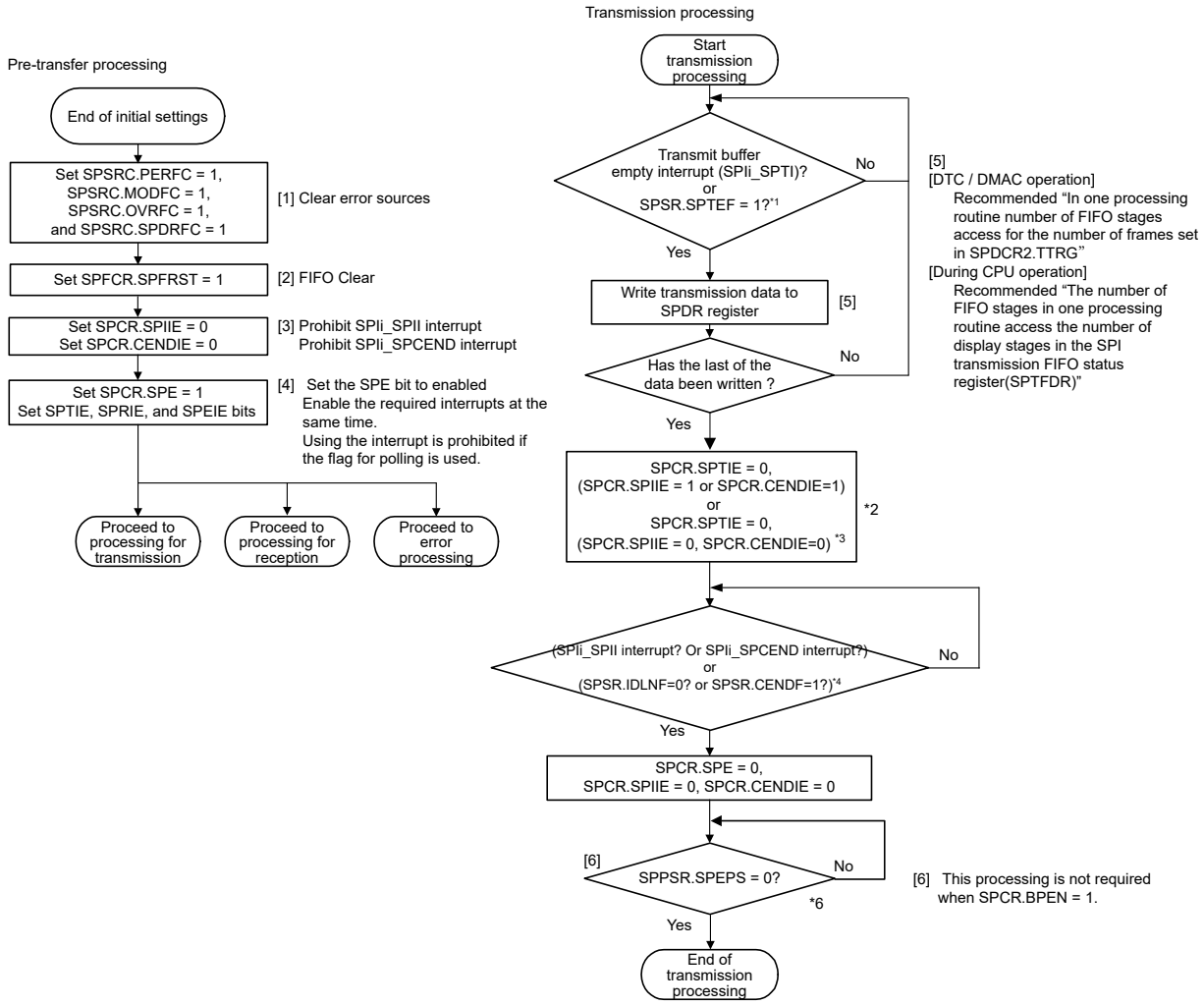
Figure 36.63 Example of initialization flow in master mode for SPI operation

(9) Software processing flow

Figure 36.64 to Figure 36.67 show examples of the software processing flow.

Transmit processing flow

When transmitting data, with the SPIi_SPII or SPIi_SPCEND interrupt enabled, the CPU is notified of the completion of data transmission after the last data write for transmission.

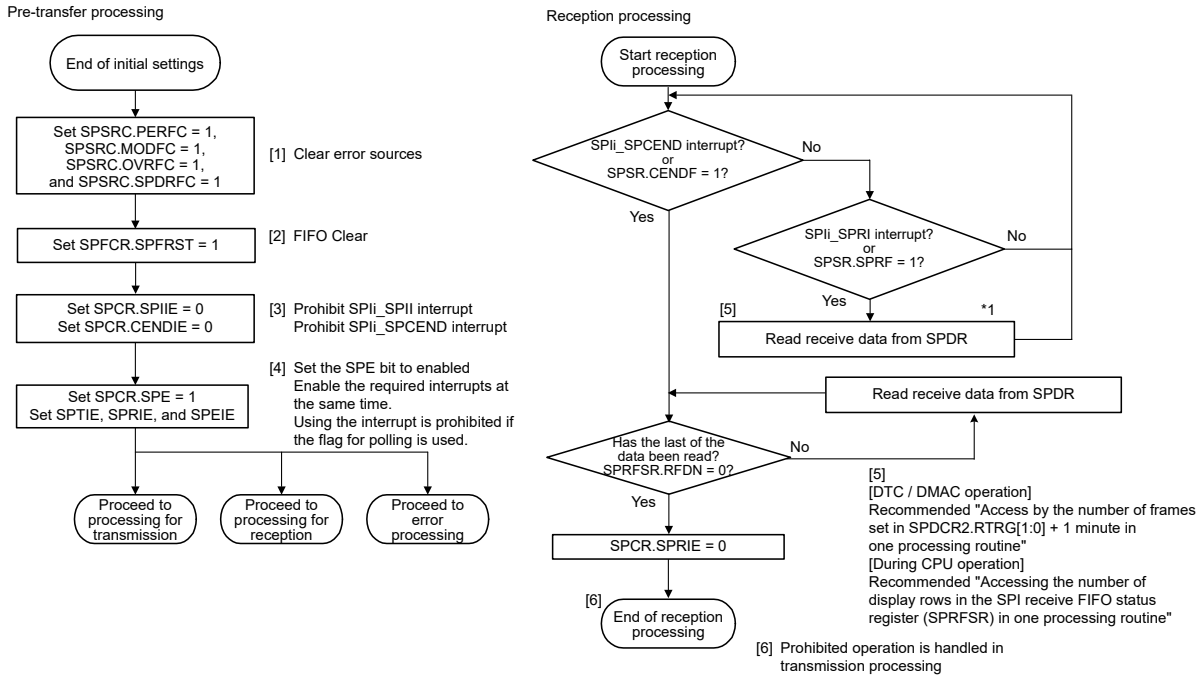


Note 1. Proceed to process of writing data for transmission after reading SPSR.SPTEF flag is 1, if the user uses the flag for polling.
Note 2. The user can know the communication end by SPIi_SPII interrupt (IDLNF flag) or SPIi_SPCEND interrupt (CENDF flag). Use either SPIi_SPII interrupt (IDLNF flag) or SPIi_SPCEND interrupt (CENDF flag).
Note 3. Set SPIi_SPII interrupt prohibited (SPCR.SPIIE = 0) and SPIi_SPCEND interrupt prohibited (SPCR.CENDIE = 0), if the user uses the flag for polling.
Note 4. Wait more than 1 PCLK, after writing data for transmission to SPDR, and before starting polling of SPSR.IDLNF or SPSR.CENDF, if the user uses the flag for polling.
Note 5. During CPU operation, Write 1 to SPI transmit buffer empty flag clear bit (SPSRC.SPTEFC) after one processing routine.
Note 6. When setting to 0 in slave mode and SPCR.BPEN = 0, check that SPPSR.SPEPS has become 0. (Not required when BPEN = 1.)

Figure 36.64 Transmission flow in master mode

Receive processing flow

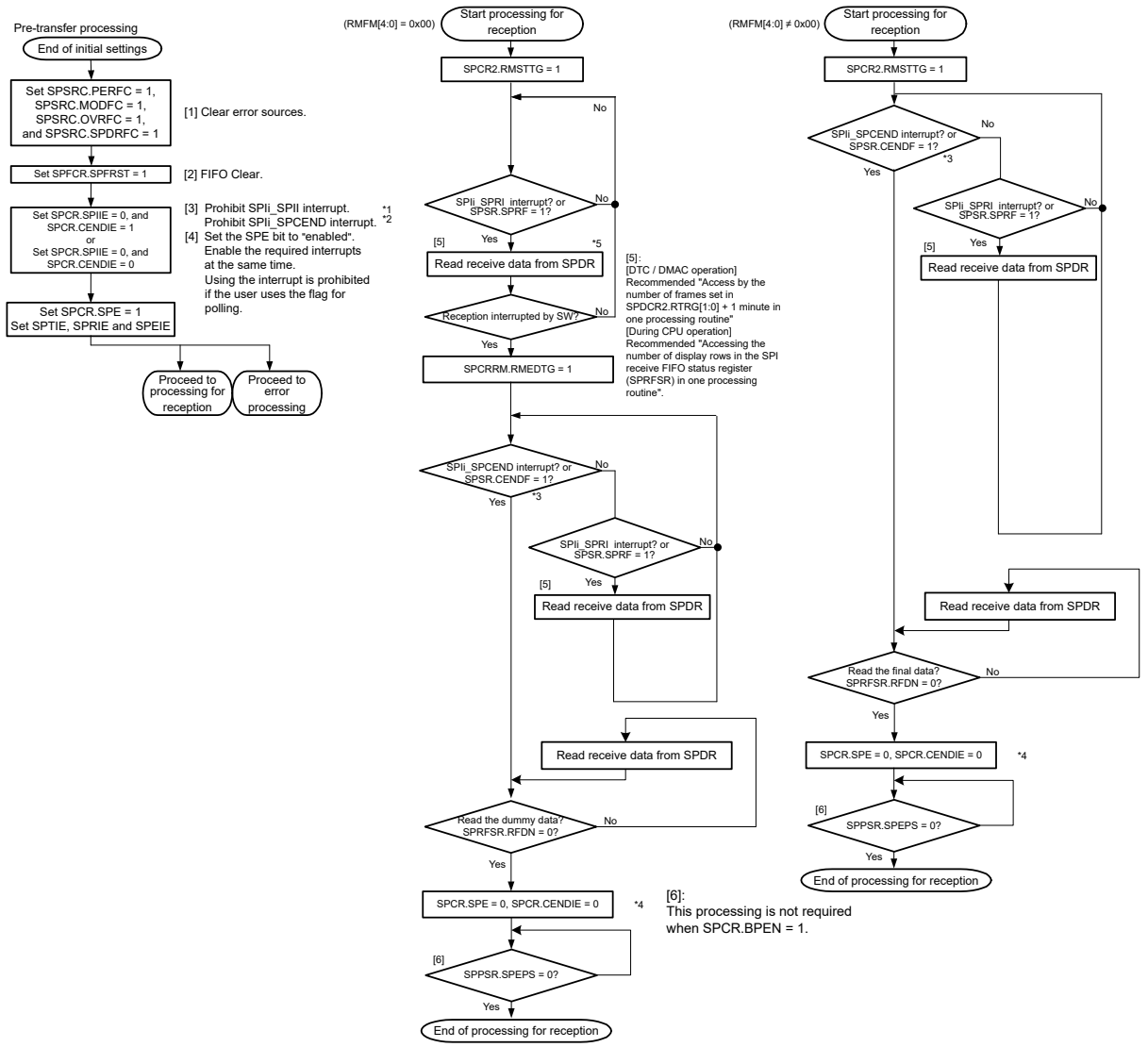
The SPI has receive only operation in slave mode.



Note 1. During CPU operation, write 1 to the SPSRC.SPRFC bit (SPI receive buffer full flag clear bit) after one processing routine.

Figure 36.65 Reception flow in master mode

Receive only processing flow in master mode



Note 1. The end of communication can be determined by the SPIi_SPCEND interrupt (CENDF flag). Use the SPIi_SPCEND interrupt (CENDF flag).
Note 2. Disable SPIi_SPCEND interrupt (SPCR.CENDIE = 0) during flag polling.
Note 3. A communication completion interrupt at reception only operation (Master) occurs at a predetermined timing under any of the following conditions.
- Frame number processing set by RMFM completed (RMFM[4:0] ≠ 0x00)
- RMEDTG = 1, when processing of the number of frames set by RMFM is not completed (including RMFM[4:0] = 0x00)

Note 4. Internal trigger is also cleared.
Note 5. During CPU operation, write 1 to the SPI receive buffer full flag clear bit (SPSRC.SPRFC) after one processing routine.

Figure 36.66 Software Processing Flowchart in Master Mode (Reception-only)

Error processing flow

The SPI detects the following errors:

- Mode fault error
- Underrun error
- Overrun error
- Parity error

When a mode fault error is generated, the SPCR.SPE bit is automatically cleared, stopping operations for transmission and reception. For errors from other sources, the SPCR.SPE bit is not cleared and operations for transmission and reception continue. Therefore, Renesas recommends clearing the SPCR.SPE bit to stop operations for errors other than mode fault errors. Not doing so leads to updating of the SPDCR2.SPECM[2:0] bits.

When an error is detected using an interrupt, clear the ICU.IELSRn.IR flag in the error processing routine. If this is not done, the ICU.IELSRn.IR flag might continue to indicate the SPIi_SPTI or SPIi_SPRI interrupt request. If the SPIi_SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the SPI.

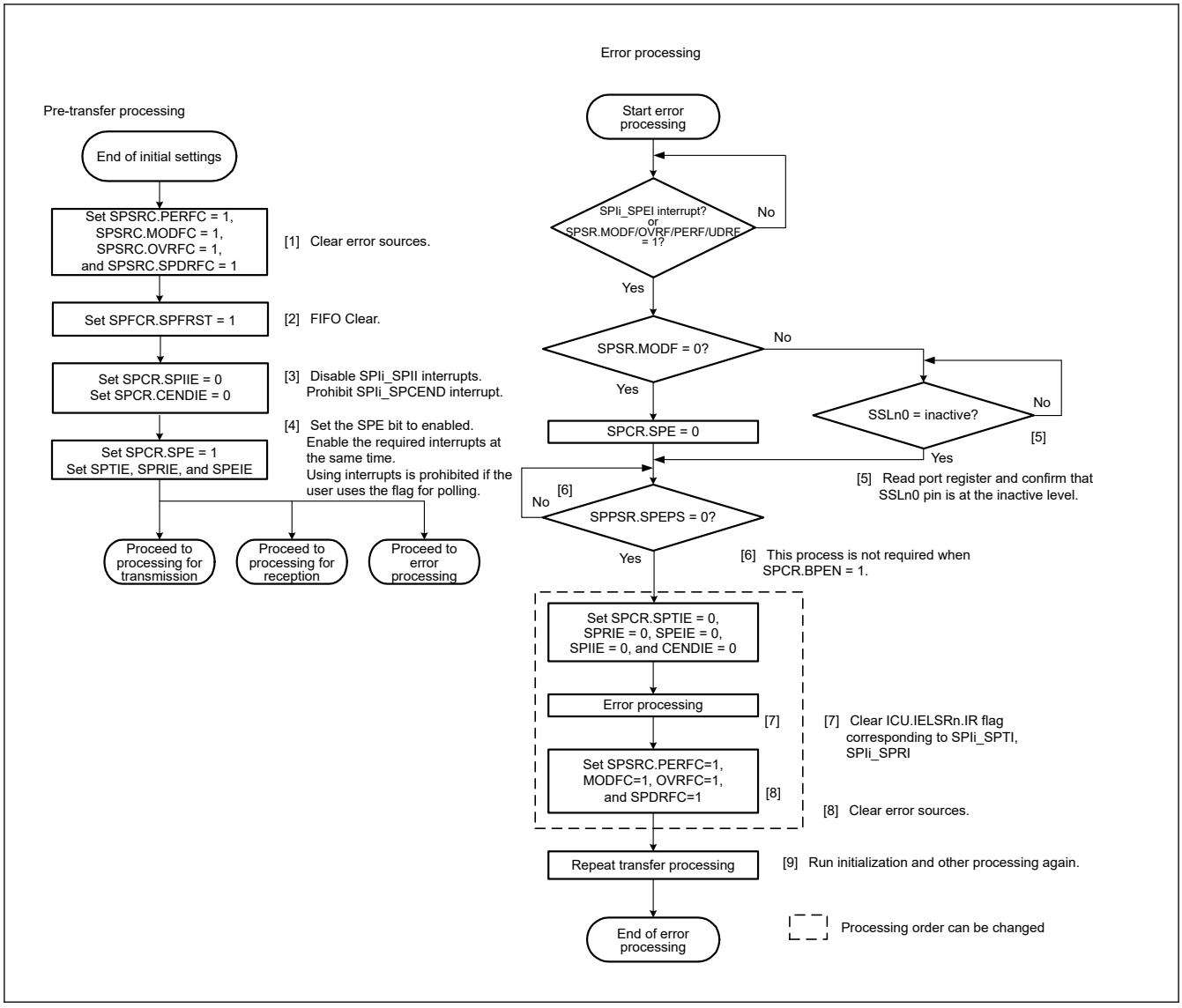


Figure 36.67 Error processing flow in master mode

36.3.12.2 Slave mode operation

(1) Starting a serial transfer

When the SPCMD0.CPHA bit is 0, if the SPI detects an SSLn0 input signal assertion, it must drive valid data to the MISOn output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLn0 input signal triggers the start of a serial transfer.

When the CPHA bit is 1, if the SPI detects the first RSPCKn edge in an SSLn0 signal asserted condition, it must drive valid data to the MISOn output signal. For this reason, when the CPHA bit is 1, the first RSPCKn edge in an SSLn0 signal asserted condition triggers the start of a serial transfer.

Regardless of the CPHA bit setting, the SPI drives the MISOn output signal on SSLn0 signal assertion. The data that is output by the SPI is either valid or invalid, depending on the CPHA bit setting.

For details on the SPI transfer format, see section 36.3.5. Transfer Formats. The polarity of the SSLn0 input signal depends on the SPCR3.SSL0P setting.

(2) Terminating a serial transfer

Regardless of the SPCMD0.CPHA bit setting, the SPI terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing. When the number of data stored in the receive FIFO is less than the number of FIFO stages, on termination of serial transfer, the SPI copies received data from the shift register to the receive buffer of the SPDR register. On termination of a serial transfer, the SPI changes the status of the shift register to empty, regardless of the receive buffer state. A mode fault error occurs if the SPI detects an SSLn0 input signal negation from the beginning of serial transfer to the end of serial transfer (see section 36.3.10. Error Detection).

The final sampling timing changes depending on the bit length of transfer data. In slave mode, the SPI data length is determined by the SPCMD0.SPB[4:0] bits setting. The polarity of the SSLn0 input signal is determined by the SPCR3.SSL0P bit setting. For details on the SPI transfer format, see section 36.3.5. Transfer Formats.

(3) Notes on single-slave operations

[In the Motorola-SPI case]

If the SPCMD0.CPHA bit is 0, the SPI starts serial transfers when it detects the assertion edge for an SSLn0 input signal. In the configuration shown in Figure 36.8, if the SPI is used in single-slave mode, the SSLn0 signal is fixed at an active state. Therefore, when the CPHA bit is set to 0, the SPI cannot correctly start a serial transfer. For the SPI to correctly execute transmit and receive operations in slave mode when the SSLn0 input signal is fixed at an active state, the CPHA bit must be set to 1. Do not fix the SSLn0 input signal if there is a requirement for setting the CPHA bit to 0.

[In the TI-SSP case]

When SPI is used as a single slave in the configuration shown in the Figure 36.8, the SSLn0 input signal is always fixed to the inactive state, so the SPI cannot start the serial transfer.

When using a single slave, use the configuration shown in the example in Figure 36.7.

(4) Burst transfer

[In the Motorola-SPI case]

If the SPCMD0.CPHA bit is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSLn0 input signal. When the CPHA bit is 1, the serial transfer period is the period from the first RSPCKn edge to the sampling timing for the reception of the final bit in an SSLn0 signal active state. Even when the SSLn0 input signal remains at the active level, the SPI can accommodate burst transfers, because it can detect the start of an access.

When the CPHA bit is 0, the second and subsequent serial transfers during burst transfer cannot be executed correctly.

[In the TI-SSP case]

In serial transfer, data transfer starts after the SSLn0 input signal is asserted for RSPCK 1 cycle. Since frame transfer starts from the SSLn0 input signal, SSLn0 must be asserted between frames.

(5) Initialization flow

Figure 36.68 shows an example of initialization flow for SPI operation when the SPI is in slave mode. For a description of how to set up the ICU, DMAC or DTC, and I/O ports, see the descriptions given in the individual blocks.

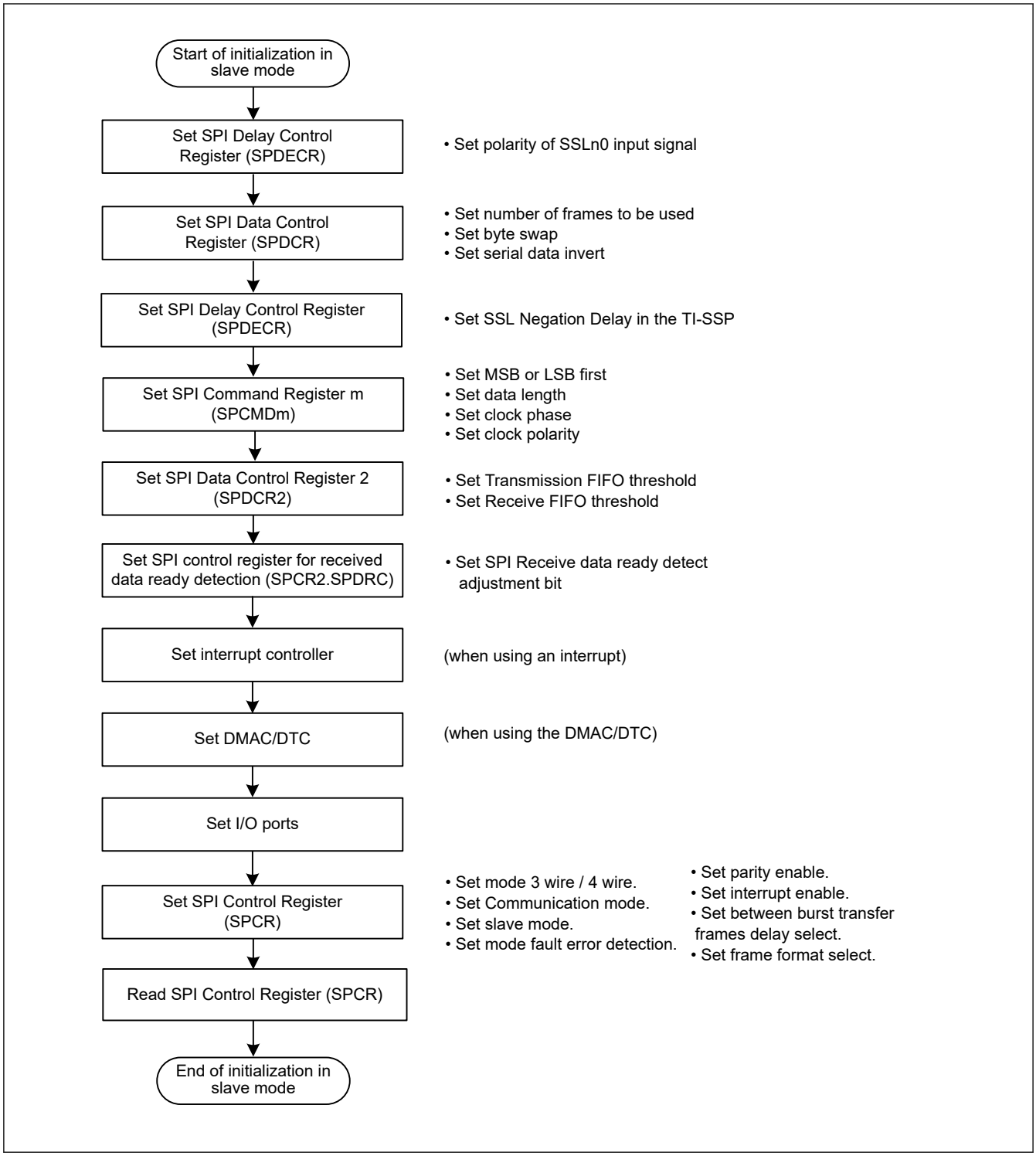
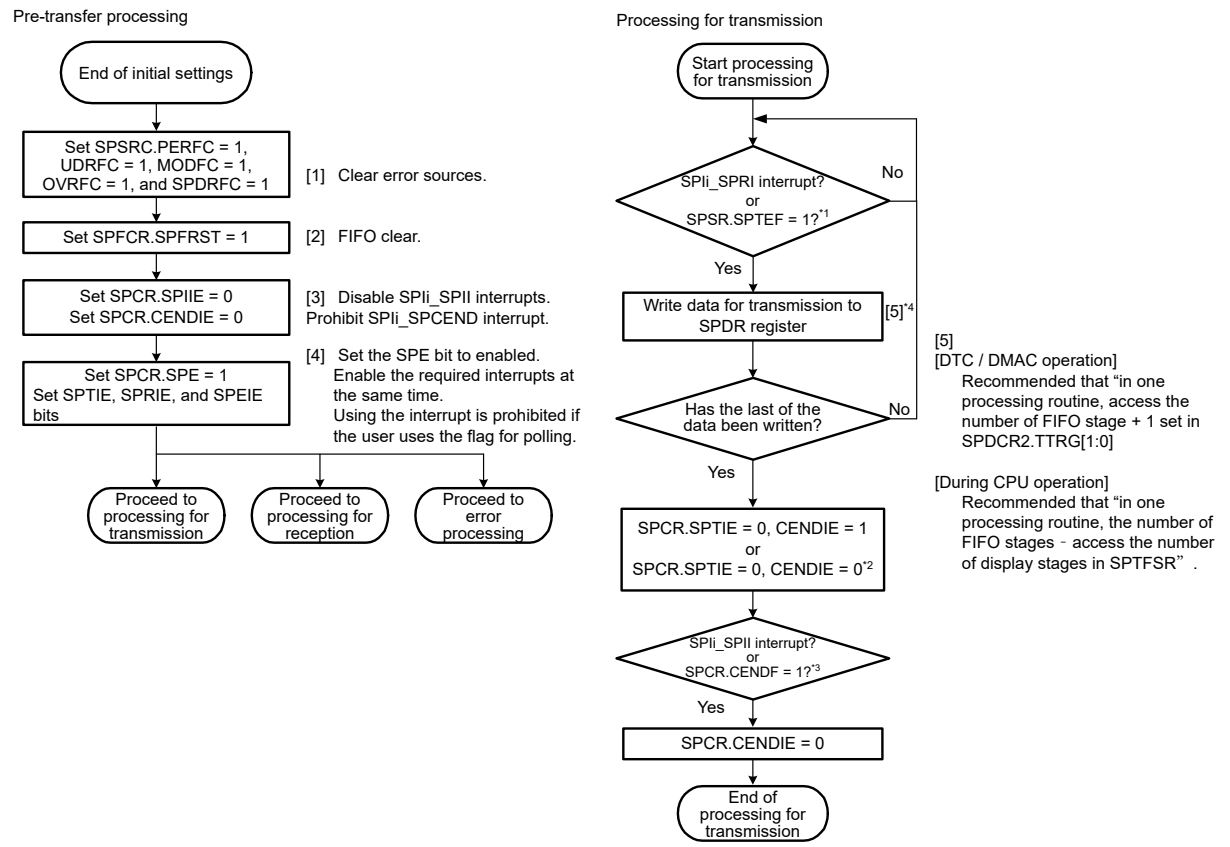


Figure 36.68 Example initialization flow in slave mode for SPI operation

(6) Software processing flow

Figure 36.69 to Figure 36.72 show examples of the flow of software processing.

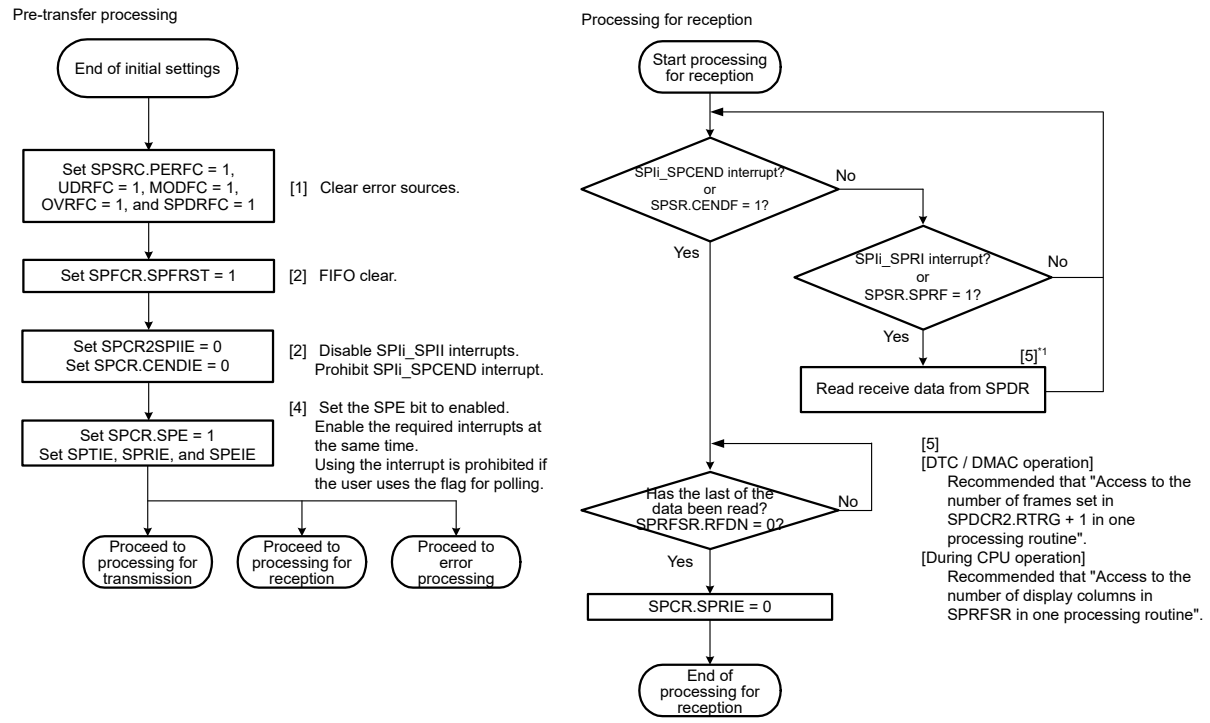
Transmit processing flow



Note 1. Before writing data for transmission to SPDR register, check the transmit buffer empty interrupt by reading SPSR.SPTEF flag, if the flag for polling is used.
Note 2. Set SPIi_SPCEND interrupt prohibited (SPCR.CENDIE = 0), if the user uses the flag for polling.
Note 3. Wait more than 1 PCLK, after writing data for transmission to SPDR, and before starting polling of SPSR.IDLNF or SPSR.CENDF, if the user uses the flag for polling.
Note 4. During CPU operation, write 1 to SPI transmit buffer empty flag clear bit (SPSRC.SPTEFC) after one processing routine.

Figure 36.69 Transmission flow in slave mode

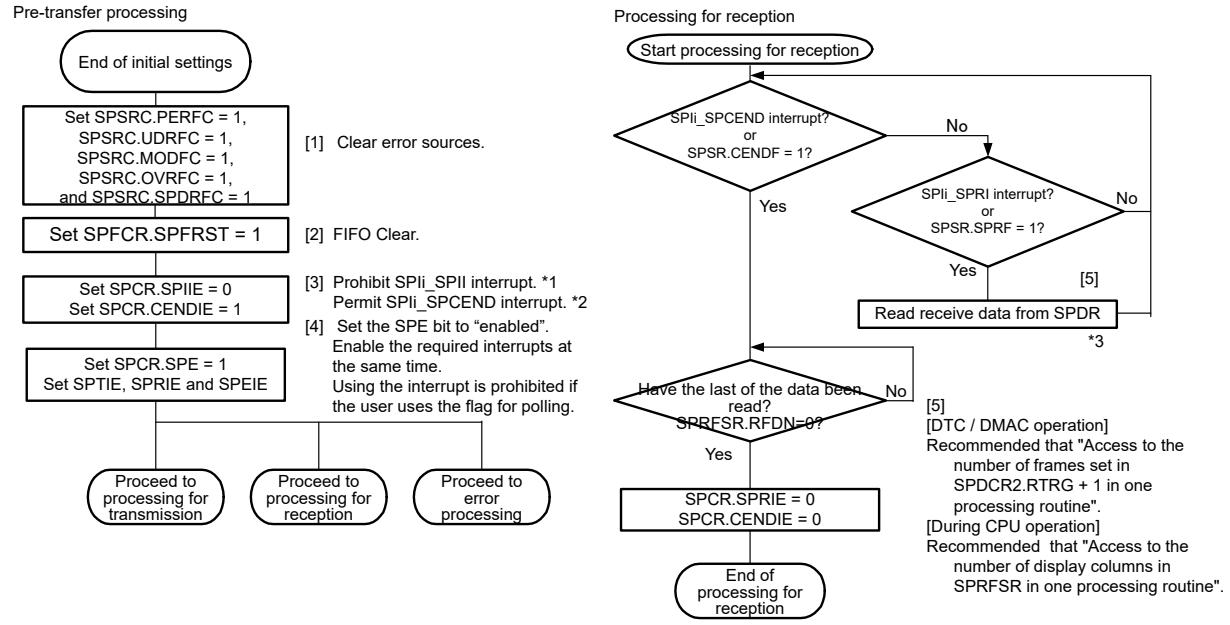
Receive processing flow



Note 1. During CPU operation, write 1 to the SPI receive buffer full flag clear bit (SPSRC.SPTEFC) after one processing routine.

Figure 36.70 Reception flow in slave mode

Master Reception-only processing flow



Note 1. The end of communication can be determined by the SPIi_SPCEND interrupt (CENDF flag). Use the SPIi_SPCEND interrupt (CENDF flag).
Note 2. Disable the SPIi_SPCEND interrupt (SPCR.CENDIE = 0) during flag polling.
Note 3. During CPU operation, write 1 to the SPI receive buffer full flag clear bit (SPSRC.SPTEFC) after one processing routine.

Figure 36.71 Software Processing Flowchart in Master Mode (Reception-only)

Error processing flow

In slave mode operation, even when a mode fault error is generated, the SPSR.MODF flag can be cleared regardless of the state of the SSLn0 pin.

When an error is detected by using an interrupt, clear the ICU.IELSRn.IR flag in the error processing routine. If this is not done, the ICU.IELSRn.IR flag might continue to indicate the SPIi_SPTI or SPIi_SPRI interrupt request. If the SPIi_SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the SPI.

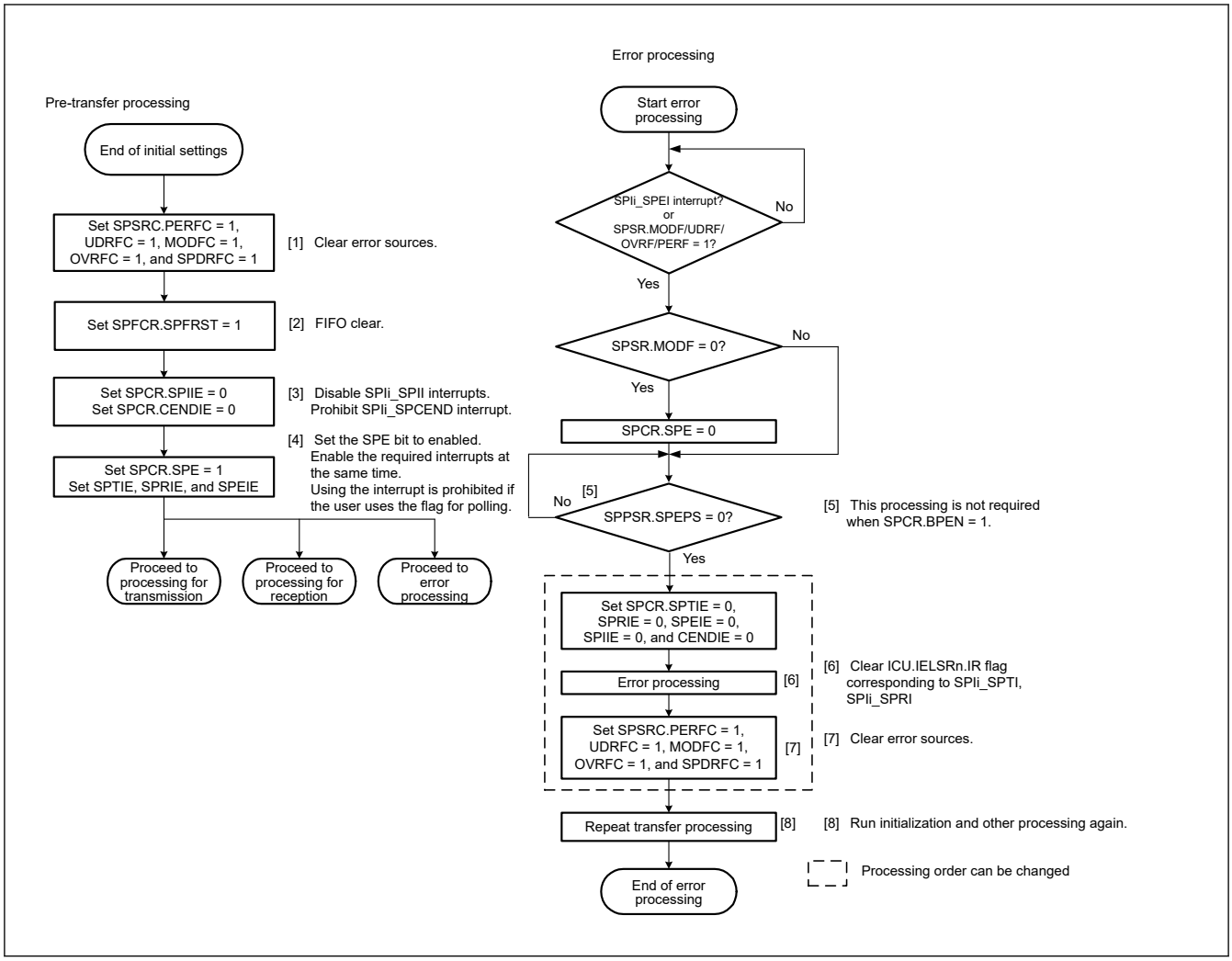


Figure 36.72 Error processing flow for slave mode

36.3.13 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the SPI. In clock synchronous operation, the SSLni pin is not used, and the RSPCKn, MOSIn, and MISOn pins handle communications. All SSLni pins are available as I/O port pins.

Although clock synchronous operation does not require the use of the SSLni pin, operation of the module is the same as in SPI operation. In both master mode and slave mode operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected, because the SSLni pin is not used.

Additionally, do not perform operation if clock synchronous operation is enabled when the SPCMDm.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0).

36.3.13.1 Master mode operation

(1) Starting serial transfer

When data is written to the SPI data register (SPDR) while the next transfer data is not set in the transmit FIFO, the SPI updates the transmit buffer (SPTXn, n = 0 to 3) data in SPDR.While the shift register is empty, the SPI copies transmit buffer data to the shift register to start serial transfer. After the SPI copies transmit data to the shift register, it changes the shift register status to full. Upon completion of serial transfer, the SPI changes the shift register status to empty. The shift register status cannot be monitored.

For details about the SPI transfer format, see section 36.3.5. Transfer Formats. In clock synchronous operation, however, the SSLn0 output signal is not used for communication.

(2) Terminating serial transfer

The SPI terminates the serial transfer after transmitting an RSPCKn edge corresponding to the sampling timing. If the number of data stored in the receive FIFO < the number of FIFO stages, on termination of serial transfer, the SPI copies data from the shift register to the receive buffer of the SPI Data Register (SPDR).

The final sampling timing varies depending on the bit length of transfer data. In master mode, the SPI data length depends on the SPCMDm.SPB[4:0] bits setting. Transfer in clock synchronous operation is conducted without the SSLn0 output signal. For details on the SPI transfer format, see section 36.3.5. Transfer Formats.

(3) Sequence control

The transfer format used in master mode is determined by the SPCR3, SPCMDm, and SPDECR registers. Although the SSLni signals are not output in clock synchronous operation, these settings are valid.

The SPCR3.SPSLN[2:0] bits determine the sequence configuration for serial transfers that are executed by the SPI in master mode. The following parameters are specified in the SPCMDm register:

- SSLni output signal value
- MSB or LSB first
- Data length
- Some of the bit rate settings
- RSPCKn polarity and phase
- Whether SPDECR.SCKDL is to be referenced
- Whether SPDECR.SLNDL is to be referenced
- Whether SPDECR.SPNDL is to be referenced

SPCR3.SPBR holds some of the bit rate settings such as SPDECR.SCKDL, an SPI clock delay value, SPDECR.SLNDL, an SSL negation delay, and SPDECR.SPNDL, a next-access delay value.

Based on the sequence length that is assigned to SPCR3, the SPI makes up a sequence comprised of a part or all of SPCMDm register. The SPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPDCR2.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the SPI function is enabled, the SPI loads the pointer to the commands in SPCMD0 register, and incorporates the SPCMD0 register setting into the transfer format at the beginning of serial transfer. The SPI increments the pointer each time the next-access delay period for a data transfer ends. On completion of the serial transfer that corresponds to the final command comprising the sequence, the SPI sets the pointer to the SPCMD0 register, and in this manner the sequence is executed repeatedly.

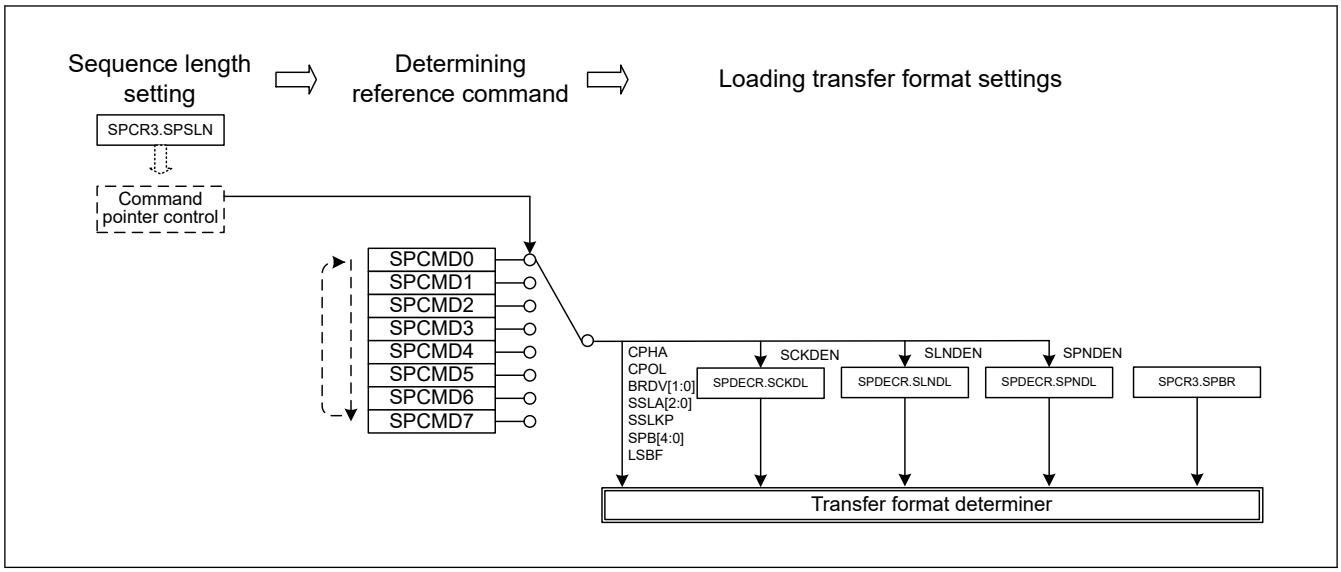


Figure 36.73 Procedure for determining the form of serial transmission in master mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

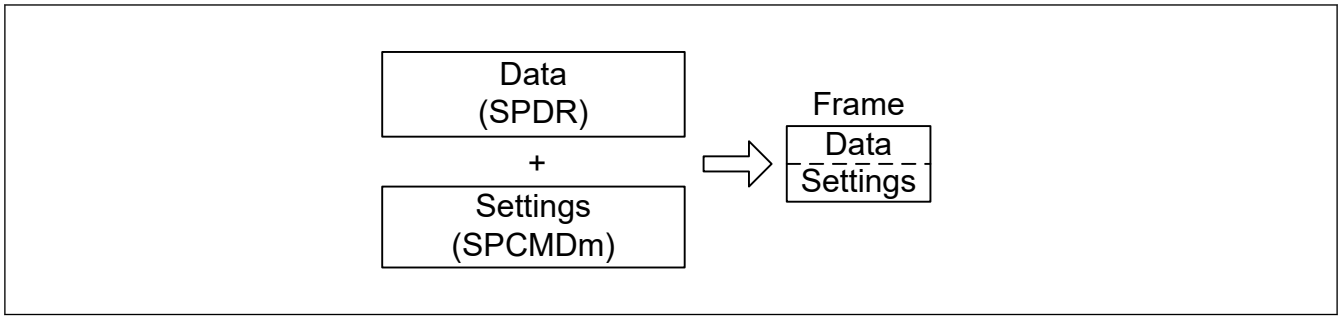


Figure 36.74 Conceptual diagram of frames

Figure 36.75 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings.

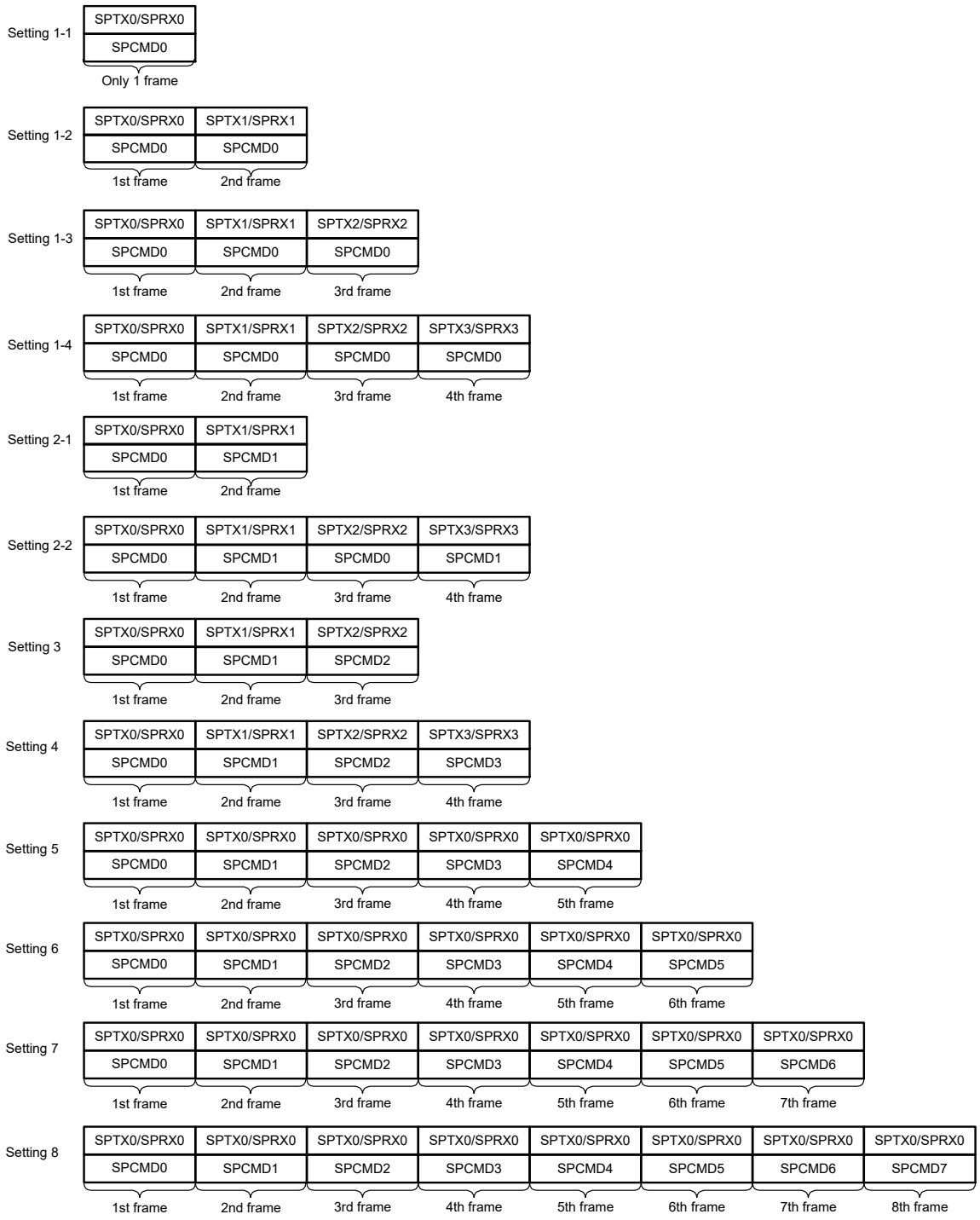


Figure 36.75 Correspondence between SPI Command Register and transmit and receive buffers in sequence operations

(4) Initialization flow

Figure 36.76 shows an example of initialization flow for clock synchronous operation when the SPI is used in master mode. For information on how to set up the ICU, DMAC or DTC, and I/O ports, see the descriptions given in the individual blocks.

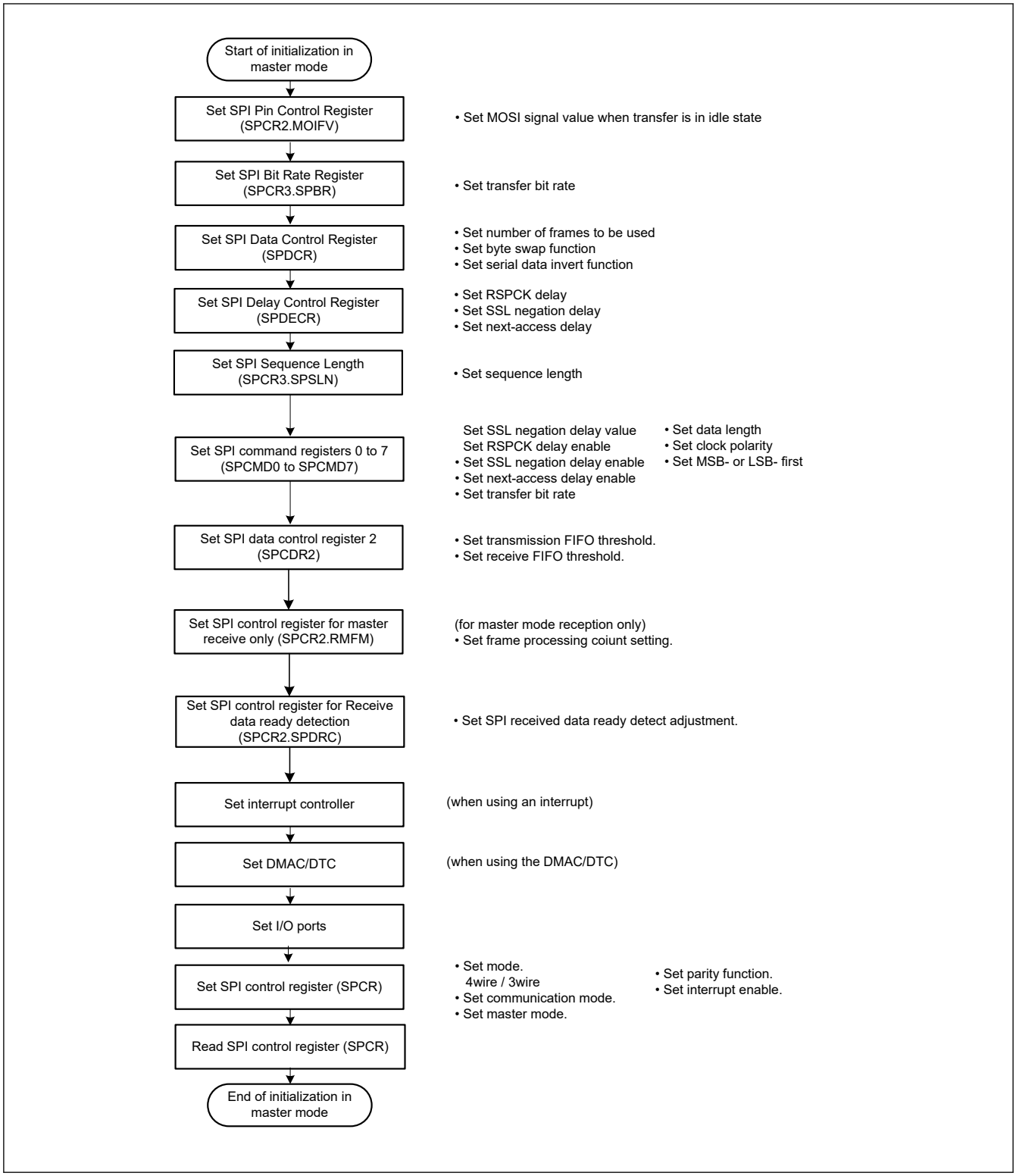


Figure 36.76 Example of initialization flow in master mode for clock synchronous operation

(5) Software processing flow

Software processing during clock synchronous master operation is the same as that for SPI master operation. For details, see (9) Software processing flow in section 36.3.12.1. Master mode operation. Mode fault errors do not occur in clock synchronous operation.

36.3.13.2 Slave mode operation

(1) Starting serial transfer

When the SPCR.SPMS bit is 1, the first RSPCKn edge triggers the start of a serial transfer in the SPI, and the SPI drives the MISOn output signal. The SSLn0 input signal is not used in clock synchronous operation. For details on the SPI transfer format, see section 36.3.5. Transfer Formats.

(2) Terminating serial transfer

The SPI terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing. When the number of data stored in the receive FIFO < the number of FIFO stages, on termination of serial transfer, the SPI copies received data from the shift register to the receive buffer of the SPDR register. On termination of a serial transfer, the SPI changes the status of the shift register to empty regardless of the receive buffer.

The final sampling timing changes depending on the bit length of transfer data. In slave mode, the SPI data length depends on the SPCMD0.SPB[4:0] bits setting. For details on the SPI transfer format, see section 36.3.5. Transfer Formats.

(3) Initialization flow

Figure 36.77 shows an example of initialization flow for clock synchronous operation when the SPI is used in slave mode. For a description of how to set up the ICU, DMAC or DTC, and I/O ports, see the descriptions given in the individual blocks.

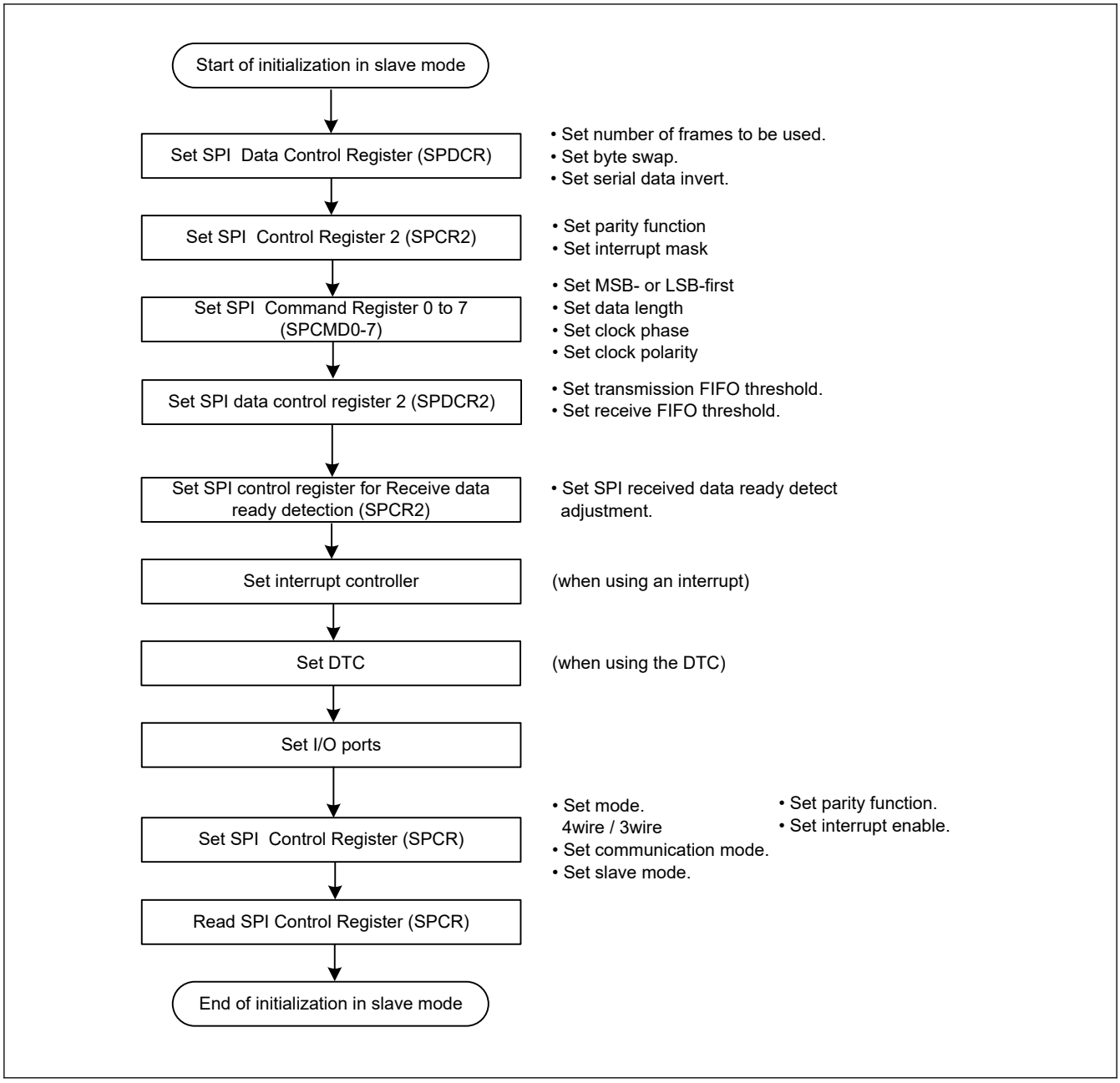


Figure 36.77 Example of initialization flow in slave mode for clock synchronous operation

(4) Software processing flow

Software processing during clock synchronous slave operation is the same as that for SPI slave operation. For details, see (6) Software processing flow. Mode fault errors do not occur in clock synchronous mode.

36.3.14 Loopback Mode

When 1 is written to the SPCR2.SPLP2 bit or SPCR2.SPLP bit, the SPI shuts off the path between the MISOn pin and the shift register if the SPCR.MSTR bit is 1, or between the MOSIn pin and the shift register if the SPCR.MSTR bit is 0, and connects the input and output paths of the shift register, establishing a loopback mode. The SPI does not shut off the path between the MOSIn pin and the shift register if the SPCR.MSTR bit is 1, or between the MISOn pin and the shift register if the SPCR.MSTR bit is 0. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data for the SPI or the reversed transmit data becomes the received data for the SPI.

Table 36.13 lists the relationship between the SPLP2 and SPLP bits and the received data. Figure 36.78 shows the configuration of the shift register I/O paths when the SPI in master mode is set to loopback mode (SPCR2.SPLP2 = 0, SPCR2.SPLP = 1).

Table 36.13 SPLP2 and SPLP bit settings and received data

SPCR2.SPLP2 bit	SPCR2.SPLP bit	Received data
0	0	Input data from the MOSIn pin or MISOn pin
0	1	Inverted transmit data
1	0	Transmit data
1	1	Transmit data

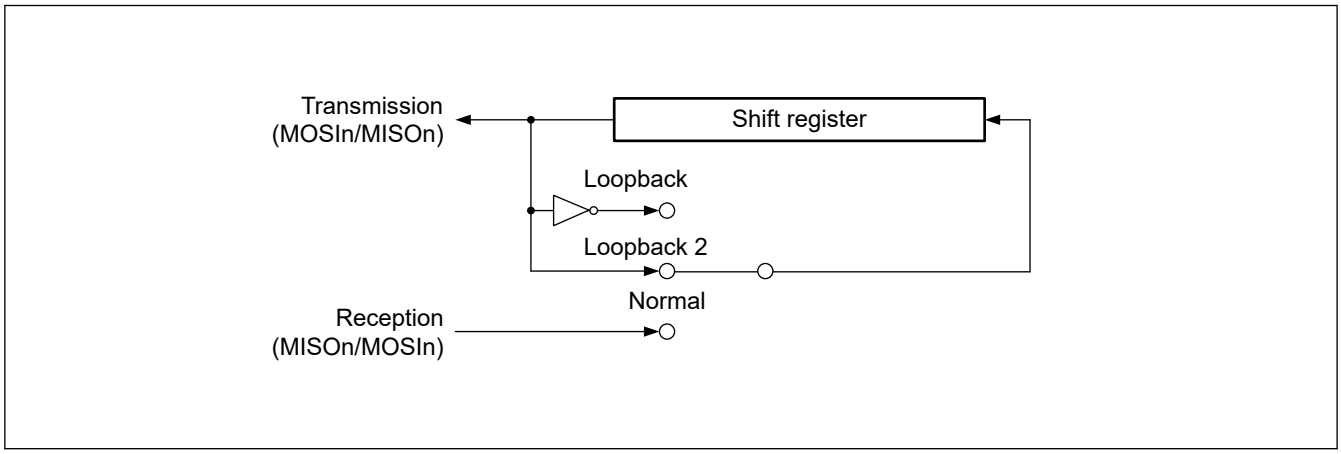


Figure 36.78 Configuration of shift register I/O paths in loopback mode for master mode

36.3.15 Self-Diagnosis of Parity Bit Function

The parity circuit consists of a parity bit adding unit used for transmit data and an error detecting unit used for received data. To detect defects in the parity bit adding unit and error detecting unit, the parity circuit performs self-diagnosis as shown in Figure 36.79.

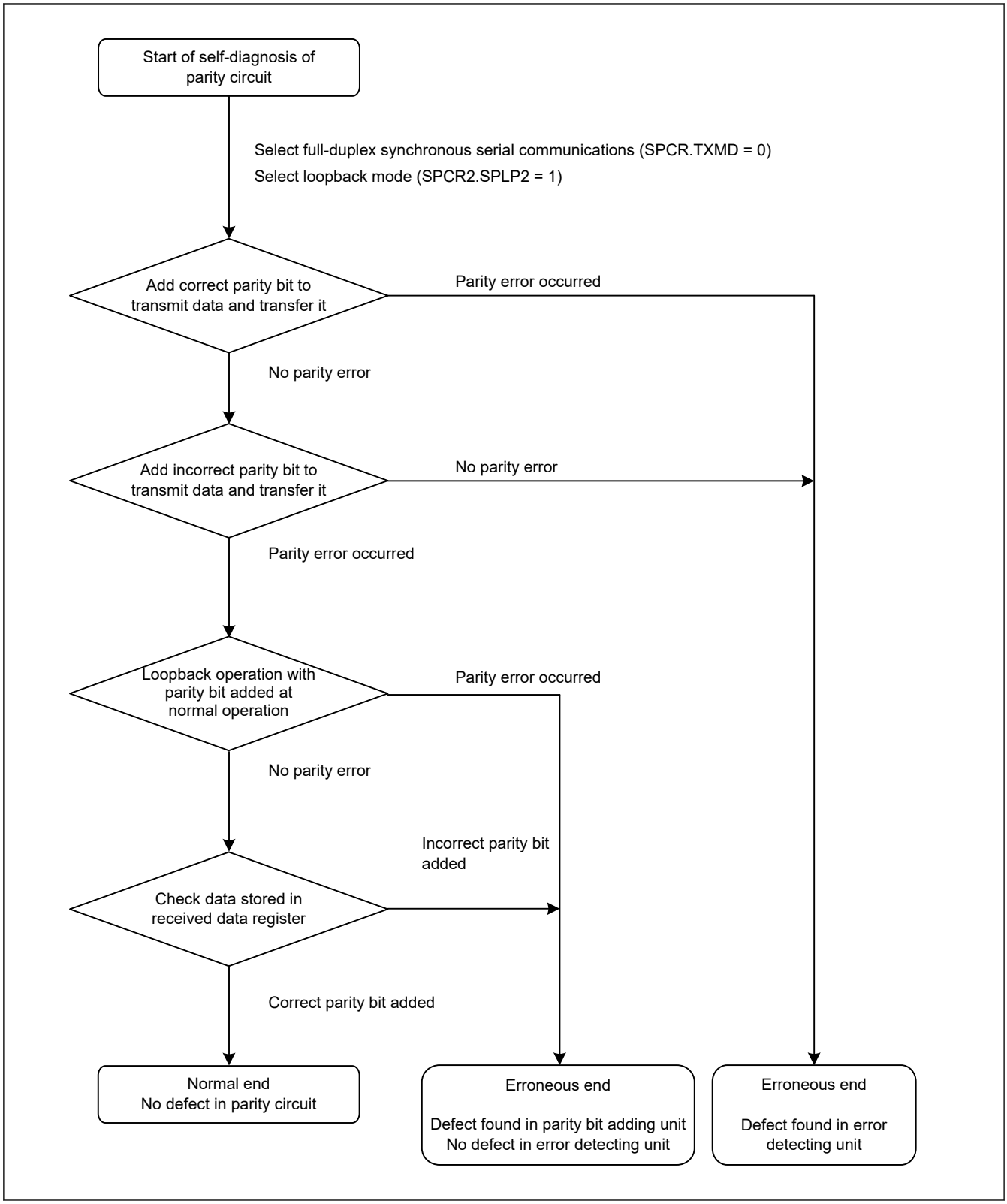


Figure 36.79 Self-diagnosis flow for parity circuit

36.3.16 Interrupt Sources

The SPI has the following interrupt sources:

- Receive buffer full
- Transmit buffer empty

- SPI error (mode-fault, underrun, overrun, or parity error)
- SPI idle
- Communication-end

The DMAC or DTC can be activated by the receive buffer full or transmit buffer empty interrupt to perform data transfer.

Because the vector address for the SPIi_SPEI (SPI error interrupt) is allocated to interrupt requests on mode-fault, underrun, overrun, and parity errors, the actual interrupt source must be determined from the flags. Interrupt sources for the SPI are listed in Table 36.14. An interrupt is generated on satisfaction of one of the interrupt conditions in Table 36.14. Clear the receive buffer full and transmit buffer empty sources through a data transfer.

When using the DMAC or DTC to perform data transmission and reception, you must first set up the DMAC or DTC to be in a transfer-enabled status before setting the SPI. For information on setting up the DMAC or DTC, see section 16, DMA Controller (DMAC) and section 17, Data Transfer Controller (DTC).

If the conditions for generating a transmit buffer empty or receive buffer full interrupt occur while the ICU.IELSRn.IR flag is 1, the interrupt is not output as a request for the ICU but is retained internally (the capacity for retention is one request per source). A retained interrupt request is output when the ICU.IELSRn.IR flag becomes 0. A retained interrupt request is automatically discarded when it is output as an actual interrupt request. The interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) for an internally retained interrupt request can also be set to 0.

Table 36.14 SPI interrupt sources

Interrupt source	Symbol	Interrupt condition	DTC/DMAC activation
Receive buffer full	SPIi_SPRI	The receive buffer becomes full (SPSR.SPRF flag is 1) while the SPCR.SPRIE bit is 1 or The receive data become ready (SPSR.SPDRF flag is 1) while the SPCR.SPDRES bit is 0	Possible
Transmit buffer empty	SPIi_SPTI	The transmit buffer becomes empty (SPSR.SPTEF flag is 1) while the SPCR.SPTIE bit is 1	Possible
SPI error (mode-fault, underrun, overrun, or parity error)	SPIi_SPEI	The SPSR.MODF, OVRF, or PERF flag sets to 1, or the SPSR.SPDRES and SPDRF flag set to 1 while the SPCR.SPEIE bit is 1	Impossible
SPI idle	SPIi_SPII	The SPSR.IDLNF flag sets to 0 while the SPCR.SPIIE bit is 1	Impossible
Communication-end	SPIi_SPCEND	CENDIE = 1 and CENDF = 1	Impossible

36.4 Event Link Controller Event Output

The Event Link Controller (ELC) can produce the following event output signals:

- Receive buffer full event output
- Transmit buffer empty event output
- Mode-fault, underrun, overrun, or parity error event output
- SPI idle event output
- Transmission-completed event output

The event link output signal is output regardless of the interrupt enable bit setting.

36.4.1 Receive Buffer Full Event Output

When the number of data stored in the receive FIFO > the threshold value, or when the number of data stored in the receive FIFO ≤ the threshold value and SPDRES = 0 has elapsed after writing to the receive FIFO, and the SPDRC [7:0] has elapsed outputs an event.

36.4.2 Transmit Buffer Empty Event Output

An event is output when the number of empty transmission FIFO stage > the threshold or when the SPCR.SPE bit changes from 0 to 1.

36.4.3 Mode-Fault, Underrun, Overrun, Parity Error, or received data ready Event Output

This event signal is output when mode-fault, underrun, overrun, or parity error is detected. See section 36.5.4. Constraints on Mode-Fault, Underrun, Overrun, Parity Error, or Receive Data Ready Event Output if using this event signal.

(1) Mode-fault

Table 36.15 lists the conditions for occurrence of a mode-fault event.

Table 36.15 Conditions for mode-fault occurrence

SPI mode	SPCR.MODFEN bit	SSLn0 pin	Remarks
SPI operation (SPMS = 0) Slave (SPCR.MSTR = 0) Motorola-SPI (SPCR.SPFRF = 0)	1	Not active	Event is output only when the SSLn0 pin is deactivated during transmission
SPI operation (SPMS = 0) Slave (SPCR.MSTR = 0) TI-SSP (SPCR.SPFRF = 1)	1	active	Event is output only when the SSLn0 pin is activated during transmission

(2) Underrun

This event signal is output in response to an underrun when a serial transfer starts while the transmission data is not ready, and the value of the SPCR.MSTR bit is 0 and the SPCR.SPE bit is 1. Under these conditions, the MODF and UDRF flags are set to 1.

(3) Overrun

This event signal is output in response to an overrun when a serial transfer completes while the receive buffer contains unread data and the value of the SPCR.TXMD[1:0] bits are 00b or 10b. Under these conditions, the OVRF flag is set to 1.

(4) Parity error

This event signal is output in response to a parity error detected on completion of a serial transfer while the value of the SPPE bit in SPCR is 1.

(5) Receive Data Ready

When TXMD[1:0] of SPCR = 00b or 10b and SPDRES = 1 as the receive data ready event output condition, the number of data stored in the receive FIFO is received after writing the receive FIFO. An event will be output when the set value of SPDRC[7:0] has elapsed while the number is less than the FIFO threshold.

36.4.4 SPI Idle Event Output

(1) In master mode

In Transmit-Receive / Transmit-only master mode, an event is output when the IDLNF flag in SPSR changes from 1 to 0.

The IDLNF flag changes from 1 to 0 only when either of the conditions 1) and 2) below is met.

1. The SPE bit in SPCR is cleared to 0 (SPI initialized) during transmission.
2. All of the following three conditions are met.
   - The transmit buffer (SPTXn, n = 0 to 3) is empty (next transfer data has not been set).
   - The SPCP [2:0] bits in SPSR are 000b (at the start of sequence control).
   - Operation completed by the next access delay (when the master main state machine transitions to the idle state).

In receive only master mode

Any of the following 2 conditions is met.

1. SPE bit of SPCR is 0 (SPI initialization)
2. When any of the following is met
   - When RMFM [4:0] = 0x00, after writing 1 to RMEDTG, operation completed by the next access delay (when the master main state machine transitions to the idle state).

- When RMFM [4:0] ≠ 0x00, after writing 1 to RMEDTG, operation completed by the next access delay (when the master main state machine transitions to the idle state).
- When RMFM [4:0] ≠ 0x00, the SPI internal sequencer transitions to the idle state after operation completed by the next access delay (when the master main state machine transitions to the idle state).

(2) In slave mode

In slave mode, an event is output when the SPCR.SPE bit is set to 0 (SPI is initialized).

36.4.5 Communication End Event Output

In master mode, an event is output when the IDLNF flag (SPI idle flag) changes from 1 to 0. In slave mode, an event occurs with conditions shown in Table 36.16 and Table 36.17

Table 36.16 Communication End Event Generating Conditions (transmit-receive/transmit slave mode)

	Transmit Buffer Status	Shift Register Status	Others
SPI operation (SPMS = 0, SPFRF = 0)	Empty	Empty	SSLn0 input is negated
SPI operation (SPMS = 0, SPFRF = 1)	Empty	Empty	SSL negation delay completed
Clock synchronous operation (SPMS = 1)	Empty	Empty	The last even edge of RSPCK of last data was detected (CPHA = 1)

Table 36.17 Communication End Event Generating Conditions (receive only slave mode)

	Others
SPI operation (SPMS = 0, SPFRF = 0)	After storing the frames corresponding to the SPFC setting value in the receive buffer, negate SSLn0 input.
SPI operation (SPMS = 0, SPFRF = 1)	After storing the frames corresponding to the SPFC setting value in the receive buffer, SSL negation delay completed
Clock synchronous operation (SPMS = 1)	RSPCK last even edge detection when receiving the last frame for the SPFC set value (CPHA = 1)

Regardless of master mode or slave mode, no event is output when 0 is written to the SPCR.SPE bit during transmission or when the SPCR.SPE bit is cleared due to a mode fault error or an underrun error.

A communication end event is output at the following timing. The communication end event output timing in master operation is omitted because it is output at the same timing as an idle event.

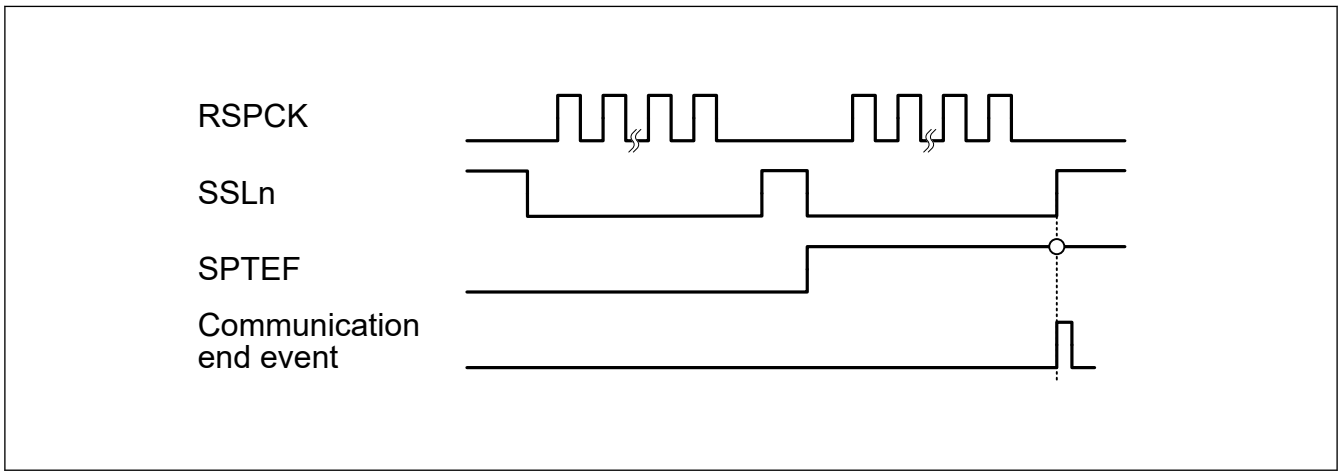


Figure 36.80 Communication End Event Output Timing (Transmit slave mode, Motorola SPI operation)

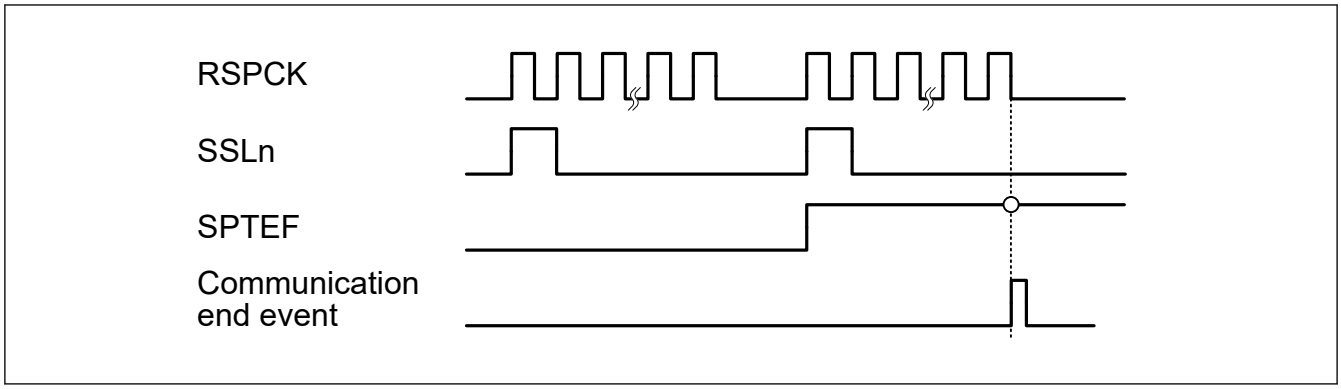


Figure 36.81 Communication End Event Output Timing (Transmit slave mode, TI-SSP Operation)

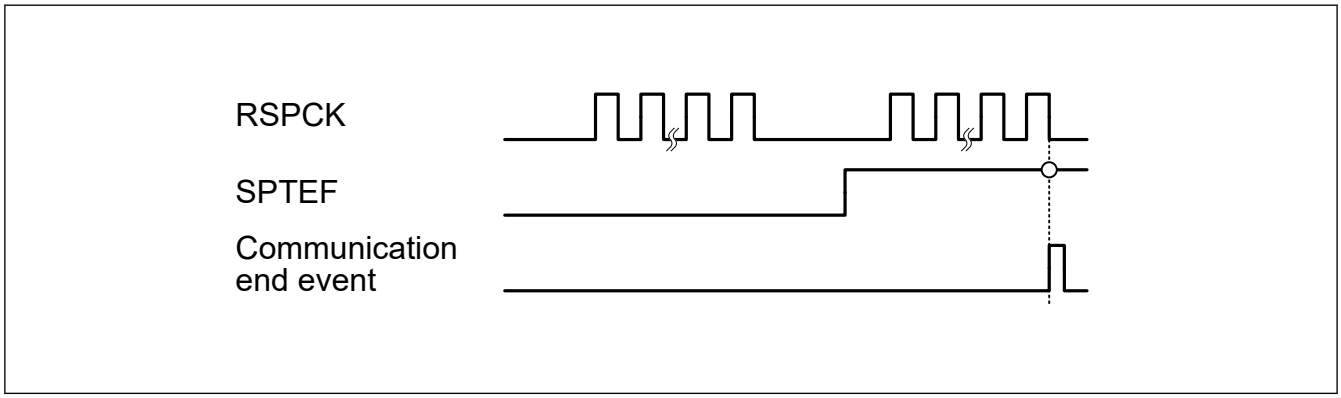


Figure 36.82 Communication End Event Output Timing (Transmit slave mode, Clock Synchronous Operation)

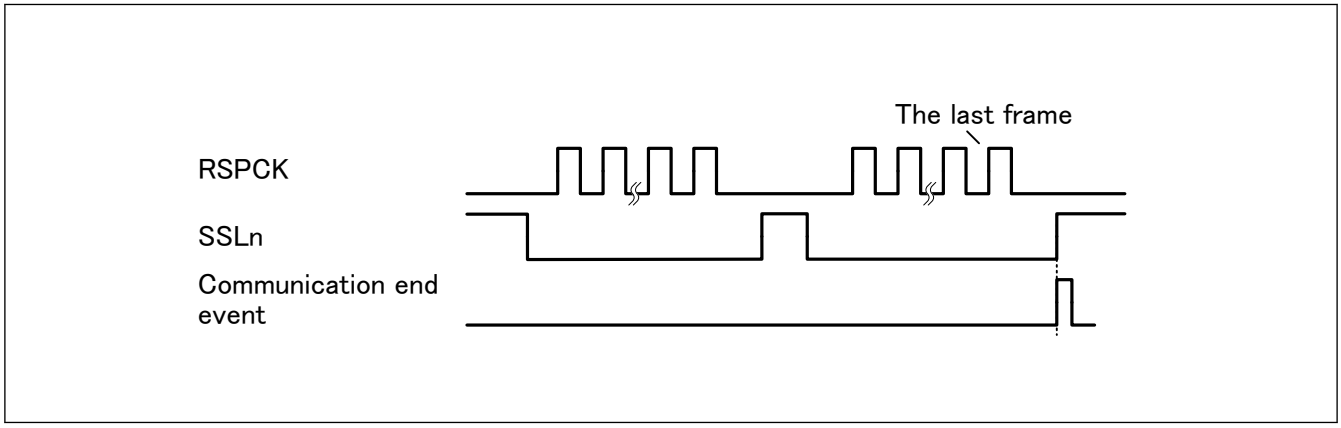


Figure 36.83 Communication End Event Output Timing (Receive only slave mode, Motorola SPI operation)

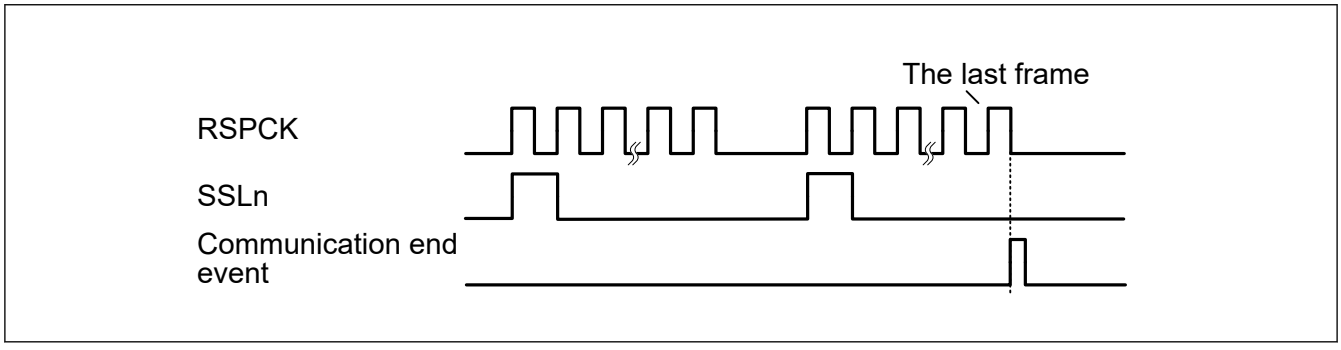


Figure 36.84 Communication End Event Output Timing (Receive only slave mode, TI-SSP Operation)

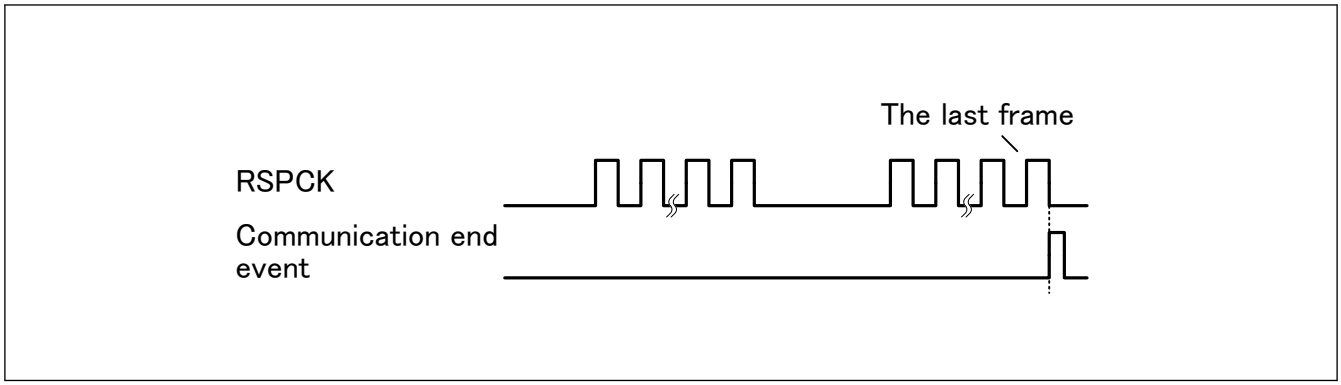


Figure 36.85 Communication End Event Output Timing (Receive only slave mode, Clock Synchronous Operation)

36.4.6 Synchronization bypass function

SPI has an internal clock (PCLK) and an operation clock (TCLK), and each has its own operation circuit. Therefore, a synchronization circuit is inserted between the signals between different clocks, and a signal delay between different clocks requires a synchronization delay time.

However, the synchronization circuit can be bypassed by the BPEN = 1 of the SPI control register (SPCR) only when the same clock is input as the internal bus clock and the operation clock. In this case, the synchronization delay time is excluded, and responsiveness is improved.

In addition, SPI has a synchronization circuit between the communication clock (RSPCK) and the operation clock (TCLK), but this synchronization circuit cannot be bypassed.

36.5 Usage Notes

36.5.1 Settings for the Module-Stop State

The Module Stop Control Register B (MSTPCRB) can enable or disable the SPI operation. The SPI is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details on the Module Stop Control Register B, see section 10, Low Power Modes.

36.5.2 Constraint on Low-Power Functions

When using the module-stop function and entering a low-power mode other than CPU Sleep mode or CPU Deep Sleep mode, set the SPCR.SPE bit to 0 before completing communication.

36.5.3 Constraints on Starting Transfer

If the ICU.IELSRn.IR flag is 1 when transfer starts, the interrupt request is internally retained, which can lead to unanticipated behavior of the ICU.IELSRn.IR flag.

To prevent this, use the following procedure to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1):

1. Confirm that transfer stopped (the SPCR.SPE bit is 0).
2. Set the associated interrupt enable bit (SPCR.SPTIE bit or SPCR.SPRIE bit) to 0.
3. Read the associated interrupt enable bit (SPCR.SPTIE bit or SPCR.SPRIE bit) and confirm that its value is 0.
4. Set the ICU.IELSRn.IR flag to 0.

36.5.4 Constraints on Mode-Fault, Underrun, Overrun, Parity Error, or Receive Data Ready Event Output

Using the mode-fault, underrun, overrun, parity error, or receive data ready event is prohibited if the SPI is in multi-master mode (when the SPCR.SPMS bit is 0, the SPCR.MSTR bit is 1, and the SPCR.MODFEN bit is 1).

36.5.5 Constraints on the SPSR.SPRF and SPSR.SPTEF Flags

If the polling flags, SPRF and SPTEF, are used, using the interrupts is prohibited, and you must set the SPCR.SPRIE and SPCR.SPTIE bits to 0. Either the interrupts or the flags can be used, but not both.

37. Octal Serial Peripheral Interface (OSPI)

This is the OSPI_B version of the OSPI peripheral module.

OSPI_B is referred to as OSPI in this chapter.

37.1 Overview

The xSPI (eXpanded Serial Peripheral Interface) protocol specifies the interface for Non-Volatile Memory Devices, which provides high data throughput, low signal count, and limited backward compatibility with legacy SPI devices. The electrical interface can deliver up to 200 Mbytes per second raw data throughput. The OSPI is compliant with JEDEC standard JESD251(Profile 1.0 and 2.0), JESD251-1 and JESD252.

JESD251 specifies two interface profiles where profile 1.0 is Octal SPI and profile 2.0 is HyperBus™ (HyperRAM™ and HyperFlash™).

OSPI supports QSPI protocol.

Table 37.1 lists the OSPI specifications, Figure 37.1 shows a block diagram, and Table 37.2 lists the I/O pins.

Table 37.1 OSPI specifications

Item	Description
Protocol	Compliant with the xSPI protocol
Data transmission and reception	Issue the transaction for up to 2 Slave as Master Only one of the memory devices can operate at a time.
Transfer speed	Support the transfer at xSPI200
Mode	• Support Protocol modes below – 1/4/8pin with SDR/DDR (1S-1S-1S, 4S-4D-4D, 8D-8D-8D) – 2/4pin with SDR (1S-2S-2S, 2S-2S-2S, 1S-4S-4S, 4S-4S-4S) • Configurable address length • Configurable initial access latency cycle • Support XiP mode
OSPI function	• Support Write Data Mask • Support In-band Reset • Memory-mapping – Support up to 256 MB address space each CS – 2 control channels per unit for the bus masters ch1: GLCDC1 bus master ch0: Other bus master – Prefetch function for burst-read with low latency – Outstanding buffer for burst-write with high throughput • Manual command – Configurable up to 4 commands – Status Register Polling function • Input Strobe port timing shift
Decryption function	Decryption on the fly is available for memory map read
Interrupt source	Error interrupt Completion interrupt
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security attribution can be set for IO register area External address space is defined as Non-secure

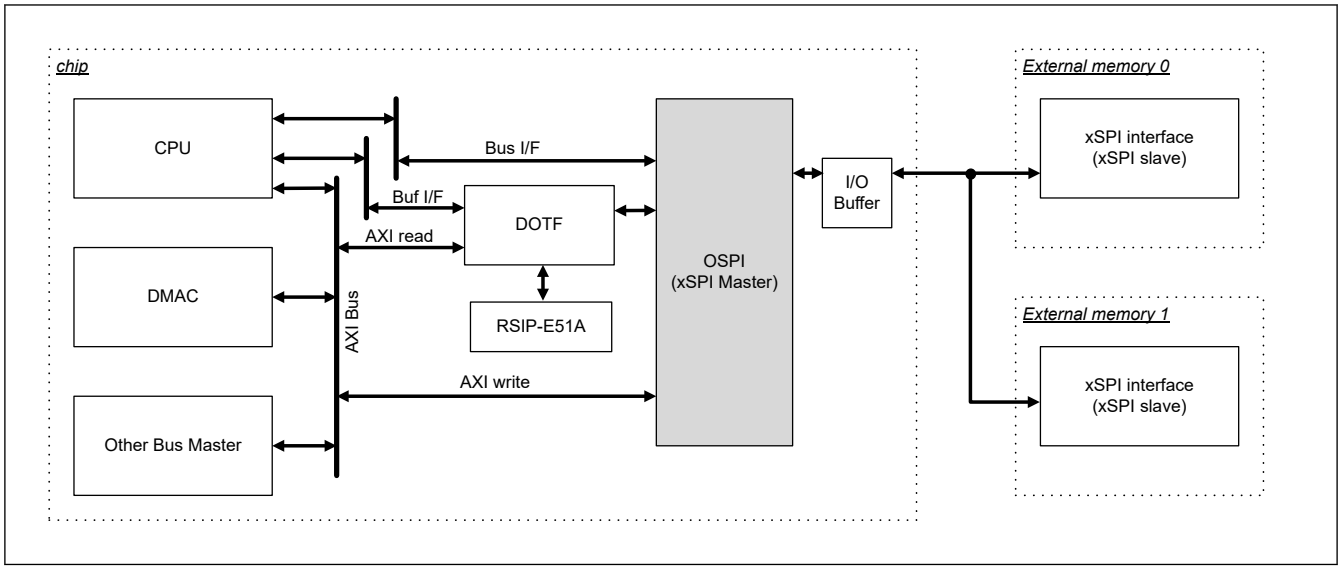


Figure 37.1 Block diagram

Table 37.2 OSPI I/O pins

Pin name	I/O	Function
OM_SCLK	Output	Clock Positive
OM_SCLKN	Output	Clock Negative
OM_CS0	Output	Chip Select for slave0
OM_CS1	Output	Chip Select for slave1
OM_DQS	I/O	Read Data Strobe / Write Data Mask
OM_SIO0	I/O	Data 0 input/output
OM_SIO1	I/O	Data 1 input/output
OM_SIO2	I/O	Data 2 input/output
OM_SIO3	I/O	Data 3 input/output
OM_SIO4	I/O	Data 4 input/output
OM_SIO5	I/O	Data 5 input/output
OM_SIO6	I/O	Data 6 input/output
OM_SIO7	I/O	Data 7 input/output
OM_RESET	Output	Master reset status for slave0,1
OM_RSTO1	Input	Slave reset status for slave1
OM_ECSINT1	Input	Interrupt for slave1 / Error Correction Status for slave1
OM_WP1	Output	Write Protect for slave1

Note: For OM_SIO7-0, OM_SCLK, OM_SCKN and OM_DQS pins, 36Ω±5% resistor need to be put on the board to comply with JESD251 I/O driver definition. It is recommended for the operation with proper signal quality.

37.2 Register Descriptions

37.2.1 OSPI Configuration Registers

These registers configure xSPI Master function. These registers should be configured in the initialization phase before issuing xSPI transaction.

37.2.1.1 WRAPCFG : OSPI Wrapper Configuration Register

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DSSFTCS1[4:0]					—	—	—	—	—	—	—	—	—	—	—	DSSFTCS0[4:0]					—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
12:8	DSSFTCS0[4:0]	OM_DQS shift for slave0 This field configures the number of delay cell for a OM_DQS port. It is used to adjust the OM_DQS sampling timing. When automatic calibration is enabled, it could be updated automatically. In this case, it shall not be written by user. 0x00: No shift 0x01: Add a delay of 1 cell ⋮ 0x1E: Add a delay of 30 cells 0x1F: Add a delay of 31 cells	R/W
23:13	—	These bits are read as 0. The write value should be 0.	R/W
28:24	DSSFTCS1[4:0]	OM_DQS shift for slave1 The function is same as one of slave0.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to configure xSPI Master function.

37.2.1.2 COMCFG : OSPI Common Configuration Register

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OENEGEX	OEASTEX
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARBMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ARBMD[1:0]	Channel arbitration mode This field selects the behavior when system bus accesses from ch0 and ch1 to slave devices occurred simultaneously. It is used only for memory-mapping mode. 0 0: Round-Robbin (ch0-ch1-ch0-ch1…)*1 0 1: Always ch0 win 1 0: Always ch1 win*1 1 1: Reserved	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	OEASTEX	Output Enable Asserting extension This bit extends 1 cycle output enable of OM_SIO7-0 and OM_DQS during output enable asserting. When set to 1, OM_CSn (n = 0, 1) asserting should be extended (LIOCFGCSn.CSASTEX = 1 (n = 0, 1)). This bit shall not be used in case of no latency cycle. Because OSPI output data could be conflicted with OSPI input data. 0: No extend 1 cycle Output enable 1: Extend 1 cycle Output enable	R/W
17	OENEGEX	Output Enable Negating extension This bit extends 1 cycle output enable of OM_SIO7-0 and OM_DQS during output enable negating. This bit should not be used in case of no latency cycle. Because OSPI output data could be conflicted with OSPI input data. 0: No extend 1 cycle Output enable 1: Extend 1 cycle Output enable	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. ch1 can only be used for GLCDC1 bus master.

This register has functions to configure xSPI Master function.

37.2.1.3 BMCFGCHn : OSPI Bridge Map Configuration Register chn (n = 0, 1)

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x008 + 0x004 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CMBTIM[7:0]								—	—	—	—	—	—	—	PREEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MWRSIZE[7:0]								MWRCOMB	—	—	—	—	—	—	WRMD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WRMD	System bus Write Response mode This bit selects the timing of System bus write response in memory-mapping mode. When set to 1, it returns the response after transmitting a frame on xSPI bus. When enabled this mode, Memory Write Combination mode must be disabled. 0: Return response after storing to Internal Write Buffer 1: Return response after issuing write transaction to xSPI bus	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	MWRCOMB	Memory Write Combination mode This bit selects to combine the OSPI data in write access of memory-mapping mode. When set to 0, OSPI data size depends on system bus's burst type and size. When this field is set to "1", the data size depends on MWRSIZE[7:0] field.*1 When set to 1, any write transaction could be held in this xSPI master temporarily. 0: Disable combination mode 1: Enable combination mode	R/W

Bit	Symbol	Function	R/W
15:8	MWRSIZE[7:0]	Memory Write Size These bits select the size to combine incremental address in memory-mapping mode. It transmits an xSPI frame with the data combined up to the configured size while the address is incremental. When detected non-incremental address or a read transaction before reaching to the target size, it transmits the pending data into xSPI bus. 0x00: Combine incremental address up to 4 bytes 0x01: Combine incremental address up to 8 bytes ⋮ 0x0E: Combine incremental address up to 60 bytes 0x0F: Combine incremental address up to 64 bytes 0xFF: Combine incremental address up to 2 bytes Others: Setting prohibited	R/W
16	PREEN	Prefetch enable This bit enables prefetch function for read transaction in memory-mapping mode. It could reduce the latency for read transaction with incremental address. 0: Disable prefetch function 1: Enable prefetch function	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
31:24	CMBTIM[7:0]	Combination timer This field specifies expiration period of combination timer. This timer is counted by PCLKA. 0x00 means disabling the combination timer. When the timer is expired, the data in the combination buffer is pushed to memory device.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Writing to OSPI memory space other than 64 bit is prohibited during combination mode.

This register has functions to configure xSPI Master function.

37.2.1.4 CMCFG0CSn : OSPI Command Map Configuration Register 0 CSn (n = 0, 1)

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x010 + 0x010 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ADDRPCD[7:0]								ADDRPEN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	ARYAMD	WPBSTMD	ADDSIZE[1:0]		FFMT[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	FFMT[1:0]	Frame format These bits configure xSPI frame format in memory-mapping mode. Please see Table 37.7 for detail. 0 0: Normal format: Command 1 byte, Address ADDSIZE, Data up to system bus transaction. 0 1: 8D-8D-8D profile 1.0 format: Command 2 bytes, Address ADDSIZE, Data up to system bus transaction 1 0: 8D-8D-8D profile 2.0 Command Modifier format: Command & Modifier 6 bytes, Data up to system bus transaction 1 1: 8D-8D-8D profile 2.0 Commands with Extended Command Modifier format: Command & Modifier 6 bytes, Data up to system bus transaction	R/W

Bit	Symbol	Function	R/W
3:2	ADDSIZE[1:0]	Address size These bits configure the number of address byte in memory-mapping mode. In case of 8D-8D-8D profile 2.0, it should be configured to 4 bytes. 0 0: 1 byte (256-byte address space) 0 1: 2 bytes (64 KB address space) 1 0: 3 bytes (16 MB address space) 1 1: 4 bytes (4 GB address space)	R/W
4	WPBSTMD	Wrapping burst mode When this field is set to 1, the wrapping boundary between system bus access and xSPI memory shall be matched. 0: Separate xSPI transfer at the wrapping address boundary 1: Not separate xSPI transfer at the wrapping address boundary	R/W
5	ARYAMD	Array address mode When this field is set to 1, address for memory is mapped as {A[25:10], A[9:4], 6{RSV}, A[3:0]} where A[25:0] is normal address, and RSV is reserved value(0b). This field is effective only when FFMT=1d. 0: Normal address mode 1: Array address mode	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W
23:16	ADDRPEN[7:0]	Address Replace Enable These bits select the bits to replace for MSByte of System bus address in memory-mapping mode. 0: No replacement (xSPI frame address field is same as System bus address) 1: Replacement	R/W
31:24	ADDRPCD[7:0]	Address Replace Code These bits configure the code to replace the MSByte of System bus address in memory-mapping mode. It replaces the corresponding bits when Address Replace Enable bit is set to 1.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to configure xSPI Master function.

37.2.1.5 CMCFG1CSn : OSPI Command Map Configuration Register 1 CSn (n = 0, 1)

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x014 + 0x010 × n

Bit position:	31											20			16	15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	RDLATE[4:0]				RDCMD[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RDCMD[15:0]	Read command These bits configure the command field of read transaction in memory-mapping mode. Normal format and 8D-8D-8D profile 2.0 format use only upper 1 byte. 8D-8D-8D profile 1.0 format uses 2 bytes.	R/W
20:16	RDLATE[4:0]	Read latency cycle These bits configure the latency cycle of read transaction in memory-mapping mode. 0x00: No latency 0x01: 1 cycle ⋮ 0x08: 8 cycle (default) ⋮ 0x1E: 30 cycles 0x1F: 31 cycles Others: Setting prohibited	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to configure xSPI Master function.

37.2.1.6 CMCFG2CSn : OSPI Command Map Configuration Register 2 CSn (n = 0, 1)

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x018 + 0x010 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	16	15	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	WRLATE[4:0]		WRCMD[15:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WRCMD[15:0]	Write command These bits configure the command field of write transaction in memory-mapping mode. Normal format and 8D-8D-8D profile 2.0 format use only upper 1 byte. 8D-8D-8D profile 1.0 format uses 2 bytes.	R/W
20:16	WRLATE[4:0]	Write latency cycle These bits configure the latency cycle of write transaction in memory-mapping mode. 0x00: No latency 0x01: 1 cycle ⋮ 0x08: 8 cycle (default) ⋮ 0x1E: 30 cycles 0x1F: 31 cycles Others: Setting prohibited	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to configure xSPI Master function.

37.2.1.7 LIOCFGCSn : OSPI Link I/O Configuration Register CSn (n = 0, 1)

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x050 + 0x004 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DDRSMPEX[3:0]				SDRSMPSFT[3:0]				SDRSMPMD	SDRDRV	CSNEGEX	CSASTEX	CSMIN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	WRMSKMD	LATEMD	PRTMD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>9:0</td><td>PRTMD[9:0]</td><td>Protocol mode
These bits configure the protocol mode and the pin to sample data inputs.
In case of using not SPI clock but Data strobe for sampling in SDR mode, it is required to set PRTMD[9] to 1.
0x000: 1S-1S-1S
0x3B2: 4S-4D-4D
0x3FF: 8D-8D-8D
0x048: 1S-2S-2S
0x049: 2S-2S-2S
0x090: 1S-4S-4S
0x092: 4S-4S-4S
Others: Setting prohibited</td><td>R/W</td></tr>
<tr><td>10</td><td>LATEMD</td><td>Latency mode
This bit selects the behavior of initial access latency phase for both direct-manual mode and memory-mapping mode. When set to 0, the latency cycle is equal to each configured cycle from transmitting address field. When set to 1, the latency cycle is incremented from the last byte-pair of Address field and is extended 2 times of each configured cycle depending on OM_DQS port.
It is used only for profile 2.0 frame format of 8D-8D-8D protocol mode with 6 bytes command/address field. Please refer to xSPI protocol for detail.
<table>
<tr><th>Value: Function</th><th>Frame format</th><th>Usage</th></tr>
<tr><td rowspan="2">0: Configurable latency</td><td>profile 2.0</td><td>The configurable latency cycle should be set as minus 1.</td></tr>
<tr><td>Others</td><td>Latency cycle increments after address field.</td></tr>
<tr><td rowspan="2">1: Variable latency</td><td>profile 2.0</td><td>Latency cycle increments from address [23:16]. And it should not be set to 1.</td></tr>
<tr><td>Others</td><td>Not support</td></tr>
</table></td><td>R/W</td></tr>
<tr><td>11</td><td>WRMSKMD</td><td>Write mask mode
This bit selects to use OM_DQS port as write data mask. It could be useful for write access of odd byte.
It is used only for 8D-8D-8D protocol mode.
0: Write mask disable
1: Write mask enable</td><td>R/W</td></tr>
<tr><td>15:12</td><td>—</td><td>These bits are read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>19:16</td><td>CSMIN[3:0]</td><td>CS minimum idle term
This bit configures the minimum cycle between xSPI frames.
0x0: 1 cycle
0x1: 2 cycles
⋮
0x7: 8 cycles (default)
⋮
0xE: 15 cycles
0xF: 16 cycles
Others: Setting prohibited</td><td>R/W</td></tr>
<tr><td>20</td><td>CSASTEX</td><td>CS asserting extension
This bit extends 1 cycle chip select pins when asserting.
0: No extension
1: Extend 1 cycle</td><td>R/W</td></tr>
<tr><td>21</td><td>CSNEGEX</td><td>CS negating extension
This bit extends 1 cycle chip select pins when negating.
0: No extension
1: Extend 1 cycle</td><td>R/W</td></tr>
<tr><td>22</td><td>SDRDRV</td><td>SDR driving timing
This bit configures the timing of data output in SDR.
This bit should not be set to 1 in case of no latency cycle. Because OSPI output data could be conflicted with OSPI input data.
0: Drive at 1/2 cycle before CK rising-edge
1: Drive at CK rising-edge</td><td>R/W</td></tr>
</table>

Bit	Symbol	Function	R/W
23	SDRSMPMD	SDR Sampling mode This bit selects the edge of sampling in SDR. In DDR, regardless of this setting, it samples OM_SIOn (n = 0 to 7) ports with both edges of OM_DQS. When set to 1, it samples at rising-edge before falling-edge. 0: Samples data input at falling-edge 1: Samples data input at rising-edge	R/W
27:24	SDRSMPSFT[3:0]	SDR Sampling window shift These bits shift the timing of CK sampling in SDR. In case of using OM_DQS in SDR, there is no influence on the behavior. In case of DDR or using OM_DQS in SDR, it should be set to 0. 0x0: Sample without delay 0x1: Sample at 1 cycle delay ⋮ 0x6: Sample at 6 cycle delay 0x7: Sample at 7 cycle delay Others: Setting prohibited	R/W
31:28	DDRSMPEX[3:0]	DDR sampling window extend These bits configure the cycle of extending the sampling window in DDR. In DDR, the input data is sampled during the expected cycle soon after latency cycle. The input data out of range is ignored. It can be configured depending on OM_DQS propagation delay. 0x0: Expand no cycle 0x1: Expand 1 cycle ⋮ 0x6: Expand 6 cycles 0x7: Expand 7 cycles Others: Setting prohibited	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to configure xSPI Master function.

37.2.2 OSPI Control Registers

These registers control xSPI Master function.

37.2.2.1 BMCTL0 : OSPI Bridge Map Control Register 0

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x060

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CH1CS1ACC [1:0]		CH1CS0ACC [1:0]		CH0CS1ACC [1:0]		CH0CS0ACC [1:0]	
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
1:0	CH0CS0ACC[1:0]	System bus ch0 to slave0 memory area access enable This field enables the access from ch0 to CS0 memory. 0 0: Read/Write disable 0 1: Read enable, Write disable 1 0: Read disable, Write enable 1 1: Read/Write enable	R/W

Bit	Symbol	Function	R/W
3:2	CH0CS1ACC[1:0]	System bus ch0 to slave1 memory area access enable This field enables the access from ch0 to CS1 memory. 0 0: Read/Write disable 0 1: Read enable, Write disable 1 0: Read disable, Write enable 1 1: Read/Write enable	R/W
5:4	CH1CS0ACC[1:0]	System bus ch1 to slave0 memory area access enable This field enables the access from ch1 to CS0 memory.*1 0 0: Read/Write disable 0 1: Read enable, Write disable 1 0: Read disable, Write enable 1 1: Read/Write enable	R/W
7:6	CH1CS1ACC[1:0]	System bus ch1 to slave1 memory area access enable This field enables the access from ch1 to CS1 memory.*1 0 0: Read/Write disable 0 1: Read enable, Write disable 1 0: Read disable, Write enable 1 1: Read/Write enable	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. ch1 can only be used for GLCDC1 bus master.

This register has functions to control xSPI Master function.

This register shall be configured in the initialization phase. When setting is needed to be changed after beginning xSPI transaction, please stop all communication, see section 37.3.7.2. Flow of Communication Stop before changing the value of BMCTL0.

37.2.2.2 BMCTL1 : OSPI Bridge Map Control Register 1

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x064

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	PBUFCLRCH1	PBUFCLRCH0	MWRPUSHCH1	MWRPUSHCH0	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	The write value should be 0.	W
8	MWRPUSHCH0	Memory Write Data Push for ch0 This field requests to push the pending data in combination mode.*1 0: No command 1: Push request	W
9	MWRPUSHCH1	Memory Write Data Push for ch1*2 The function is same as one of ch0.*1	W
10	PBUFCLRCH0	Prefetch Buffer clear for ch0 This field requests to clear the prefetch buffer when the prefetch function is enabled.*1 It should not be set during memory access (COMSTT.MEMACCCH0 = 1). 0: No command 1: Clear request	W

Bit	Symbol	Function	R/W
11	PBUFCLRCH1	Prefetch Buffer clear for ch1*2 This function is same as one of ch0.*1	W
31:12	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3
Note 1. Prefetch/Combination behavior is not defined when cycle base race condition between asserting these bits and system bus access is occurred.
Note 2. ch1 can only be used for GLCDC1 bus master

This register has functions to control xSPI Master function.

37.2.2.3 CMCTLCHn : OSPI Command Map Control register chn (n = 0, 1)

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x0068 + 0x004 × n (n = 0, 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	XIPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	XIPEXCODE[7:0]								XIPENCODE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	XIPENCODE[7:0]	XiP mode enter code These bits configure the code to enter XiP mode in memory-mapping mode.	R/W
15:8	XIPEXCODE[7:0]	XiP mode exit code These bits configure the code to exit XiP mode in memory-mapping mode.	R/W
16	XIPEN	XiP mode enable This bit enables XiP mode in memory-mapping mode. When set to 1, XiP enter code is inserted in the latency field, and the command field in next transaction is omitted. When set to 0, XiP exit code is inserted in the latency field. And it is set to 0 automatically when transmitting XiP disable pattern. It should not be used for 8D-8D-8D protocol mode profile 2.0 frame format. 0: Disable XiP mode 1: Enable XiP mode	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

37.2.2.4 CDCTL0 : OSPI Command Manual Control Register 0

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x070

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	PEREP[3:0]				—	—	—	PERITV[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	TRNUM[1:0]		CSSEL	—	PERMD	TRREQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TRREQ	Transaction request This bit requests to issue the transaction of manual-command. When set to 1, it starts the transaction. It is cleared to 0 when the transaction completed. The transaction is canceled by clearing to 0 while the transaction is ongoing. 0: No transaction 1: Request transaction	R/W
1	PERMD	Periodic mode This bit enables the periodic transaction mode. When set to 1, it repeats a transaction periodically and compares the read value with the expected value. It alternates the status polling operation for external memory. 0: Direct manual-command mode 1: Periodic manual-command mode	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CSSEL	Chip select This bit selects a target memory to issue manual-command. 0: CS0 1: CS1	R/W
5:4	TRNUM[1:0]	Transaction number These bits configure the number of transactions in normal manual-command mode. In periodic manual-command, regardless of this setting, the read data of last command is compared. 0 0: Issue 1 command (using command buffer 0) 0 1: Issue 2 commands (using command buffer 0-1) 1 0: Issue 3 commands (using command buffer 0-2) 1 1: Issue 4 commands (using command buffer 0-3)	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W
20:16	PERITV[4:0]	Periodic transaction interval These bits configure the interval of transaction in periodic manual-command mode. Too short interval compared with CPU bus cycle could result in no store into command buffer0. The interval should be longer than 4 times of CPU bus cycle. 0x00: 2 (= 2^1) cycles 0x01: 4 (= 2^2) cycles ⋮ 0x1E: 2,147,483,648 (= 2^{31}) cycles 0x1F: 4,294,967,296 (= 2^{32}) cycles	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
27:24	PERREP[3:0]	Periodic transaction repeat These bits configure the number of transaction repetitions in periodic manual-command mode. 0x0: 1 (= 2^0) time 0x1: 2 (= 2^1) times ⋮ 0xE: 16384 (= 2^{14}) times 0xF: 32768 (= 2^{15}) times	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

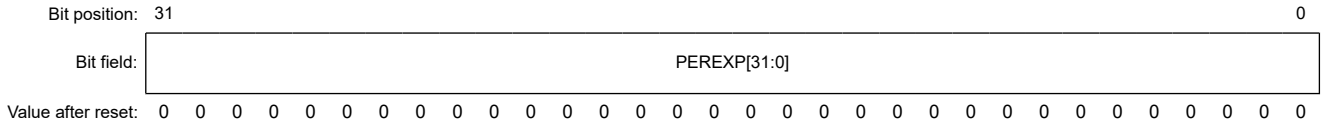
Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

37.2.2.5 CDCTL1 : OSPI Command Manual Control Register 1

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x074



Bit	Symbol	Function	R/W
31:0	PEREXP[31:0]	Periodic transaction expected value These bits configure the expected value to compare with the read value in periodic manual-command mode. For example, in case of comparing 1 byte, the lower byte should be configured.	R/W

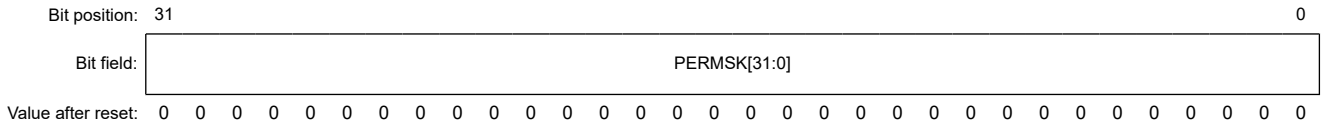
Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

37.2.2.6 CDCTL2 : OSPI Command Manual Control Register 2

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x078



Bit	Symbol	Function	R/W
31:0	PERMSK[31:0]	Periodic transaction masked value These bits configure the masked value for the expected value in periodic manual-command mode. When set 1 to any bit, the corresponding bit configured as expected value (CDCTL1.PEREXP[31:0]) is ignored. In 8D-8D-8D, the data bytes are transferred only in byte pairs on xSPI bus. It means the dummy read data could be stored. It should be masked for unused bits. For example, in case of read lower 1 byte, it should be configured to 0xFFFFFF00.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

37.2.2.7 CDTBUFn : OSPI Command Manual Type Buf n (n = 0 to 3)

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x080 + 0x010 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CMD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TRTYPE	—	LATE[4:0]					DATASIZE[3:0]				ADDSIZE[2:0]			CMDSIZE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CMDSIZE[1:0]	Command Size These bits configure the size of command field. In case of 8D-8D-8D, it should be fixed to 10b. It should not be configured both command size and address size to zero. 0 0: 0 bytes (No command phase) 0 1: 1 byte 1 0: 2 bytes Others: Setting prohibited	R/W
4:2	ADDSIZE[2:0]	Address size These bits configure the size of address field. 0 0 0: 0 bytes (No address phase) 0 0 1: 1 byte 0 1 0: 2 bytes 0 1 1: 3 bytes 1 0 0: 4 bytes Others: Setting prohibited	R/W
8:5	DATASIZE[3:0]	Write/Read Data Size These bits configure the size of data field. In 8D-8D-8D, the data bytes are transferred only in byte pairs on xSPI bus. For example, even if configuring 1 byte for read, 2 bytes data is received. The last byte should be ignored. The 0 bytes must not configured for read transaction. 0x0: 0 bytes (No data phase) 0x1: 1 byte ⋮ 0x7: 7 bytes 0x8: 8 bytes Others: Setting prohibited	R/W
13:9	LATE[4:0]	Latency cycle These bits configure the latency cycle in manual-command mode. 0x0: No latency 0x1: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
14	—	This bit is read as 0. The write value should be 0.	R/W
15	TRTYPE	Transaction Type This bit selects the type of transaction. 0: Read transaction (Readout data from slave device) 1: Not read transaction	R/W

Bit	Symbol	Function	R/W
31:16	CMD[15:0]	Command (1-2 bytes) These bits configure the command field in manual-command mode. The number of bytes configured in Command Size bit is transferred. 1S-1S-1S, 4S-4D-4D: CMD[15:8] is command field, CMD[7:0] is not used. 8D-8D-8D profile 1.0: CMD[15:8] is command field, CMD[7:0] is extension field. 8D-8D-8D profile 2.0: CMD[15:0] is upper 2 bytes of command & modifier field. (bit 47-32 in xSPI protocol)	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

37.2.2.8 CDABUFn : OSPI Command Manual Address Buf n (n = 0 to 3)

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x084 + 0x10 × n

Bit position:	31																												0
Bit field:	ADD[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	ADD[31:0]	Address These bits configure the address field in manual-command mode. 1S-1S-1S, 4S-4D-4D, 8D-8D-8D profile 1.0: It is address field. 8D-8D-8D profile 2.0: It is lower 4 bytes of command & modifier field. (bit 31-0 in xSPI protocol)	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

37.2.2.9 CDD0BUFn : OSPI Command Manual Data 0 Buf n (n = 0 to 3)

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x088 + 0x10 × n

Bit position:	31																												0
Bit field:	DATA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DATA[31:0]	Write/Read Data These bits configure the data field in manual-command mode. In case of write transaction, the write data should be configured. In case of read transaction, the read data is stored.	R/W

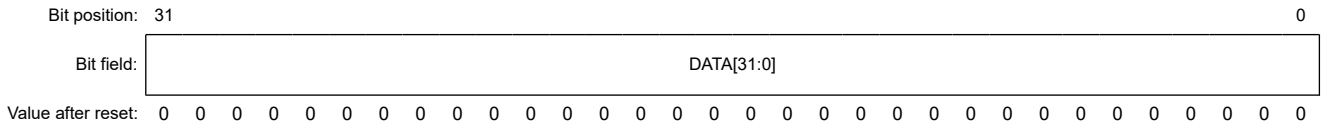
Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

37.2.2.10 CDD1BUFn : OSPI Command Manual Data 1 Buf n (n = 0 to 3)

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x08C + 0x10 × n



Bit	Symbol	Function	R/W
31:0	DATA[31:0]	Write/Read Data These bits configure the data field in manual-command mode. In case of write transaction, the write data should be configured. In case of read transaction, the read data is stored.	R/W

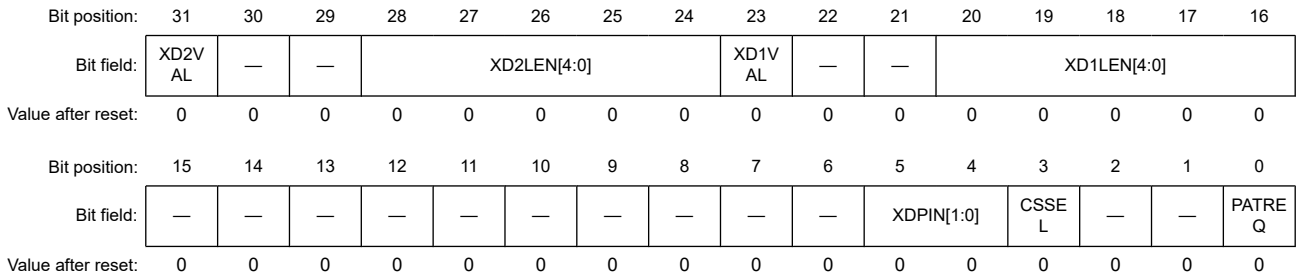
Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

37.2.2.11 LPCTL0 : OSPI Link Pattern Control Register 0

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x100



Bit	Symbol	Function	R/W
0	PATREQ	Pattern request This bit requests to issue the pattern. When set to 1, it starts the pattern. It is cleared to 0 when the pattern completed. 0: No request XiP Disable pattern 1: Request XiP Disable pattern	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	CSSEL	Chip select This bit selects a target memory to issue a pattern. 0: slave0 (CS0) 1: slave1 (CS1)	R/W
5:4	XDPIN[1:0]	XiP Disable pattern pin These bits select the data output pins to transmit XiP Disable pattern. 0 0: 1 pin 0 1: 2 pins 1 0: 4 pins 1 1: 8 pins	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
20:16	XD1LEN[4:0]	XiP Disable pattern 1st phase length These bits select the length of 1st phase in XiP disable pattern. The pattern with zero-length both 1st phase and 2nd phase should not be configured. 0x0: 0 cycles 0x1: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
22:21	—	These bits are read as 0. The write value should be 0.	R/W
23	XD1VAL	XiP Disable pattern 1st phase value This bit selects the value of 1st phase in XiP disable pattern. 0: Low drive 1: High drive	R/W
28:24	XD2LEN[4:0]	XiP Disable pattern 2nd phase length These bits select the length of 2nd phase in XiP disable pattern. 0x00: 0 cycles 0x01: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
30:29	—	These bits are read as 0. The write value should be 0.	R/W
31	XD2VAL	XiP Disable pattern 2nd phase value This bit selects the value of 2nd phase in XiP disable pattern. 0: Low drive 1: High drive	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

37.2.2.12 LPCTL1 : OSPI Link Pattern Control Register 1

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x104

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	RSTSU[2:0]			—	RSTWID[2:0]			—	—	RSTREP[1:0]		CSSEL	—	PATREQ[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PATREQ[1:0]	Pattern request These bits request to issue the pattern. When set to 01b or 10b, it starts the pattern. It is cleared to 00b when the pattern completed. 0 0: No request 0 1: Request Reset pattern 1 0: Request CS only pattern 1 1: Setting prohibited	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CSSEL	Chip select This bit selects a target memory to issue a pattern. 0: slave0 (CS0) 1: slave1 (CS1)	R/W

Bit	Symbol	Function	R/W
5:4	RSTREP[1:0]	Reset pattern repeat These bits select the repeating time to toggle CS from LOW to HIGH. 0 0: 4 times (Specified on Reset Signaling Protocol) 0 1: 5 times 1 0: 6 times 1 1: 7 times	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
10:8	RSTWID[2:0]	Reset pattern width These bits configure the width of cycle in reset pattern and CS only pattern. It toggles CS with the configured cycle. 0 0 0: 2 (= 2^1) cycles 0 0 1: 4 (= 2^2) cycles ⋮ 1 1 0: 128 (= 2^7) cycles 1 1 1: 256 (= 2^8) cycles	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
14:12	RSTSU[2:0]	Reset pattern data output setup time These bits configure the number of setup cycle for data output based on the edge of CS in reset pattern. It needs enough setup time because xSPI slave samples any data at the rising edge of CS. This cycle of setup time should be less than the cycle of reset pattern width (RSTWID[2:0]). 0 0 0: 1 cycle 0 0 1: 2 cycles ⋮ 1 1 0: 7 cycles 1 1 1: 8 cycles	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

37.2.2.13 LIOCTL : OSPI Link I/O Control Register

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x108

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RSTCS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WPCS1	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	—	This bit is read as 1. The write value should be 1.	R/W
1	WPCS1	WP drive for slave1 This bit controls the value of OM_WP1 port. It can be useful only for xSPI slave with write protect port. 0: Drive Low level 1: Drive High level	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	RSTCS0	Reset drive This bit controls the value of OM_RESET port. It could be useful only for xSPI slave with reset port. 0: Drive Low level 1: Drive High level	R/W
17	—	This bit is read as 1. The write value should be 1.	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

37.2.2.14 CCCTL0CSn : OSPI Command Calibration Control Register 0 CSn (n = 0, 1)

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x130 + 0x020 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	CASFTEND[4:0]					—	—	—	CASFTSTA[4:0]				
Value after reset:	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	CAITV[4:0]					—	—	—	—	—	—	CANOWR	CAEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CAEN	Automatic Calibration Enable This bit enables the automatic calibration. When set to 1, it transmits the calibration sequence periodically and adjusts the value of phase shift. When set to 0 during the calibration sequence, it stops after completed ongoing calibration sequence, and then this bit is cleared. 0: Disable automatic calibration 1: Enable automatic calibration	R/W
1	CANOWR	Calibration no write mode This bit selects to omit write command in calibration sequence. It can be used for any slave device with fixed calibration pattern data. 0: Calibration sequence with write command 1: Calibration sequence without write command	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
12:8	CAITV[4:0]	Calibration interval These bits configure the interval between calibration patterns. 0x00: 2 (= 2^1) cycle wait 0x01: 4 (= 2^2) cycle wait ⋮ 0x1E: 2,147,483,648 (= 2^{31}) cycle wait 0x1F: 4,294,967,296 (= 2^{32}) cycle wait	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
20:16	CASFTSTA[4:0]	Calibration OM_DQS shift start value These bits configure the start value of OM_DQS shift.	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
28:24	CASFTEND[4:0]	Calibration OM_DQS shift end value These bits configure the end value of OM_DQS shift. It should be equal or more than the start value (CASFTSTA[4:0]).	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

37.2.2.15 CCCTL1CSn : OSPI Command Calibration Control Register 1 CSn (n = 0, 1)

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x134 + 0x020 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	CARDLATE[4:0]					—	—	—	CAWRLATE[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CADATASIZE[3:0]				CAADDSIZE[2:0]			CACMDSIZE [1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CACMDSIZE[1:0]	Command Size These bits configure the size of command field. In case of 8D-8D-8D, it should be fixed to 10b. It should not be configured both command size and address size to zero. 0 0: 0 bytes (No command phase) 0 1: 1 byte 1 0: 2 bytes 1 1: Setting prohibited	R/W
4:2	CAADDSIZE[2:0]	Address size These bits configure the size of address field. 0 0 0: 0 bytes (No address phase) 0 0 1: 1 byte 0 1 0: 2 bytes 0 1 1: 3 bytes 1 0 0: 4 bytes Others: Setting prohibited	R/W
8:5	CADATASIZE[3:0]	Write/Read Data Size These bits configure the size of data field. In 8D-8D-8D, it should be configured with even byte. 0x0: 1 byte 0x1: 2 bytes ⋮ 0xE: 15 bytes 0xF: 16 bytes	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
20:16	CAWRLATE[4:0]	Write Latency cycle These bits configure the latency cycle in calibration frame. 0x00: No latency 0x01: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
28:24	CARDLATE[4:0]	Read Latency cycle These bits configure the latency cycle in calibration frame. 0x00: No latency 0x01: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

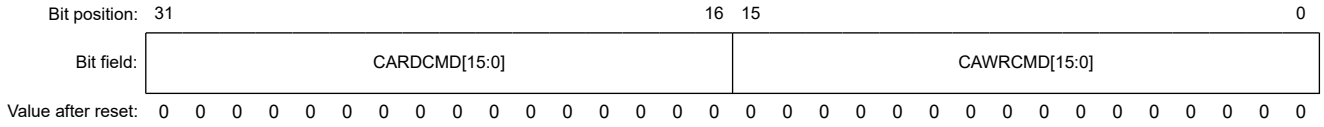
Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

37.2.2.16 CCCTL2CSn : OSPI Command Calibration Control Register 2 CSn (n = 0, 1)

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x138 + 0x020 × n



Bit	Symbol	Function	R/W
15:0	CAWRCMD[15:0]	Calibration pattern write command These bits configure the calibration pattern write command.	R/W
31:16	CARDCMD[15:0]	Calibration pattern read command These bits configure the calibration pattern read command.	R/W

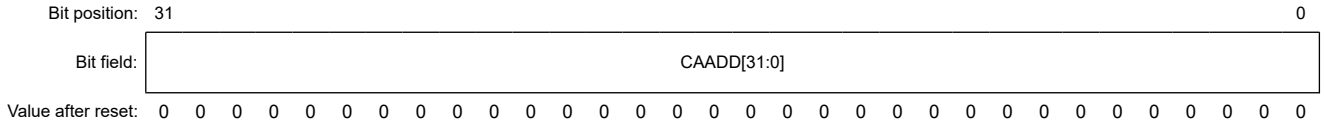
Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

37.2.2.17 CCCTL3CSn : OSPI Command Calibration Control Register 3 CSn (n = 0, 1)

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x13C + 0x020 × n



Bit	Symbol	Function	R/W
31:0	CAADD[31:0]	Calibration pattern address These bits configure the calibration pattern address.	R/W

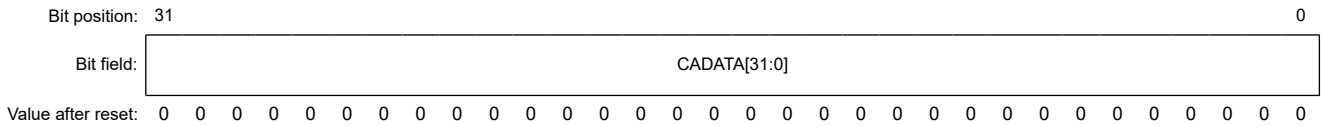
Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

37.2.2.18 CCCTL4CSn : OSPI Command Calibration Control Register 4 CSn (n = 0, 1)

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x140 + 0x020 × n



Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration pattern data These bits configure the calibration pattern data.	R/W

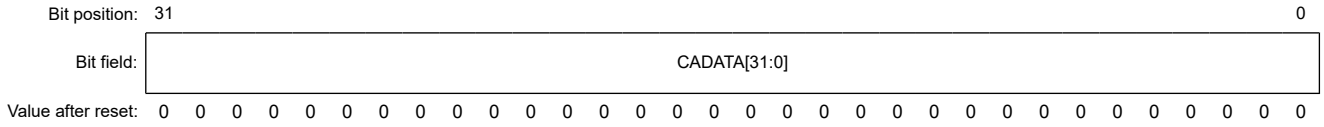
Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

37.2.2.19 CCCTL5CSn : OSPI Command Calibration Control Register 5 CSn (n = 0, 1)

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x144 + 0x020 × n



Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration pattern data These bits configure the calibration pattern data.	R/W

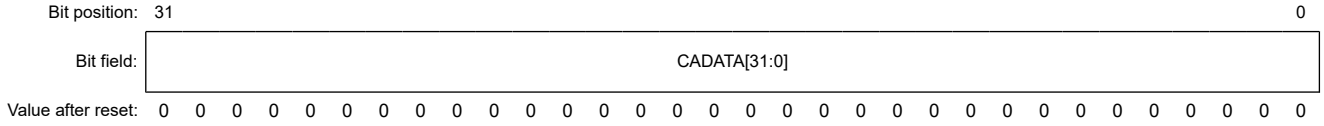
Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

37.2.2.20 CCCTL6CSn : OSPI Command Calibration Control Register 6 CSn (n = 0, 1)

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x148 + 0x020 × n



Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration pattern data These bits configure the calibration pattern data.	R/W

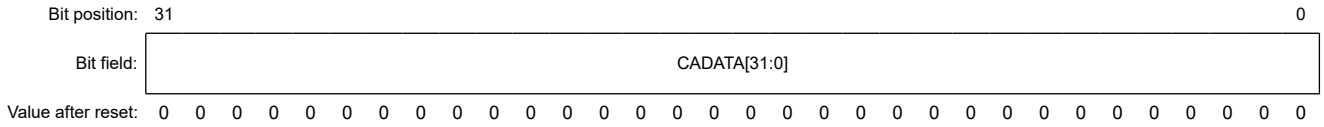
Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

37.2.2.21 CCCTL7CSn : OSPI Command Calibration Control Register 7 CSn (n = 0, 1)

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x14C + 0x020 × n



Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration pattern data These bits configure the calibration pattern data.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

37.2.3 OSPI Status Registers

These registers monitor the status of xSPI Master.

37.2.3.1 COMSTT : OSPI Common Status Register

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x184

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	RSTOCS1	INTCS1	ECSCS1	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	WRBUFNECH1	WRBUFNECH0	PBUFNECH1	PBUFNECH0	—	—	MEMACCCH1	MEMACCCH0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MEMACCCH0	Memory access ongoing from ch0 This bit is only valid in the section 37.3.7.2. Flow of Communication Stop and section 37.3.7.6. Flow of Memory-mapping Stop flows. 0: System bus bridge ch0 is not accessing to memory. 1: System bus bridge ch0 is accessing to memory.	R
1	MEMACCCH1	Memory access ongoing from ch1*1 This function is same as ch0.	R
3:2	—	These bits are read as 0.	R
4	PBUFNECH0	Prefetch Buffer Not Empty for ch0 0: Empty 1: Not empty	R
5	PBUFNECH1	Prefetch Buffer Not Empty for ch1*1 This function is same as ch0.	R
6	WRBUFNECH0	Write Buffer Not Empty for ch0 This bit is only valid in the section 37.3.7.2. Flow of Communication Stop and section 37.3.7.6. Flow of Memory-mapping Stop flows. 0: Empty 1: Not empty	R
7	WRBUFNECH1	Write Buffer Not Empty for ch1*1 This function is same as ch0.	R
19:8	—	These bits are read as 0.	R
20	ECSCS1	ECS monitor for slave1 This bit indicates the value of OM_ECSINT1 port. 0: Low level 1: High level	R
21	INTCS1	INT monitor for slave1 This bit indicates the value of OM_ECSINT1 port. 0: Low level 1: High level	R
22	RSTOCS1	RSTO monitor for slave1 This bit indicates the value of OM_RSTO1 port. 0: Low level 1: High level	R
31:23	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3
Note 1. ch1 can only be used for GLCDC1 bus master.

This register indicates the status of xSPI Master.

37.2.3.2 CASTTCSn : OSPI Calibration Status Register CSn (n = 0, 1)

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x188 + 0x004 × n

Bit position:	31																												0
Bit field:	CASUC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CASUC[31:0]	Calibration Success These bits indicate the calibration success for each OM_DQS shift value. It is updated when completed each calibration sequence. CASUC[x] indicates calibration success in OM_DQS shift value = x.	R

Note: S-TYPE-3, P-TYPE-3

This register indicates the status of xSPI Master.

37.2.4 OSPI Interrupt Registers

These registers control the interrupt function of xSPI Master.

37.2.4.1 INTS : OSPI Interrupt Status Register

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x190

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CASUCCS1	CASUCCS0	CAFAILCS1	CAFAILCS0	—	—	—	—	—	—	BUSERRCH1	BUSERRCH0	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	INTCS1	—	—	—	ECSCS1	—	—	—	DSTOCS1	DSTOCS0	PERTO	—	PATCMP	CMDCMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMDCMP	Command Completed This bit is set to 1 when completed the requested manual-command. In direct manual-command, it means all transactions completed. In periodic manual-command, it means the read data matches with the expected data. 0: No detection 1: Detection	R
1	PATCMP	Pattern Completed This bit is set to 1 when completed the requested pattern. 0: No detection 1: Detection	R
2	—	This bit is read as 0.	R
3	PERTO	Periodic transaction timeout This bit is set to 1 when the read value does not match with the expected value in periodic manual-command mode. 0: No detection 1: Detection	R

Bit	Symbol	Function	R/W
4	DSTOCS0	OM_DQS timeout for slave0 This bit is set to 1 when lost OM_DQS in read transaction with using OM_DQS. It means not receiving the data during expected read phase. In this case, xSPI master stops the read transaction and the following transaction. This error may issue error response to system bus. 0: No detection 1: Detection	R
5	DSTOCS1	OM_DQS timeout for slave1 This function is same as DSTOCS0. 0: No detection 1: Detection	R
8:6	—	These bits are read as 0.	R
9	ECSCS1	ECC error detection for slave1 This bit is set to 1 when detected the falling edge on OM_ECSINT1 port. It can be useful only for xSPI slave with ECC detection function. 0: No detection 1: Detection	R
12:10	—	These bits are read as 0.	R
13	INTCS1	Interrupt detection for slave1 This bit is set to 1 when detected the falling edge on OM_ECSINT1 port. It can be useful only for xSPI slave with interrupt function. 0: No detection 1: Detection	R
19:14	—	These bits are read as 0.	R
20	BUSERRCH0	System bus error for ch0 This bit is set to 1 when an error response occurs on system bus channel0.	R
21	BUSERRCH1	System bus error for ch1*1 This function is same as CH0.	R
27:22	—	These bits are read as 0.	R
28	CAFAILCS0	Calibration failed for slave0 This bit is set to 1 when failed calibration. 0: No detection 1: Detection	R
29	CAFAILCS1	Calibration failed for slave1 This function is same as CAFAILCS0. 0: No detection 1: Detection	R
30	CASUCCS0	Calibration success for slave0 This bit is set to 1 when success calibration. 0: No detection 1: Detection	R
31	CASUCCS1	Calibration success for slave1 This function is same as CASUCCS0. 0: No detection 1: Detection	R

Note: S-TYPE-3, P-TYPE-3
Note 1. ch1 can only be used for GLCDC1 bus master.

This register indicates the status of interrupt. The bits in this register are cleared to 0 when writing 1 on the corresponding bit of INTC register.

37.2.4.2 INTC : OSPI Interrupt Clear Register

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x194

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CASUCCS1C	CASUCCS0C	CAFAILCS1C	CAFAILCS0C	—	—	—	—	—	—	BUSERRCH1C	BUSERRCH0C	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	INTCS1C	—	—	—	ECSCS1C	—	—	—	DSTOCS1C	DSTOCS0C	PERTOC	—	PATCMPC	CMDCMPC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMDCMPC	Command Completed interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
1	PATCMPC	Pattern Completed interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
2	—	The write value should be 0.	W
3	PERTOC	Periodic transaction timeout interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
4	DSTOCS0C	OM_DQS timeout for slave0 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
5	DSTOCS1C	OM_DQS timeout for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
8:6	—	The write value should be 0.	W
9	ECSCS1C	ECC error detection for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
12:10	—	The write value should be 0.	W
13	INTCS1C	Interrupt detection for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
19:14	—	The write value should be 0.	W
20	BUSERRCH0C	System bus error for ch0 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
21	BUSERRCH1C	System bus error for ch1 interrupt clear*1 0: No change interrupt status 1: Clear interrupt status	W
27:22	—	The write value should be 0.	W
28	CAFAILCS0C	Calibration failed for slave0 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
29	CAFAILCS1C	Calibration failed for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W

Bit	Symbol	Function	R/W
30	CASUCCS0C	Calibration success for slave0 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
31	CASUCCS1C	Calibration success for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W

Note: S-TYPE-3, P-TYPE-3
Note 1. ch1 can only be used for GLCDC1 bus master.

This register clears the status of interrupt.

37.2.4.3 INTE : OSPI Interrupt Enable Register

Base address: OSPI0_B = 0x4026_8000
OSPI0_B_NS = 0x5026_8000

Offset address: 0x198

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CASUCCS1E	CASUCCS0E	CAFAILCS1E	CAFAILCS0E	—	—	—	—	—	—	BUSERRCH1E	BUSERRCH0E	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	INTCS1E	—	—	—	ECSCS1E	—	—	—	DSTOCS1E	DSTOCS0E	PERTOE	—	PATCMPE	CMDCMPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMDCMPE	Command Completed interrupt enable 0: Disabled 1: Enabled	R/W
1	PATCMPE	Pattern Completed interrupt enable 0: Disabled 1: Enabled	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	PERTOE	Periodic transaction timeout interrupt enable 0: Disabled 1: Enabled	R/W
4	DSTOCS0E	OM_DQS timeout for slave0 interrupt enable 0: Disabled 1: Enabled	R/W
5	DSTOCS1E	OM_DQS timeout for slave1 interrupt enable 0: Disabled 1: Enabled	R/W
8:6	—	These bits are read as 0. The write value should be 0.	R/W
9	ECSCS1E	ECC error detection for slave1 interrupt enable 0: Disabled 1: Enabled	R/W
12:10	—	These bits are read as 0. The write value should be 0.	R/W
13	INTCS1E	Interrupt detection for slave1 interrupt enable 0: Disabled 1: Enabled	R/W
19:14	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
20	BUSERRCH0E	System bus error for ch0 interrupt enable 0: Disabled 1: Enabled	R/W
21	BUSERRCH1E	System bus error for ch1 interrupt enable*1 0: Disabled 1: Enabled	R/W
27:22	—	These bits are read as 0. The write value should be 0.	R/W
28	CAFAILCS0E	Calibration failed for slave0 interrupt enable 0: Disabled 1: Enabled	R/W
29	CAFAILCS1E	Calibration failed for slave1 interrupt enable 0: Disabled 1: Enabled	R/W
30	CASUCCS0E	Calibration success for slave0 interrupt enable 0: Disabled 1: Enabled	R/W
31	CASUCCS1E	Calibration success for slave1 interrupt enable 0: Disabled 1: Enabled	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. ch1 can only be used for GLCDC1 bus master.

This register enables the interrupt.

37.3 Operation

xSPI Master interface has the functions to issue the transaction for external memory with xSPI Slave interface. It allows to write to registers in external memory or read from it.

This xSPI Master mainly has two modes to issue the transaction. One is a manual-command mode; Software configures all fields of xSPI frame and starts the transaction by software request. The other is a memory-mapping mode; it automatically converts system bus for pre-configured memory area into xSPI transaction. It enables to access from system bus to external memory area outside of chip via xSPI bus.

This section describes the xSPI bus operation, the direct control of xSPI frame (manual-command), the control of memory access (memory-mapping), the error operation, and the flow to operation.

37.3.1 xSPI Bus

This section describes the xSPI bus operation.

Figure 37.2 shows an example of connections between OSPI and memory devices.

In this case, OM_CS0 should be connected to the RAM device and OM_CS1 should be connected to the Flash device.

Pull-up resistors should be specified according to the instructions for each device.

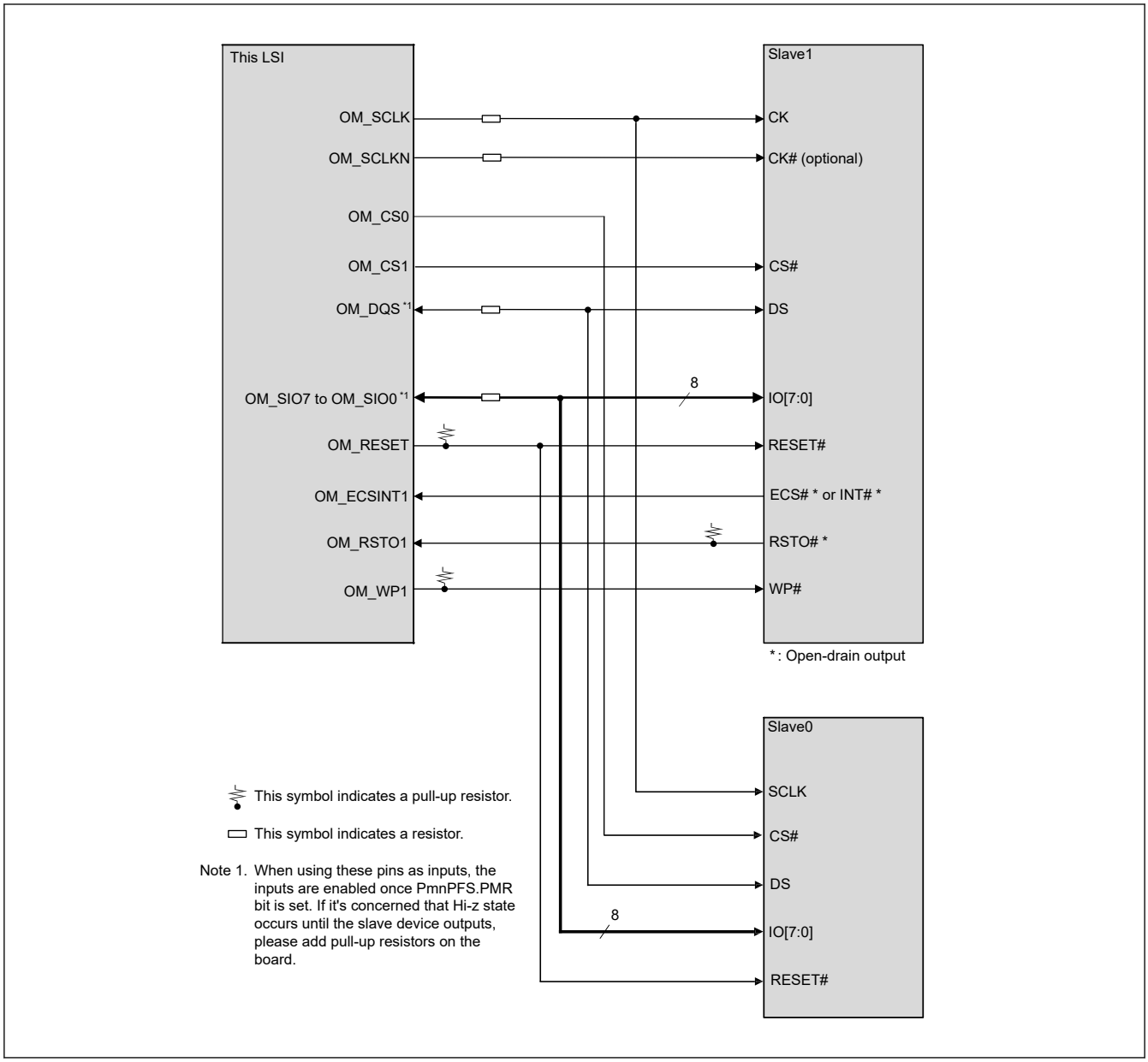


Figure 37.2 Example of Connection between this and memory devices

37.3.1.1 Supported Protocol Mode

This xSPI Master supports various protocol modes. It is configured by Protocol mode bits (LIOCFGCSn.PRTMD[9:0]). Table 37.3 shows the summary of protocol modes.

Table 37.3 Supported protocol mode (1 of 2)

Protocol mode	Function	PRTMD[9:0]	Note
1S-1S-1S	Command, Address and Data field are transferred at SDR with using 1 data input pin and 1 data output pin. Read data is sampled with CK.	0x000	Specified by xSPI protocol
4S-4D-4D	Command field is transferred at SDR with using 4 data pins. Address and Data fields are transferred at DDR with using 4 data pins. Read data is sampled with OM_DQS.	0x3B2	Specified by xSPI protocol
8D-8D-8D	Command, Address and Data fields are transferred at DDR with using 8 data pins. Read data is sampled with OM_DQS.	0x3FF	Specified by xSPI protocol
1S-2S-2S	Command field is transferred at SDR with using 1 data pin. Address and Data fields are transferred at SDR with using 2 data pins. Read data is sampled with CK.	0x048	—

Table 37.3 Supported protocol mode (2 of 2)

Protocol mode	Function	PRTMD[9:0]	Note
2S-2S-2S	Command, Address and Data fields are transferred at SDR with using 2 data pins. Read data is sampled with CK.	0x049	—
1S-4S-4S	Command field is transferred at SDR with using 1 data pin. Address and Data fields are transferred at SDR with using 4 data pins. Read data is sampled with CK.	0x090	—
4S-4S-4S	Command, Address, and Data fields are transferred at SDR with using 4 data pins. Read data is sampled with CK.	0x092	—

Note: In case of XiP mode enable, XiP code is inserted in Latency field. It is valid only for memory-mapping mode.

The bytes of Command and Address fields are transferred in highest order to lowest order sequence. The sequential bytes of Data field are transferred in lowest address to highest address order. In case of using multiple pins, the least significant bit of each byte is placed on OM_SIO0 with each higher order bit on the successively higher numbered OM_SIOn (n = 1 to 7) signals.

Figure 37.3 shows timing-chart for 1S-1S-1S protocol mode. The OM_SIO0 signal is used for output data and the OM_SIO1 signal is used for input data.

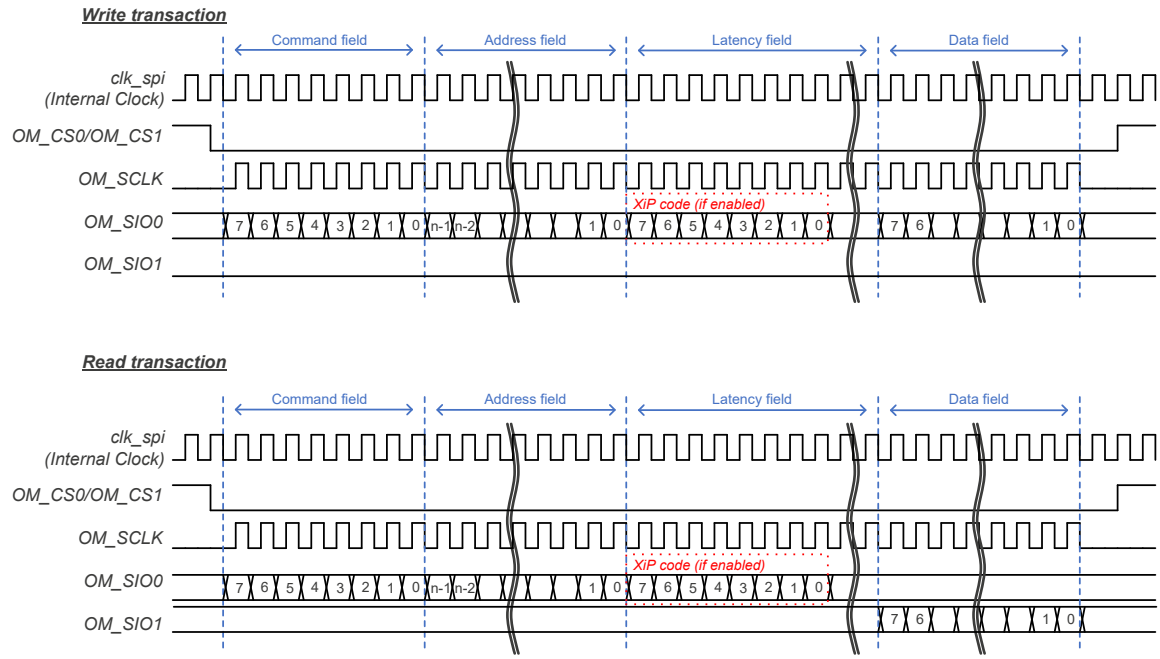


Figure 37.3 1S-1S-1S timing-chart

Figure 37.4 shows timing-chart for 1S-2S-2S protocol mode.

Figure 37.4 1S-2S-2S timing-chart

Figure 37.5 shows timing-chart for 4S-4D-4D protocol mode.

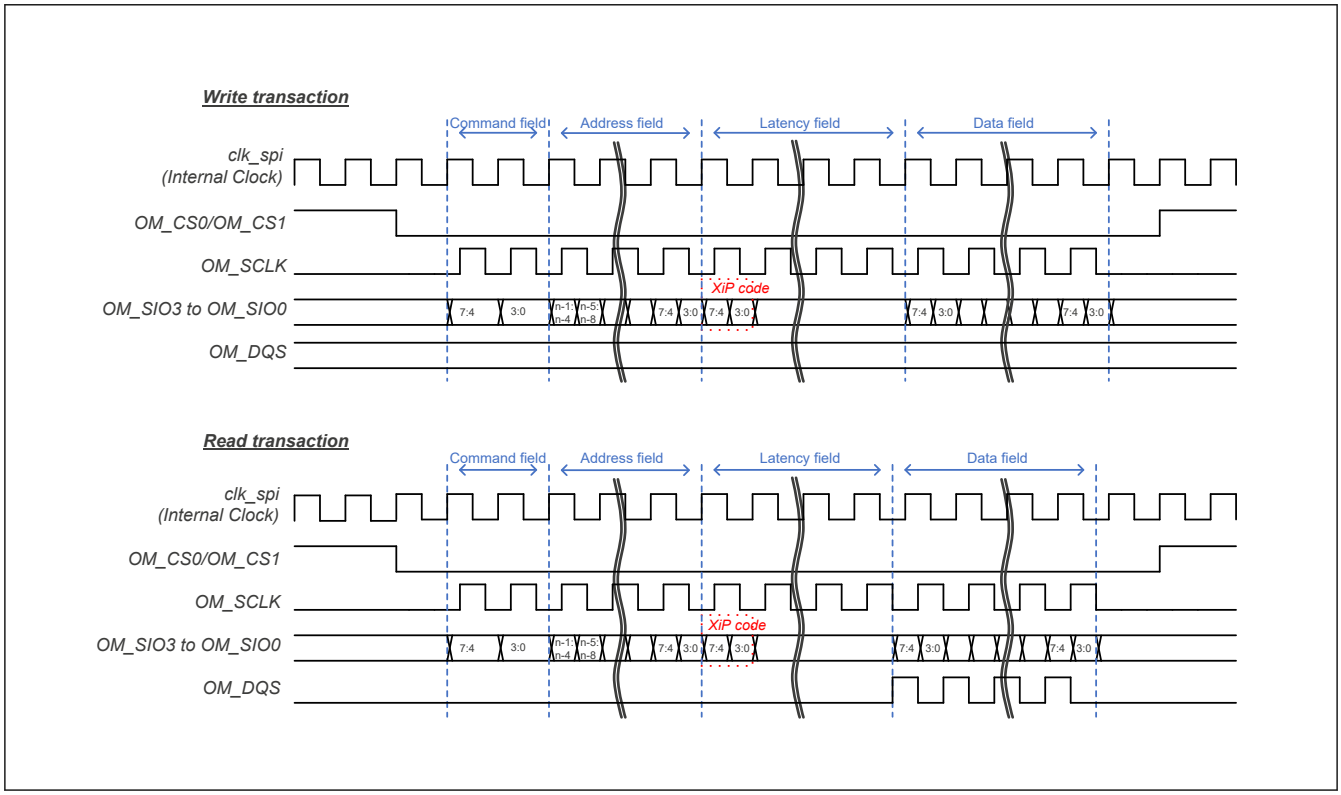


Figure 37.5 4S-4D-4D timing-chart

Figure 37.6 shows timing-chart for 8D-8D-8D profile 1.0 protocol mode. In 8D-8D-8D, the data is always transmitted with byte-pair. In case each field is odd byte, last one byte is padded with invalid data.

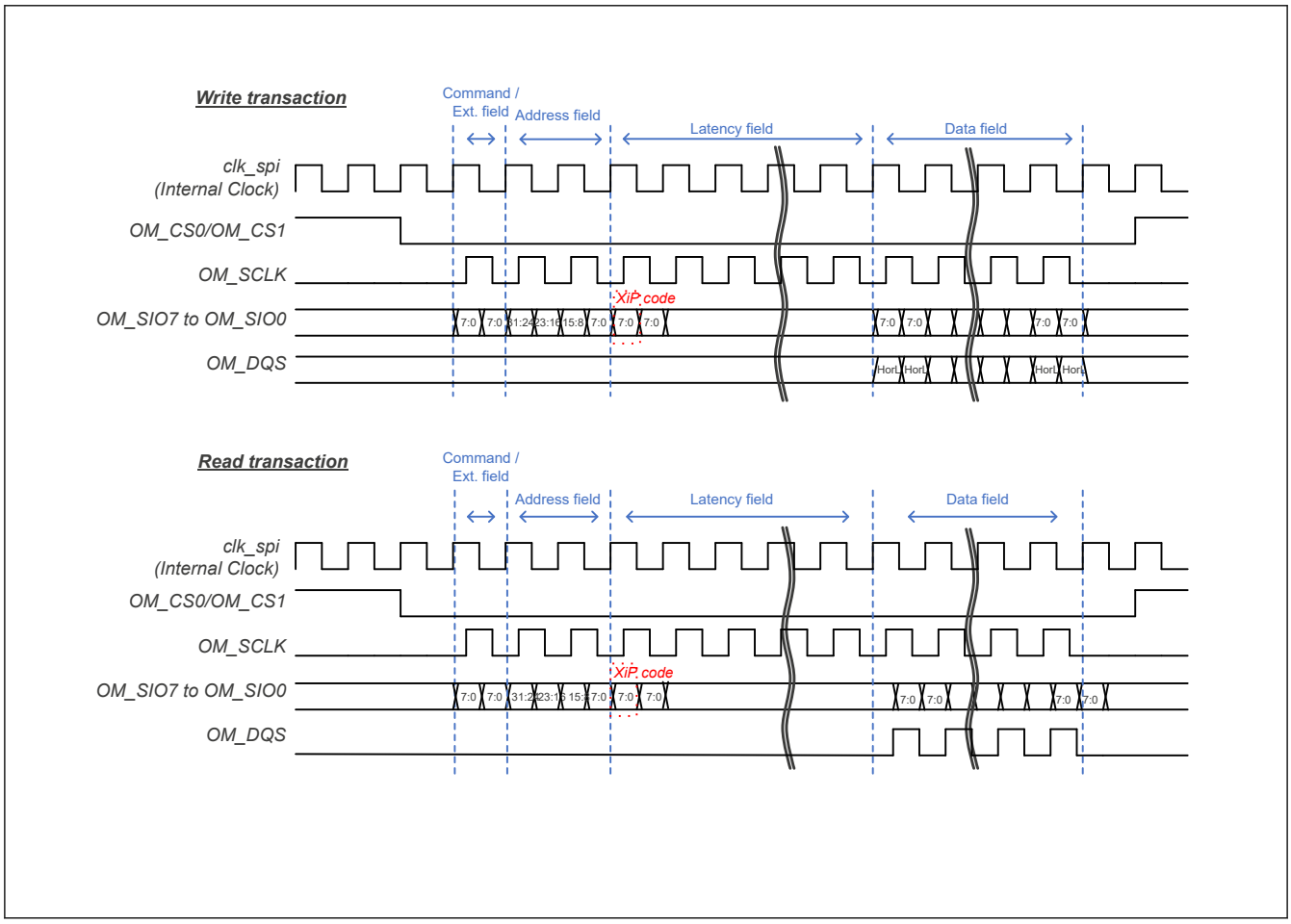


Figure 37.6 8D-8D-8D profile 1.0 timing-chart

Figure 37.7 shows timing-chart for 8D-8D-8D profile 2.0 protocol mode. In 8D-8D-8D, the data is always transmitted with byte-pair. In case each field is odd byte, last one byte is padded with invalid data.

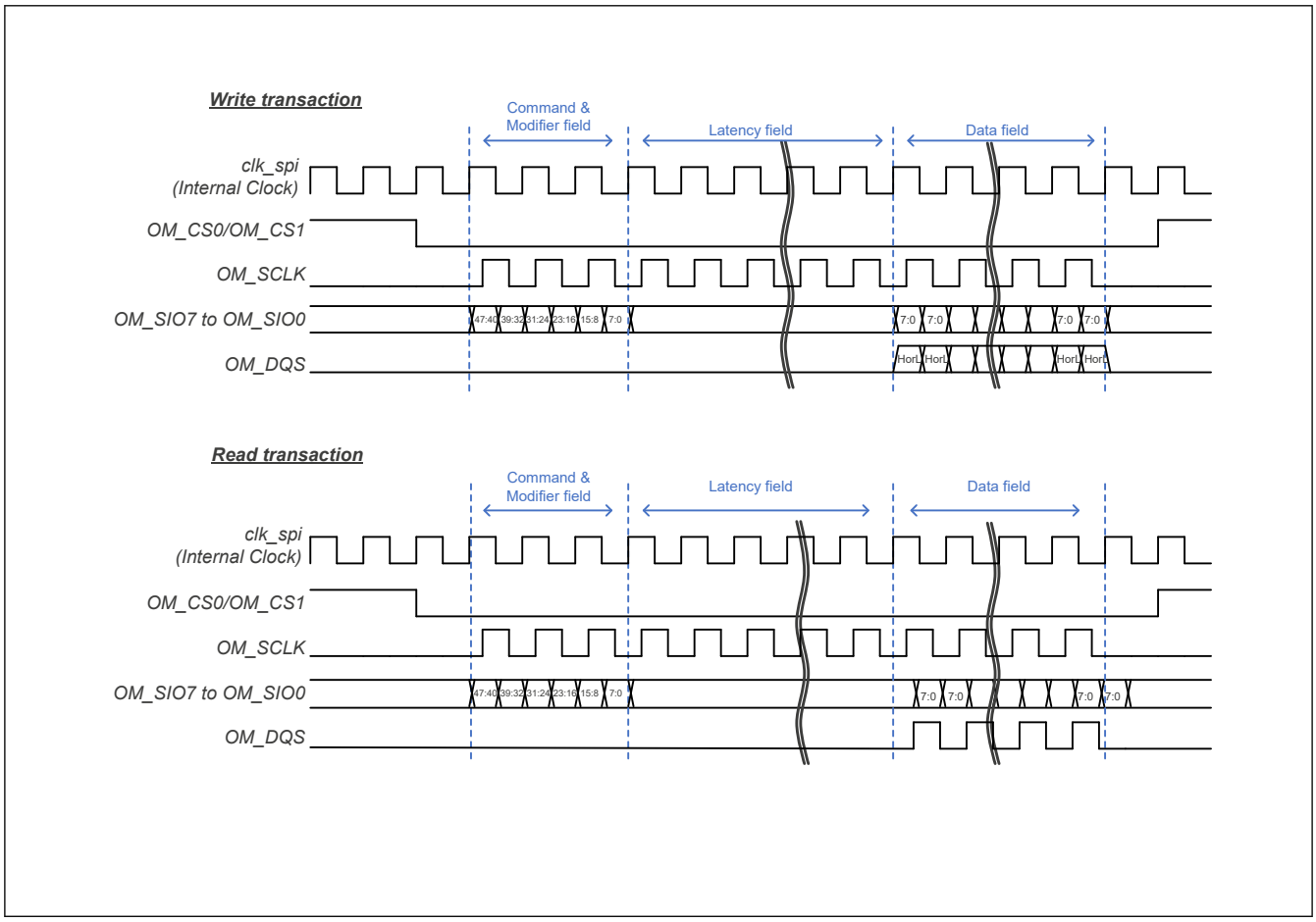


Figure 37.7 8D-8D-8D profile 2.0 timing-chart

37.3.1.2 xSPI Frame Interval

The interval between xSPI frames is configured with CS minimum idle term bits (LIOCFGCSn.CSMIN[3:0]). It depends on the specification of xSPI slave device.

37.3.1.3 OSPI Signals Timing Control

This xSPI Master supports both SDR and DDR. It is possible to sample input data with Data-Strobe (OM_DQS) signal at SDR. For various modes and easy implementation, this xSPI Master could adjust the timing to drive/sample xSPI interface signals statically. Table 37.4 shows the summary of xSPI Interface signal timing control.

Table 37.4 Summary of OSPI signals timing control (1 of 2)

Signal	Mode	Default operation	Timing control (n = 0, 1)
OM_CS0, OM_CS1 drive	—	Asserting 1 cycle before the rising-edge of first OM_SCLK	1 cycle extension for asserting with LIOCFGCSn.CSASTEX bit
		Negating 1.5 cycle after the falling-edge of last OM_SCLK	1 cycle extension for negating with LIOCFGCSn.CSNEGEX bit
OM_SCLK drive	SDR without OM_DQS SDR with OM_DQS	Reference point	—
	DDR with OM_DQS	Reference point	—

Table 37.4 Summary of OSPI signals timing control (2 of 2)

Signal	Mode	Default operation	Timing control (n = 0, 1)
OM_SIOn (n = 0 to 7) output drive	SDR without OM_DQS SDR with OM_DQS	Falling edge of clk_spi (Internal Clock)	0 or 0.5 cycle shift with LIOCFGCSn.SDRDRV bit
	DDR with OM_DQS	Both edges of clk_spi (Internal Clock)	—
OM_SIOn (n = 0 to 7) input sample	SDR without OM_DQS	Falling edge of OM_SCLK on expected data size	0 to 7 cycle shift (1 cycle unit) with LIOCFGCSn.SDRSMPSFT[3:0] bits 0 or 0.5 cycle shift with LIOCFGCSn.SDRSMPMD bit
	SDR with OM_DQS	Falling edge of OM_DQS signal on expected data size	Sample at rising edge with LIOCFGCSn.SDRSMPMD bit 0 to 1 cycle phase shift with WRAPCFG.DSSFTCSn[4:0] bits 0 to 7 cycle extension with LIOCFGCSn.DDRSMPEX[3:0] bits
	DDR with OM_DQS	Both edges of OM_DQS signal on expected data size	0 to 1 cycle phase shift with WRAPCFG.DSSFTCSn[4:0] bits 0 to 7 cycle extension with LIOCFGCSn.DDRSMPEX[3:0] bits

Note: In DDR on xSPI protocol, OM_DQS should be aligned for center of data. It means to shift the phase by 0.25 cycle (90 degrees). This xSPI master supports to adjust this phase depending on the usage conditions.

Figure 37.8 shows the default operation and timing control for SDR without OM_DQS.

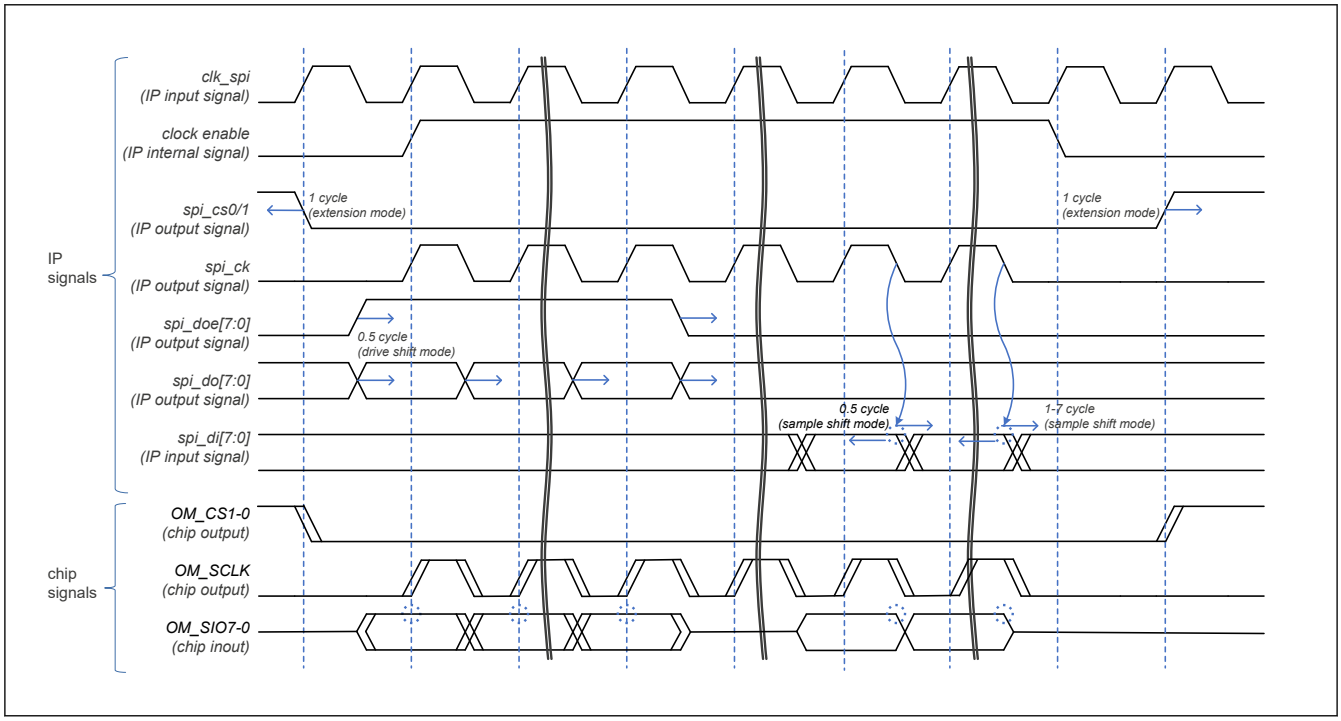


Figure 37.8 Timing control for SDR without OM_DQS

Figure 37.9 shows the default operation and timing control for SDR with OM_DQS.

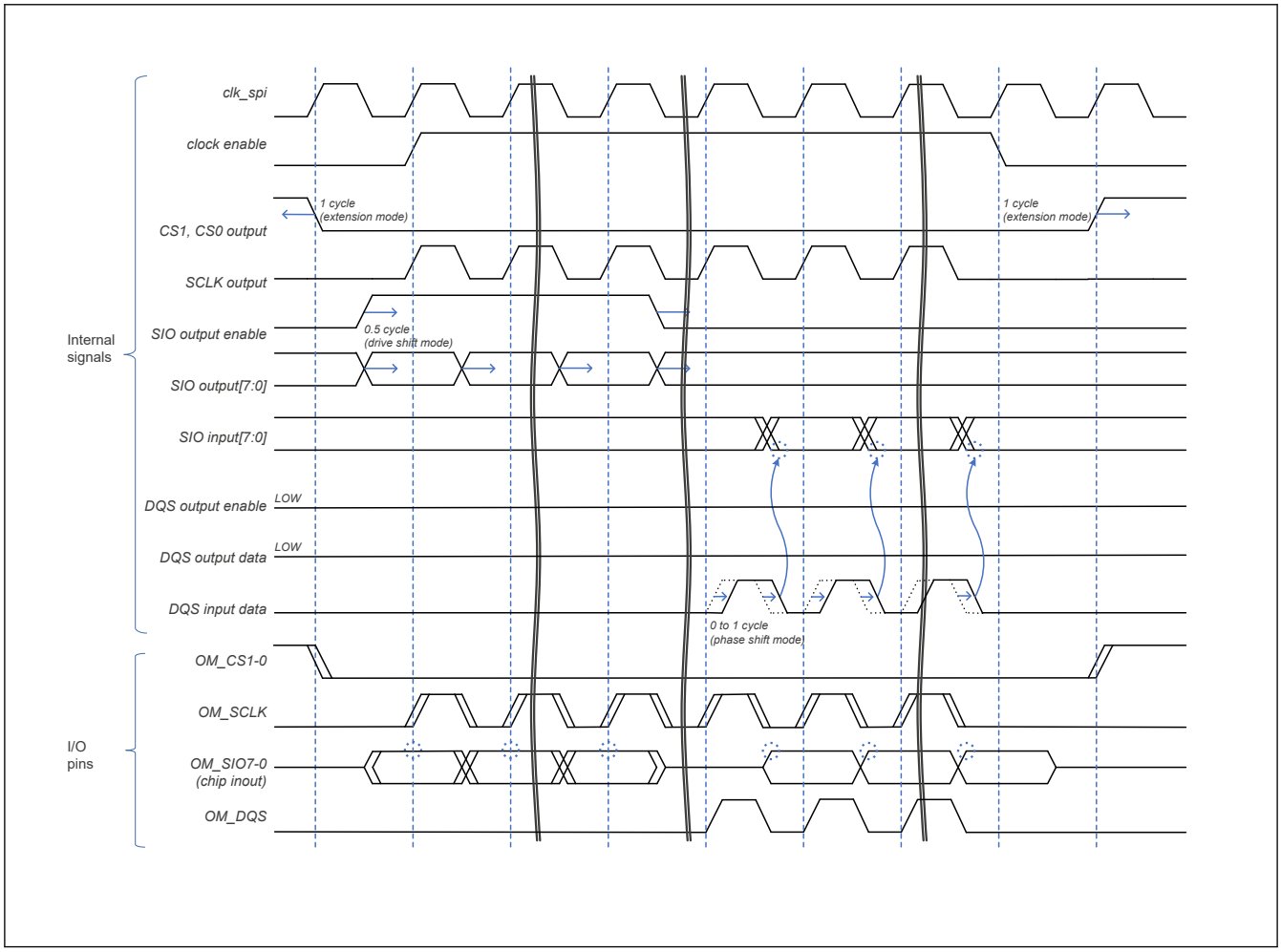


Figure 37.9 Timing control for SDR with OM_DQS

Figure 37.10 shows the default operation and timing control for DDR with OM_DQS.

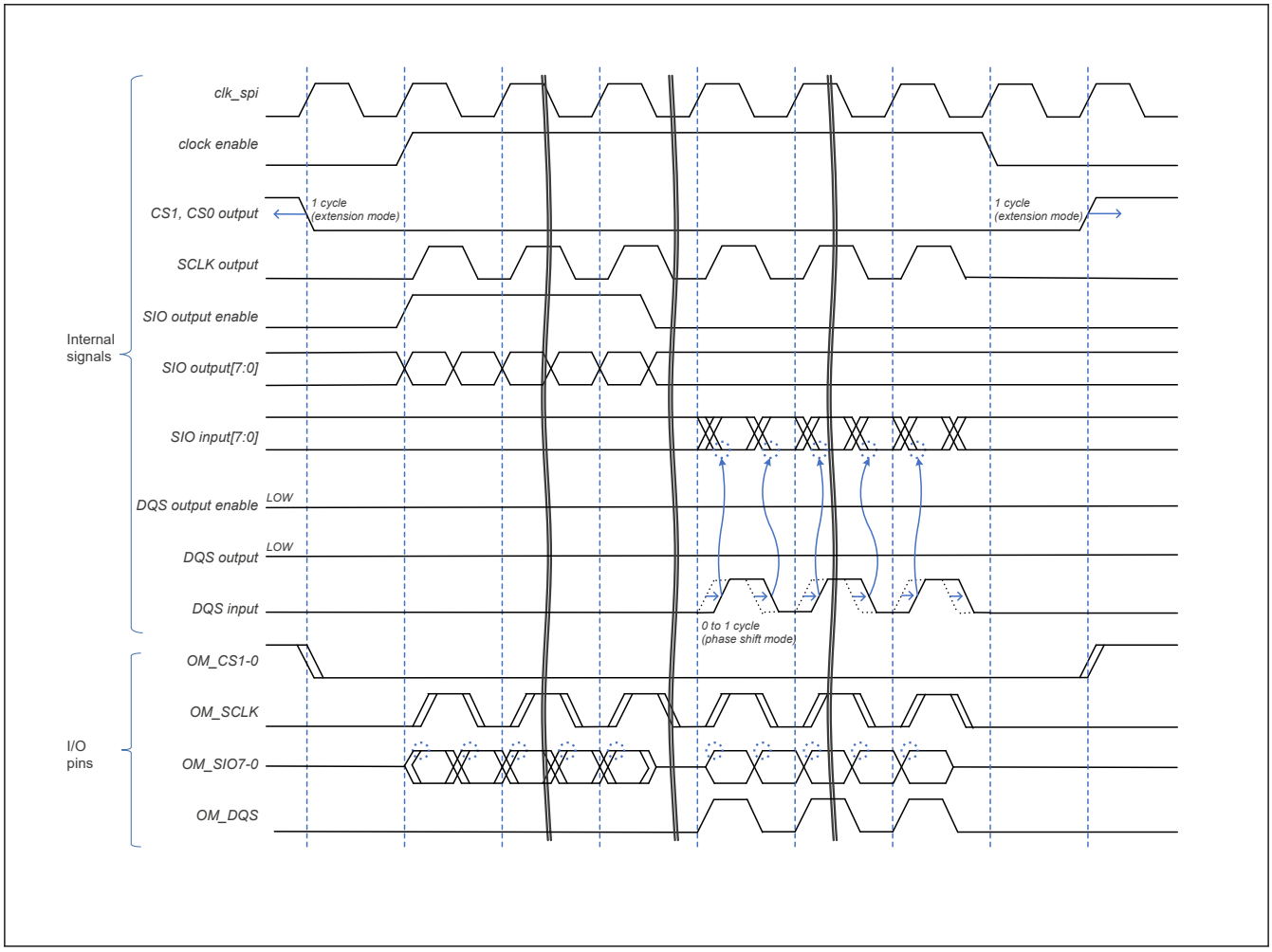


Figure 37.10 Timing control for DDR with OM_DQS

37.3.1.4 Automatic calibration

This xSPI master supports the function to adjust OM_DQS shift value (WRAPCFG.DSSFTCSn (n = 0, 1)) automatically. When this function is enabled (CCCTL0CSn.CAEN = 1), this xSPI master transmits the calibration sequence periodically and adjusts the value of phase shift.

When all read compare is mismatched in read transaction during the calibration sequence, the calibration fail bit (INTS.CAFAILCSn = 1 (n = 0, 1)) is asserted and OM_DQS shift value is not updated. When at least one read compare is matched, the calibration success bit (INTS.CASUCCSn = 1 (n = 0, 1)) is asserted and OM_DQS shift value is updated. The result of each OM_DQS shift value is monitored by Calibration Status register (CASTTCSn). Figure 37.11 shows automatic calibration.

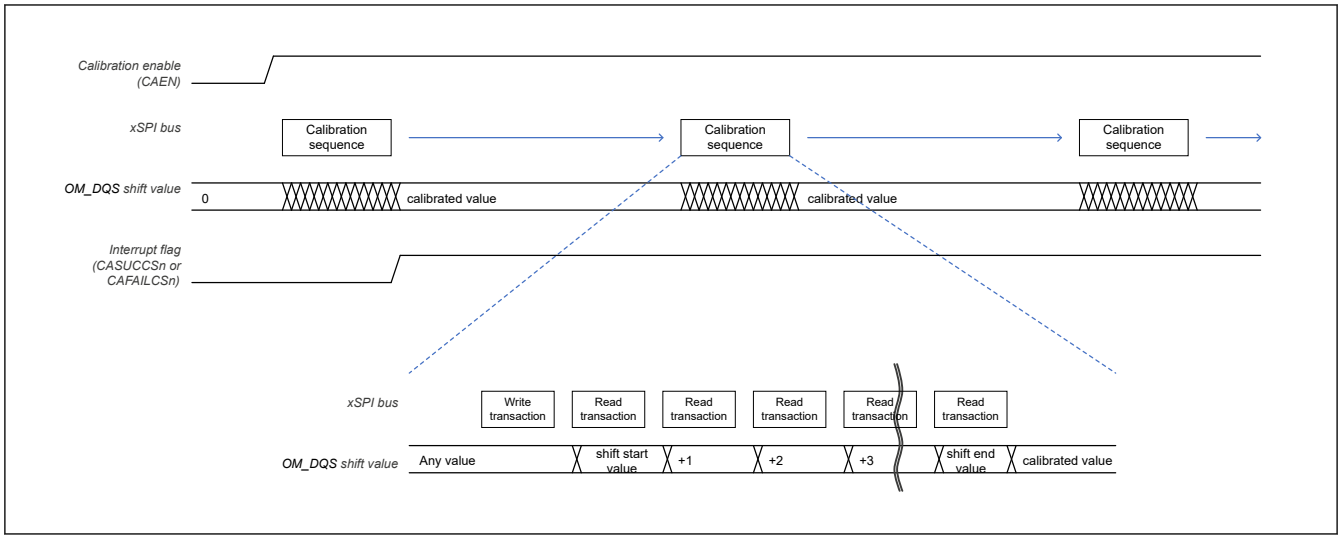


Figure 37.11 Automatic calibration

37.3.2 Manual-command

This section describes the manual-command mode. The manual-command has two functional modes: direct mode and periodic mode.

37.3.2.1 Direct Mode

This mode sequentially can issue up to four xSPI transactions configured and requested by Software. A series of transaction can be issued by a transaction request (CDCTL0.TRREQ = 1 with PERMD = 0). The number of transactions (CDCTL0.TRNUM[1:0]) can be configured up to 4. It can be used to change the mode or read the status of xSPI slave device.

Table 37.5 shows the configured register bits for direct manual-command. The operating flow is illustrated in Figure 37.25.

Table 37.5 Manual-command configuration for direct mode

Transaction	Transaction type	Command	Command size	Address	Address size	Data (x = 0, 1)	Data size	Latency cycle
1st Transaction	CDTBUF0. TRTYPE	CDTBUF0. CMD[15:0]	CDTBUF0. CMDSIZE [1:0]	CDABUF0. ADD[31:0]	CDTBUF0. ADDSIZE [2:0]	CDDxBUF0. DATA[31:0]	CDTBUF0. DATASIZE [3:0]	CDTBUF0. LATE[4:0]
2nd Transaction	CDTBUF1. TRTYPE	CDTBUF1. CMD[15:0]	CDTBUF1. CMDSIZE [1:0]	CDABUF1. ADD[31:0]	CDTBUF1. ADDSIZE [2:0]	CDDxBUF1. DATA[31:0]	CDTBUF1. DATASIZE [3:0]	CDTBUF1. LATE[4:0]
3rd Transaction	CDTBUF2. TRTYPE	CDTBUF2. CMD[15:0]	CDTBUF2. CMDSIZE [1:0]	CDABUF2. ADD[31:0]	CDTBUF2. ADDSIZE [2:0]	CDDxBUF2. DATA[31:0]	CDTBUF2. DATASIZE [3:0]	CDTBUF2. LATE[4:0]
4th Transaction	CDTBUF3. TRTYPE	CDTBUF3. CMD[15:0]	CDTBUF3. CMDSIZE [1:0]	CDABUF3. ADD[31:0]	CDTBUF3. ADDSIZE [2:0]	CDDxBUF3. DATA[31:0]	CDTBUF3. DATASIZE [3:0]	CDTBUF3. LATE[4:0]

37.3.2.2 Periodic Mode

This mode periodically issues an xSPI read transaction configured and requested by Software. And it can compare the read value up to 4 bytes with expected value. The transaction is issued by a transaction request (CDCTL0.TRREQ = 1 with PERMD = 1). It can be used to alternate the status polling operation of xSPI slave device.

The periodic term is configured in Periodic transaction interval bits (CDCTL0.PERITV[4:0]). The expected value is configured in Periodic transaction expected and masked value bits (CDCTL1.PEREXP[31:0] and CDCTL2.PERMSK[31:0]). Table 37.6 shows the configured register bits for periodic manual-command. The operating flow is illustrated in Figure 37.26.

Table 37.6 Manual-command configuration for periodic mode

Transaction	Transaction type	Command	Command size	Address	Address size	Data (x = 0, 1)	Data size	Latency cycle
Read Transaction	CDTBUF0. TRTYPE	CDTBUF0. CMD[15:0]	CDTBUF0. CMDSIZE [1:0]	CDABUF0. ADD[31:0]	CDTBUF0. ADDSIZE [2:0]	CDDxBUF0. DATA[31:0]	CDTBUF0. DATASIZE [3:0]	CDTBUF0. LATE[4:0]

37.3.3 Memory-mapping

This section describes the memory-mapping mode. This mode automatically converts system bus access for pre-configured memory area into xSPI transaction.

37.3.3.1 Configuration

In this operation, the payload of address and data field are delivered from system bus signals. The information of command field and size are delivered from the configured register bits. Table 37.7 shows the register bits configured for memory-mapping.

Table 37.7 Memory-mapping configuration for memory area access (n = 0, 1)

System bus Transaction	Format Change mode	Command	Command size	Address	Address size	Data	Data size	Latency cycle
Write for slave n memory area	Normal	CMCFG2CSn. WRCMD[15:8]	1 byte	SAWADDR[x:0]	CMCFG0CSn. ADDSIZE [1:0]	SWDATA	Up to SAWLEN and SAWSIZE	CMCFG2CSn. WRLATE[4:0]
	8D-8D-8D profile 1.0	CMCFG2CSn. WRCMD[15:0]	2 bytes					
	8D-8D-8D profile 2.0 Command Modifier	CMCFG2CSn. WRCMD[15:8]	1 byte	{SAWADDR[27:4], 0000000000000b, SAWADDR[3:1]}	5 bytes			
	8D-8D-8D profile 2.0 Extended Command Modifier	CMCFG2CSn. WRCMD[15:13]	3 bits	{0b, SAWADDR [31:4], 0000000000000b, SAWADDR[3:1]	45 bits			
Read for slave n memory area	Normal	CMCFG1CSn. RDCMD[15:8]	1 byte	SARADDR[x:0]	CMCFG0CSn. ADDSIZE [1:0]	SRDATA	Up to SARLEN and SARSIZE	CMCFG1CSn. RDLATE[4:0]
	8D-8D-8D profile 1.0	CMCFG1CSn. RDCMD[15:0]	2 bytes					
	8D-8D-8D profile 2.0 Command Modifier	CMCFG1CSn. RDCMD[15:8]	1 byte	{SARADDR[27:4], 0000000000000b, SARADDR[3:1]}	5 bytes			
	8D-8D-8D profile 2.0 Extended Command Modifier	CMCFG1CSn. RDCMD[15:13]	3 bits	{0b, SARADDR [31:4], 0000000000000b, SARADDR[3:1]}	45 bits			

Note: The MSByte of Address can be replaced with Address Replace Enable and Code bits (CMCFG0CSn.ADDRPEN[7:0] / ADDRPCD[7:0]).

37.3.3.2 Write Access Operation

At accepting write access for memory area from system bus, this xSPI Master stores all payload data in internal bridge buffer and then issues a write transaction into xSPI slave. Figure 37.12 shows the operation summary.

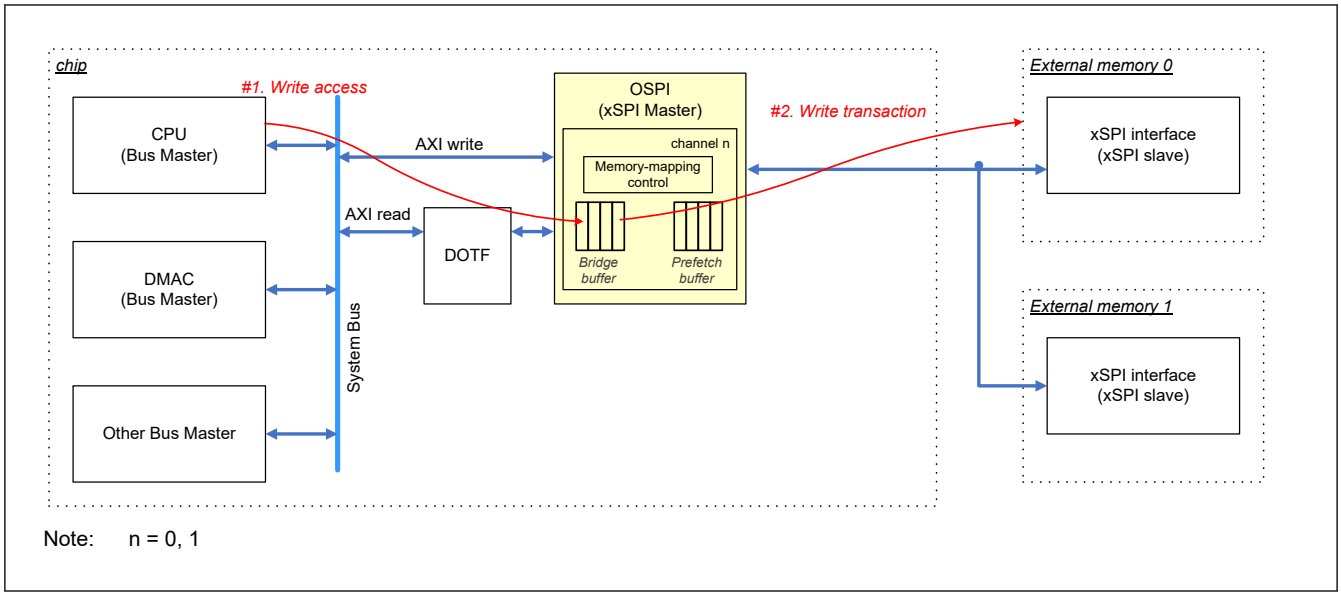


Figure 37.12 Write access for memory area

The operation of xSPI bus changes depending on system bus's burst type. When the burst type is single type or incremental type, one system bus transaction triggers for one xSPI frame. When the burst type is wrap type and the CMCFG0CSn.WPBSTMD is 0, one system bus transaction triggers two xSPI frames. Figure 37.13 shows the relationship between AXI and xSPI frames.

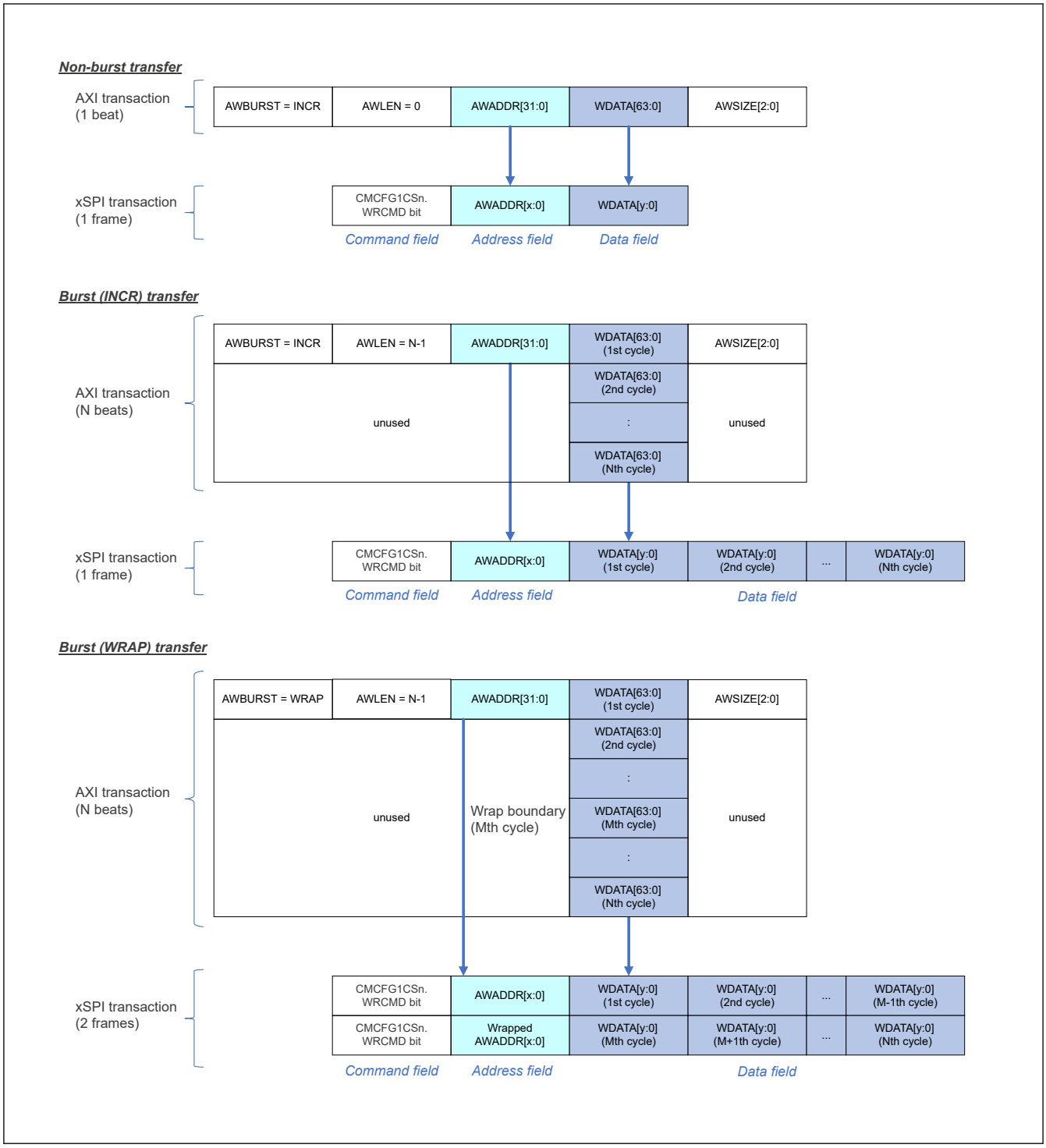


Figure 37.13 xSPI frame format in write access (Normal format)

37.3.3.3 Combination Function

At the system bus write access for memory area, this xSPI master has the function to combinate the write data for high throughput on xSPI bus. When this function is enabled (BMCFGCHn.MWRCOMB = 1), this xSPI master transmits a xSPI frame with the selected size while the sequential address is incremental[*1]. When one of the below conditions is detected, even though not reaching to the target size (BMCFGCHn.MWRSIZE[7:0]), this xSPI master transmits the pending data into xSPI bus.

- Non-incremental address is detected.
- Different burst type is detected.

- Read transaction is detected.
- Access for different slave is detected.
- Memory Write Data Push bit (BMCTL1.MWRPUSHCHn (n = 0, 1)) is set.

This function could be useful for any slave device to request a chunk of data at a time. In the case, system bus master shall continue to provide the fixed data size with incremental address. e.g. there is any device to request to write in page unit.

Figure 37.14 shows the operation when enable the combination function.

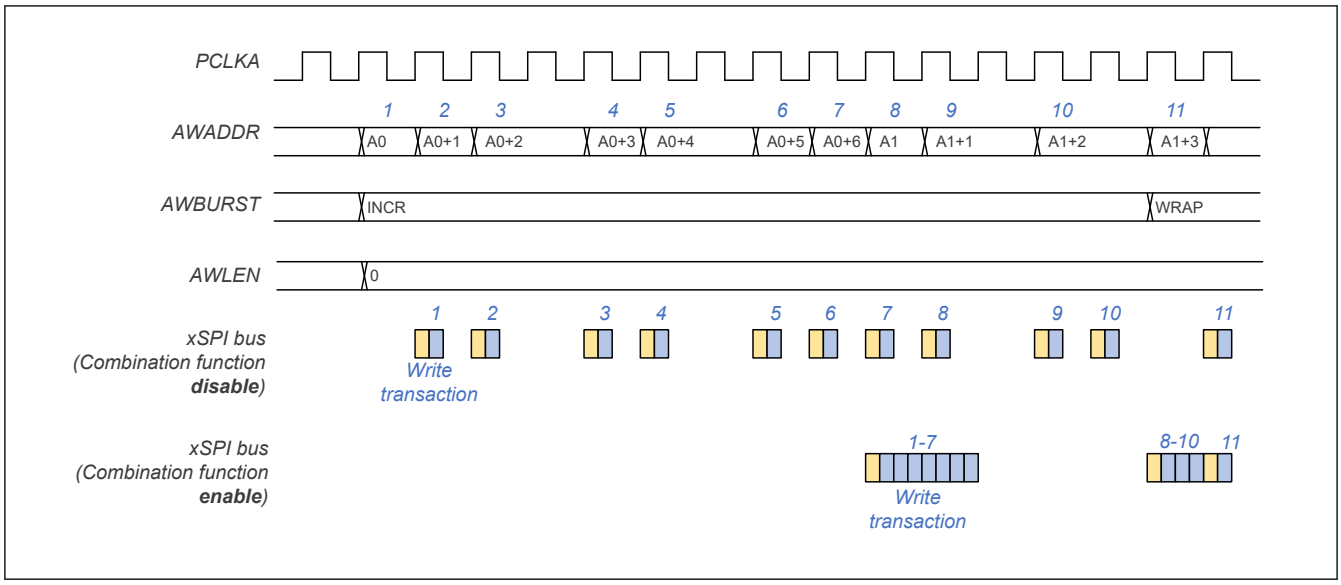


Figure 37.14 Combination function

Note 1. The access which comply all below condition is considered as incremental address.

- Transaction type is INCR.
- The access's start address is continuous to the previous last write address.
  - The access's start position of WSTRB is treated as start address.
  - Previous access's last WSTRB is treated last write address.

Figure 37.15 shows data combined example with AXI access.

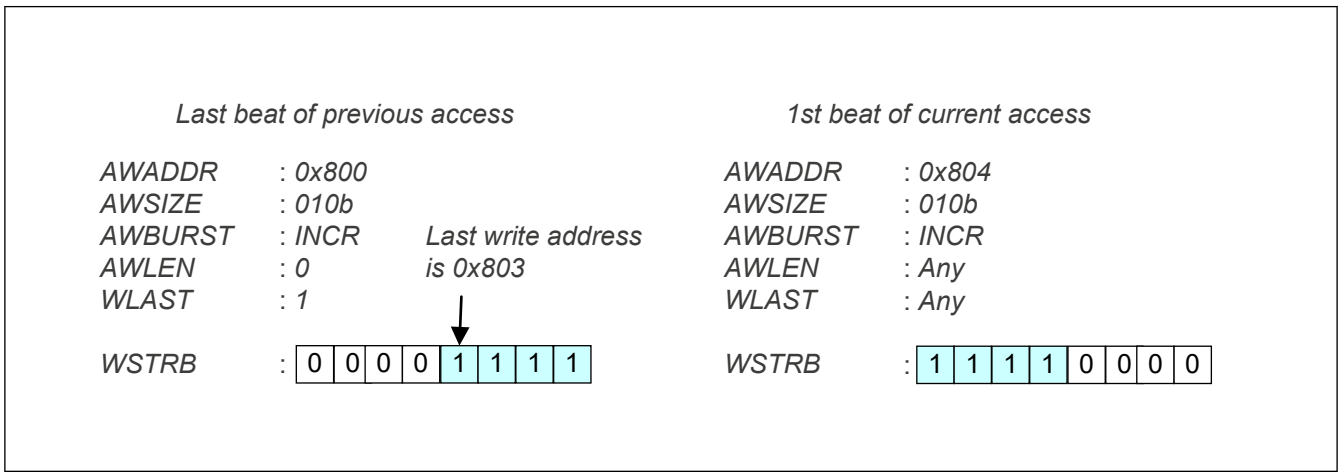


Figure 37.15 Data combined example with AXI access

37.3.3.4 Read Access Operation

At the read access for memory area, soon after detected the read access, this xSPI Master issues a read transaction into xSPI slave. Figure 37.16 shows the operation summary.

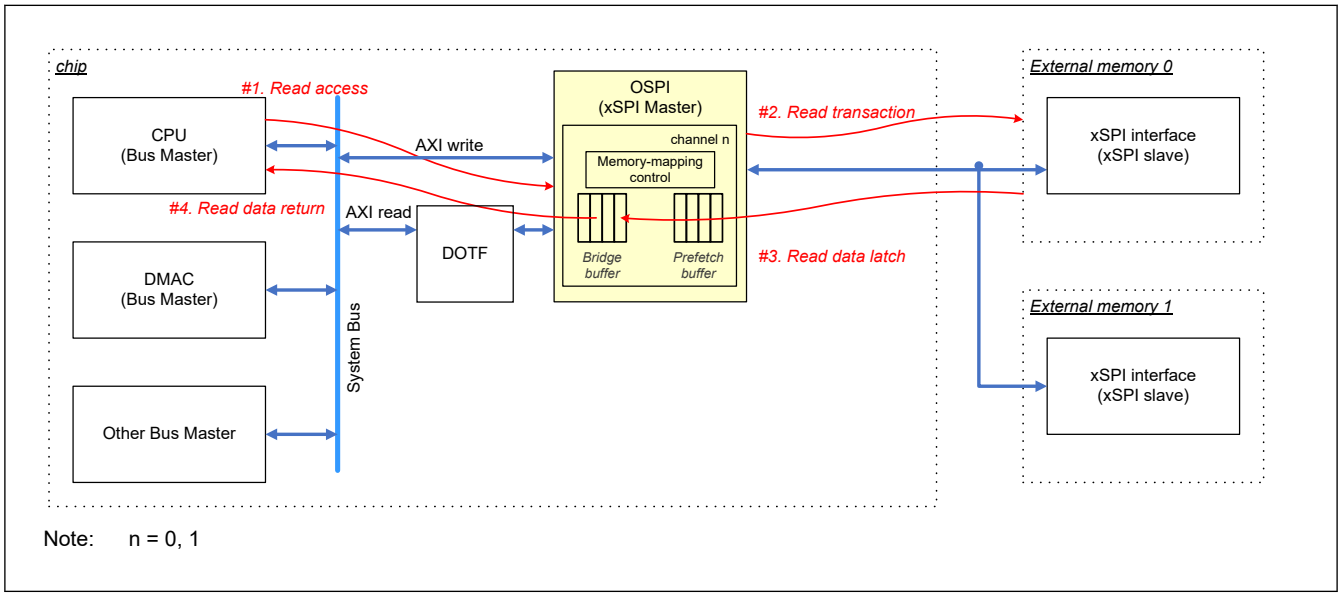


Figure 37.16 Read access for memory area

The operation of xSPI bus changes depending on burst type. When the type is single or increment type, one system bus's read transaction triggers one xSPI frame. When the type is wrap type and the CMCFG0CSn.WPBSTMD is 0, one system bus read transactions triggers two xSPI frames. Figure 37.17 shows the relationship between AXI and xSPI frames.

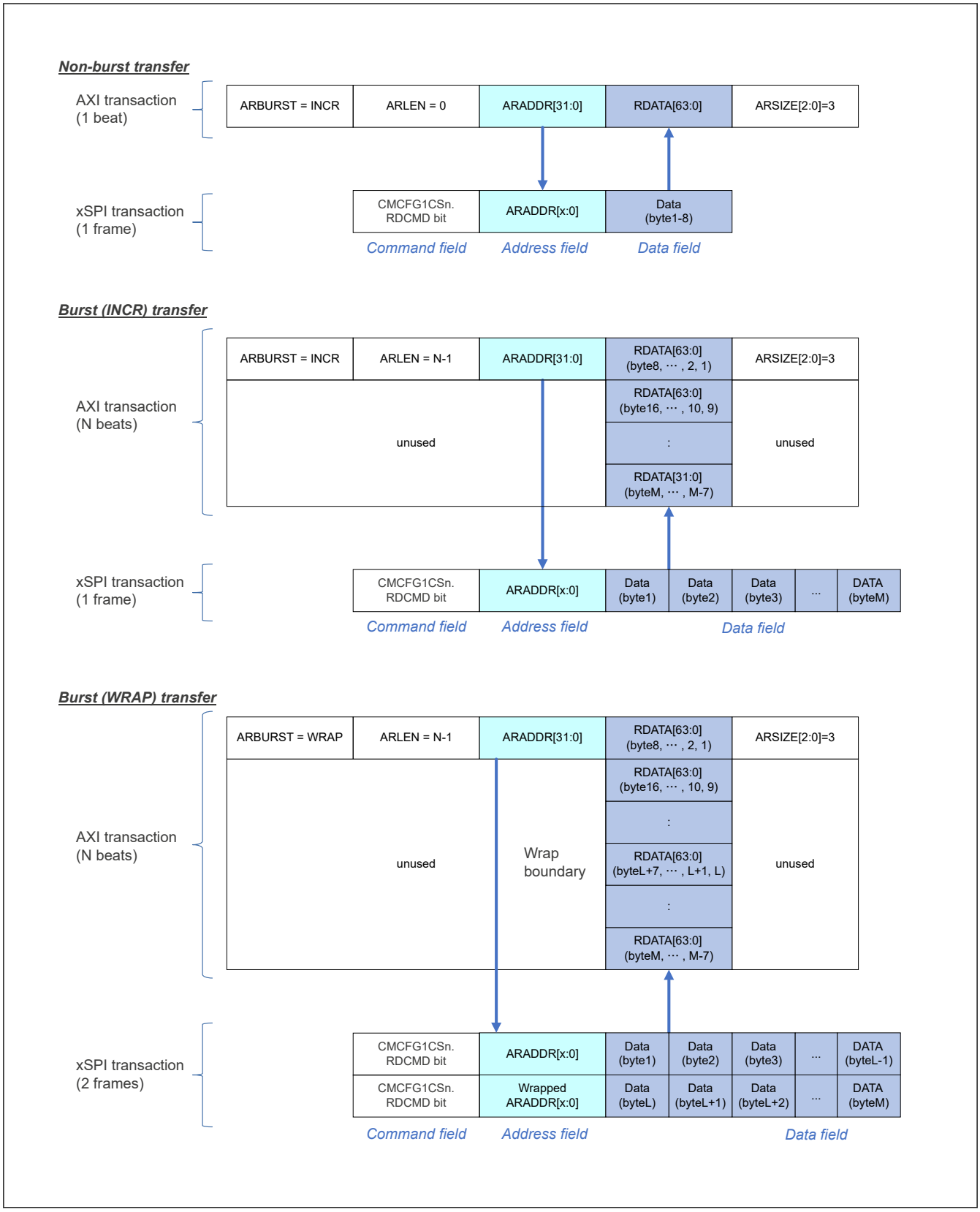


Figure 37.17 xSPI frame format in read access (Normal format)

37.3.3.5 Prefetch Function

At the read access for memory area from system bus, this xSPI Master has the function to prefetch the read data for reducing the latency. When enabled this function (BMCFGCHn.PREEN = 1), this xSPI Master continues to read the

incremental address and store the read data from xSPI slave in internal prefetch buffer. And this xSPI master searches in prefetch buffer for the following read access from system bus. If found the target read data in prefetch buffer, this xSPI Master returns the data from prefetch buffer. If not found, this xSPI Master clears prefetch buffer and newly issues a read transaction into xSPI slave. This function is effective in application such as the consecutive read addresses are close. But it is not effective in application such as the consecutive read addresses are not incremental because xSPI read frame for prefetch uses xSPI bus. Figure 37.18 shows the operation summary.

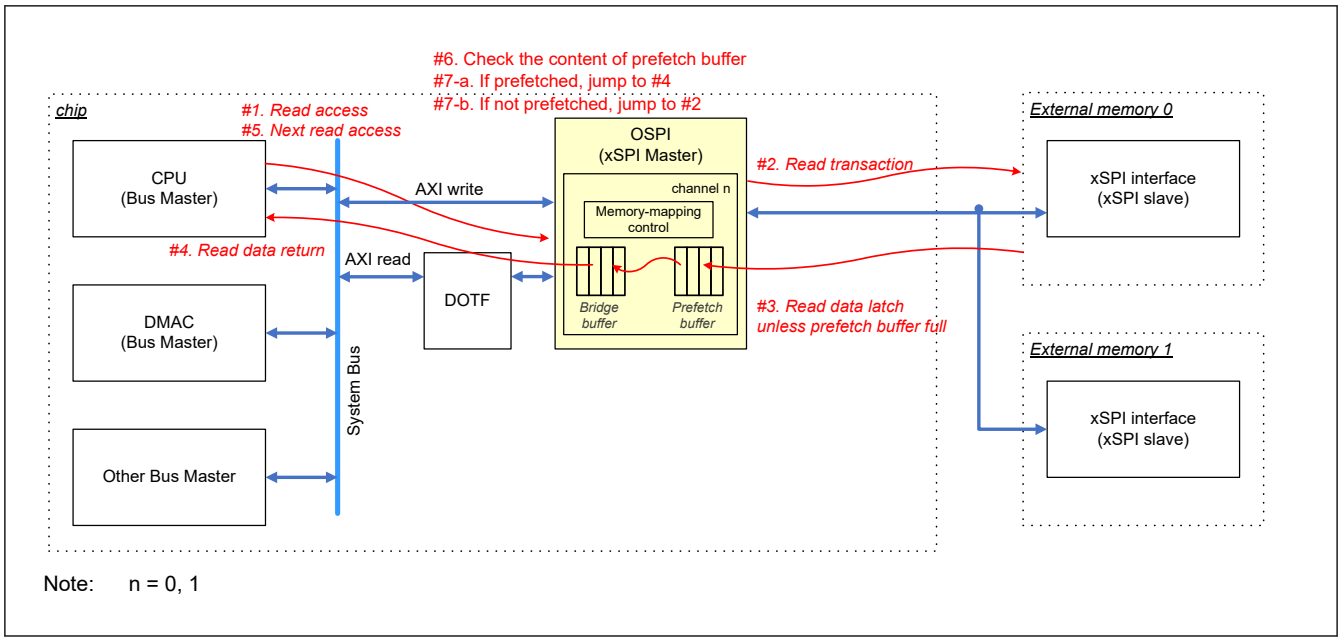


Figure 37.18 Read access for memory area with prefetch enabled

Note: When enabled this prefetch function, Bus Master could read from not a slave device but the internal prefetch buffer. When accessed to the same address from multiple Bus Masters, this xSPI Master does not guarantee to read the latest data. If Bus Master wish to read the latest data from a slave device, it should read after cleared the prefetch buffer (BMCTL1.PBUFCLRCHn (n = 0, 1)).

Note: Prefetch buffer is implemented as FIFO-based, and when read access is issued, the data before the access address is discarded from the buffer. And when next access is issued to the region which is discarded at previous access, this module issues the xSPI read access again to fill the prefetch buffer.

Note: OSPI has the 1line buffer which keeps the last 8Byte read data from prefetch buffer. When read access is issued to the same address to the data in 1line buffer, OSPI returns read data from 1line buffer.

37.3.3.6 XiP Mode

Some slave devices have a mode (XiP mode) in which the command phase is not required for lower latency. While in this mode, the xSPI master skips sending the command and the slave device implicitly performs the command that was executed in the previous transaction. When enabled XiP mode bit (CMCTLCHn.XIPEN = 1), this xSPI master inserts XiP enter code (CMCTLCHn.XIPENCODE[7:0]) in latency field. When disabled XiP mode bit (CMCTLCHn.XIPEN = 0), this xSPI master inserts XiP exit code (CMCTLCHn.XIPEXCODE[7:0]) in latency field. This function is available only for memory-mapping mode.

And when this xSPI master transmits XiP disable pattern, this master clears XiP mode bit and disables XiP mode configured for both channels. Note that it is not possible to disable XiP mode for only one channel by transmitting XiP disable pattern.

Note: When enough latency cycle does not exist for XiP code, this xSPI Master could not insert XiP code.

Note: XiP mode could be used only for unidirectional access to a slave. The write transaction and read transaction should be separated.

Note: The XiP exit code is inserted once when disabled. More details, See Figure 37.30.

37.3.4 Pattern Control

This xSPI Master has the function to transmit 3 type of patterns which is not xSPI frame format. The pattern is triggered by setting trigger bit (LPCTL0-1.PATREQ).

37.3.4.1 XiP Disable Pattern

XiP Disable pattern transmits any pattern with the configured length and value (LPCTL0.XD1LEN[4:0] / XD1VAL / XD2LEN[4:0] / XD2VAL). It uses OM_SCLK, OM_SIO7-0 signals. The number of output pin can be configured by XiP Disable pattern pin bits (LPCTL0.XDPIN[1:0]). It may be used to disable XiP mode for legacy SPI. Figure 37.19 shows the timing-chart.

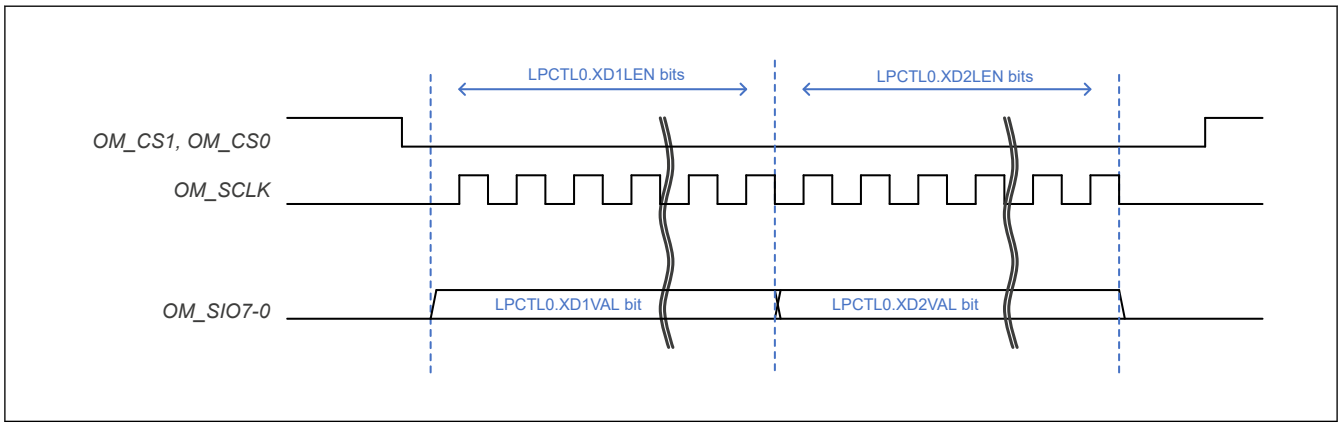


Figure 37.19 XiP Disable pattern

37.3.4.2 Reset Pattern

Reset pattern transmits the pattern specified in Serial Flash Reset Signaling Protocol. Figure 37.20 shows the timing chart.

CS Low/High width is configured with Reset Pattern Length bits (LPCTL1.RSTWID[2:0]). xSPI slave will sample the data input at the rising edge of CS. Setup time for data output is configured with Reset pattern data output setup time bits (LPCTL1.RSTSU[2:0]). The setup time should be less than Reset pattern width always.

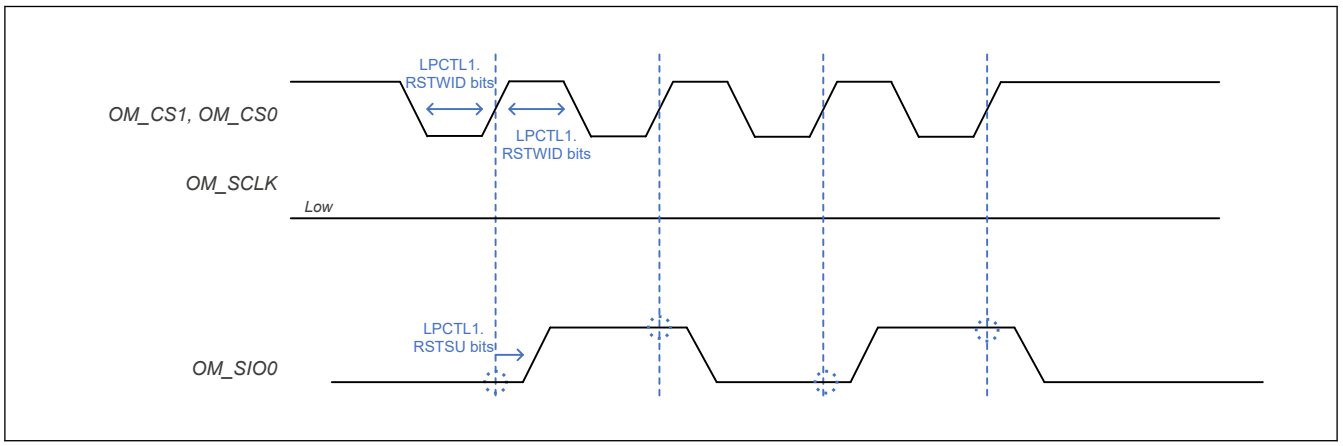


Figure 37.20 Reset pattern

Note: In the protocol, CS Low/High width is defined as minimum 500 ns and Setup time is defined as minimum 6 ns.

37.3.4.3 CS Only Pattern

CS Only pattern activates CS port with the configured length bits (LPCTL1.RSTWID[2:0]). It may be used to resume from Deep Power Down state. Figure 37.21 shows the timing-chart.

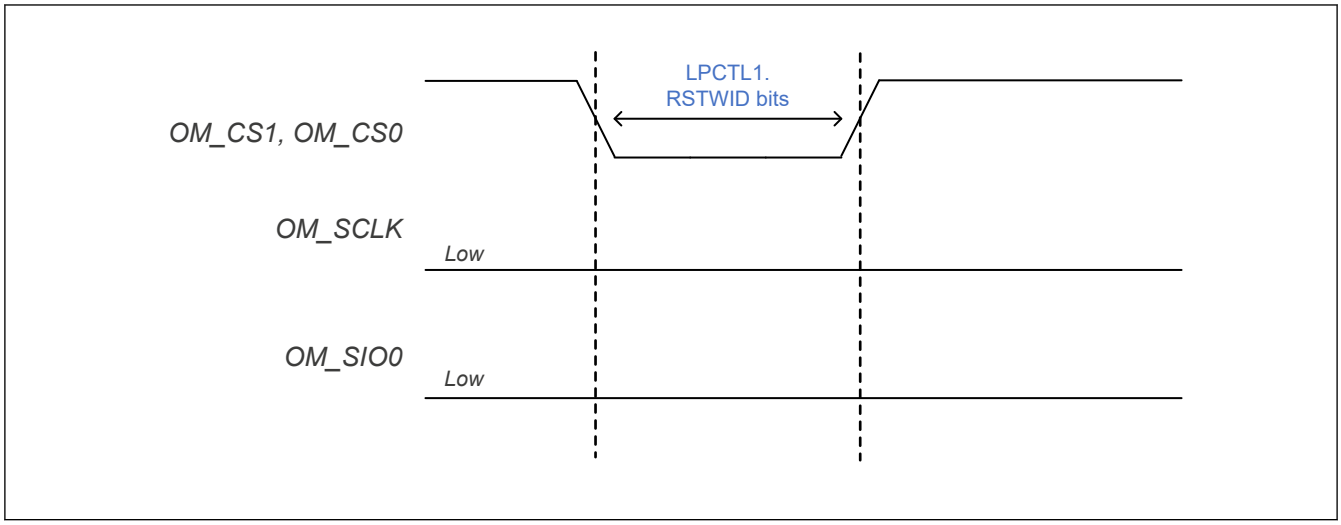


Figure 37.21 CS Only pattern

37.3.5 Integrity Checking

This xSPI Master can detect some errors. Table 37.8 shows the error list and the detail behavior.

Table 37.8 Error list (n = 0, 1)

Error type	Event	Flag bit	Note (action)
Calibration failed	When the read data did not match the expected value during automatic calibration.	INTS.CAFAILCSn	It results in writing unexpected data to xSPI slave.
System bus error	When an error response occurred on AXI slave interface for memory-mapping.	INTS.BUSERRCHn	This xSPI master shall be reset for fatal error.
ECC error detection	When detected the falling edge on OM_ECSINT1 port. It can be useful only for xSPI slave with ECC detection function.	INTS.ECSCS1	Only notify the error event of xSPI slave.
OM_DQS timeout	When OM_DQS does not toggle in read transaction with using OM_DQS.	INTS.DSTOCSn	If this error occurs during calibration, clear the interrupt status bit. Other than above, both xSPI master and xSPI slave should be reset for fatal error.
Periodic transaction timeout	When the read value does not match with the expected value in periodic manual-command mode.	INTS.PERTO	Depending on the status of function.

37.3.6 Interrupts

This xSPI Master has an interrupt port.

It can monitor with Interrupt Status Register (INTS). In case of initialization phase, it can be programmable with Interrupt Enable register (INTE). Table 37.9 shows OSPI interrupt sources, and Table 37.10 shows the related register bit.

Note: Interrupt pulse port signal is not asserted when the corresponding bit of INTE is set after the interrupt event is detected.

Table 37.9 OSPI interrupt sources

Name	Interrupt sources	DMAC activation
OSPI0_ERR	Error	Not possible
OSPI0_CMP	Complete	Not possible

Table 37.10 Interrupt register bit

Flag bit	Enable bit	Clear bit	Interrupt sources
CASUCCS1	CASUCCS1E	CASUCCS1C	OSPI0_CMP
CASUCCS0	CASUCCS0E	CASUCCS0C	
CAFAILCS1	CAFAILCS1E	CAFAILCS1C	OSPI0_ERR
CAFAILCS0	CAFAILCS0E	CAFAILCS0C	
BUSERRCH1	BUSERRCH1E	BUSERRCH1C	OSPI0_ERR
BUSERRCH0	BUSERRCH0E	BUSERRCH0C	
INTCS1	INTCS1E	INTCS1C	OSPI0_ERR
ECSCS1	ECSCS1E	ECSCS1C	OSPI0_ERR
DSTOCS1	DSTOCS1E	DSTOCS1C	OSPI0_ERR
DSTOCS0	DSTOCS0E	DSTOCS0C	
PERTO	PERTOE	PERTOC	OSPI0_ERR
PATCMP	PATCMPE	PATCMPC	OSPI0_CMP
CMDCMP	CMDCMPE	CMDCMPC	OSPI0_CMP

37.3.7 Flows of Operations

37.3.7.1 Flow of Configuration

Figure 37.22 shows flow of configuration.

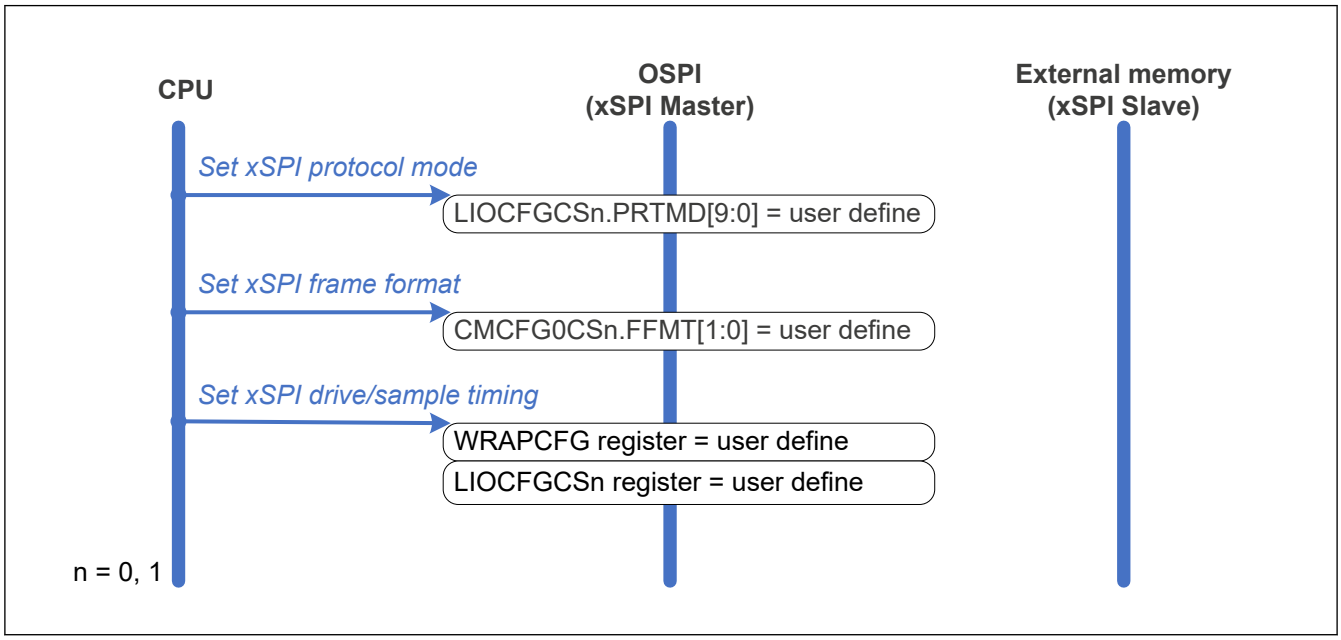


Figure 37.22 Flow of configuration

37.3.7.2 Flow of Communication Stop

Figure 37.23 shows flow of communication stop.

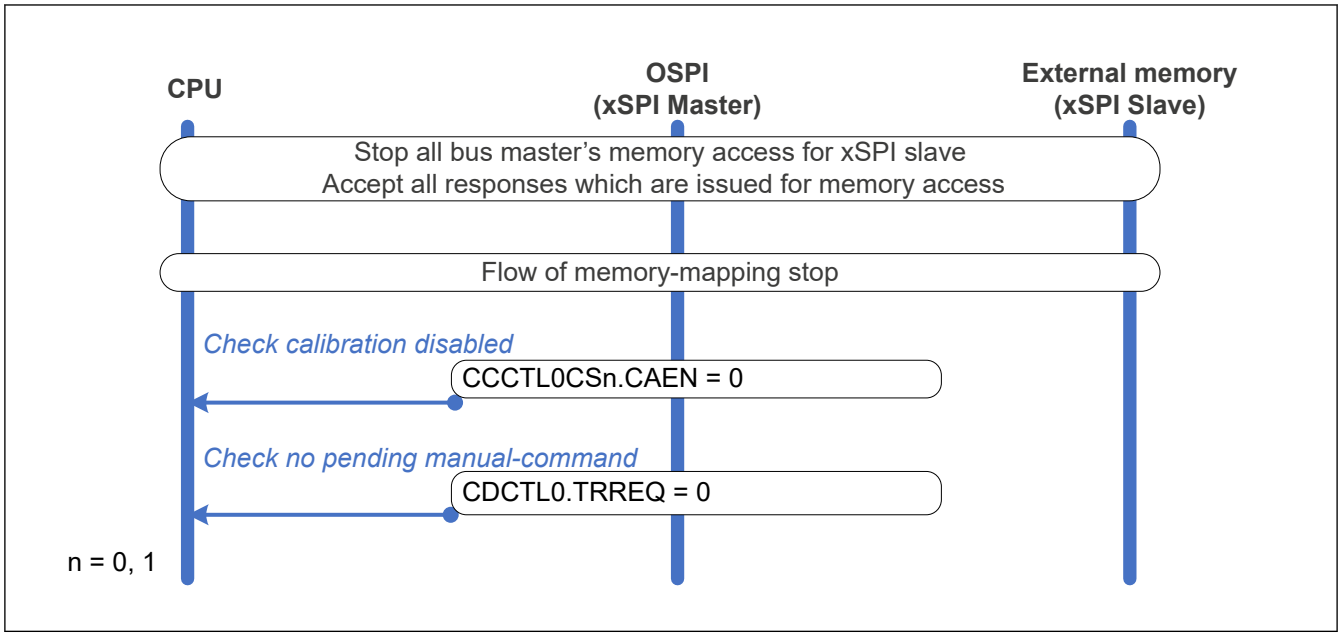


Figure 37.23 Flow of communication stop

Note: In case of re-config of any configuration register, all communication with xSPI slave shall be stopped surely to avoid race condition between register setting and memory access. It means that the automatic calibration is disabled, and there is no pending manual-command and memory-mapping access.

37.3.7.3 Flow of Automatic Calibration

Figure 37.24 shows flow of automatic calibration.

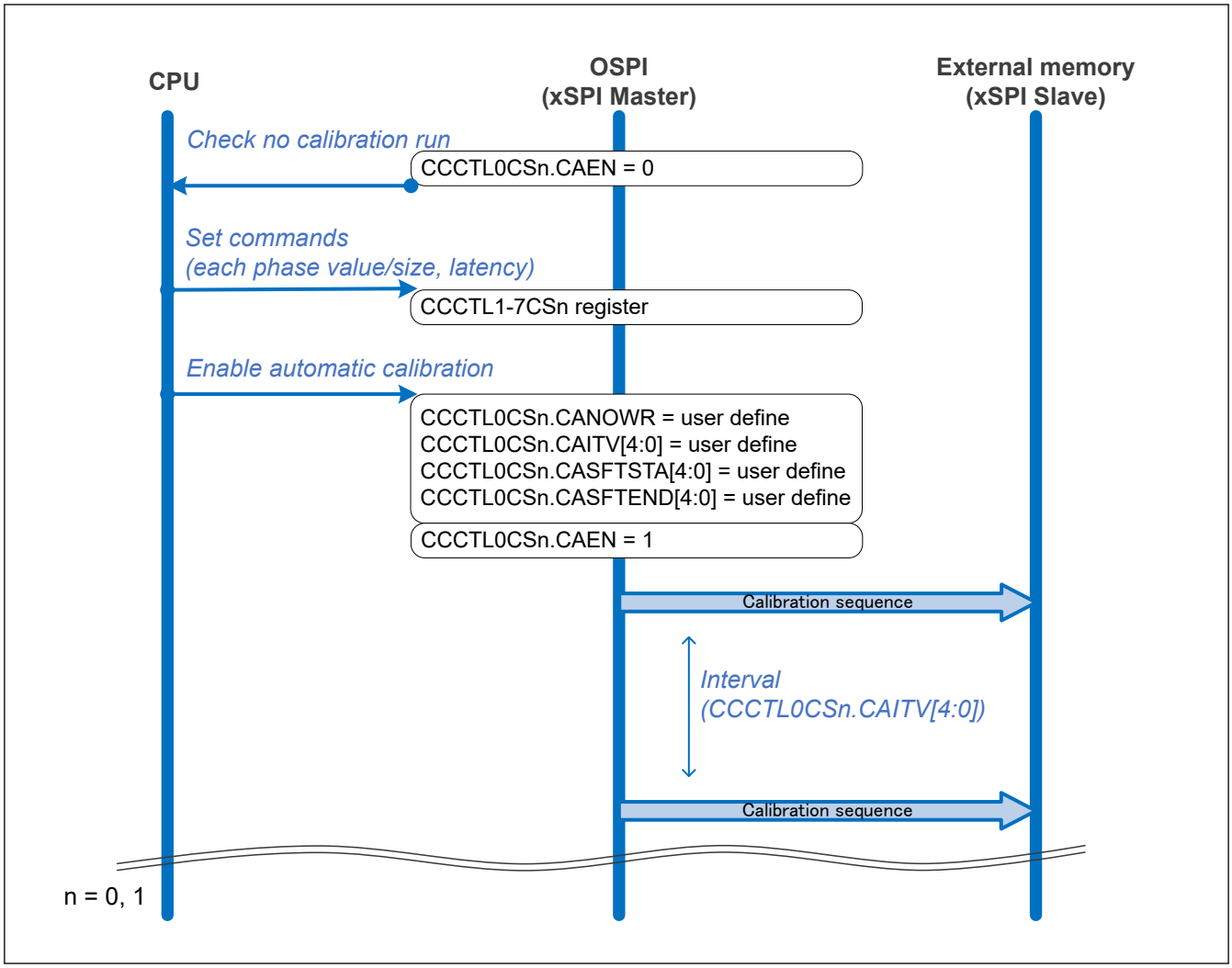


Figure 37.24 Flow of automatic calibration

37.3.7.4 Flow of Manual-command Procedure

Figure 37.25 shows manual-command procedure for direct mode.

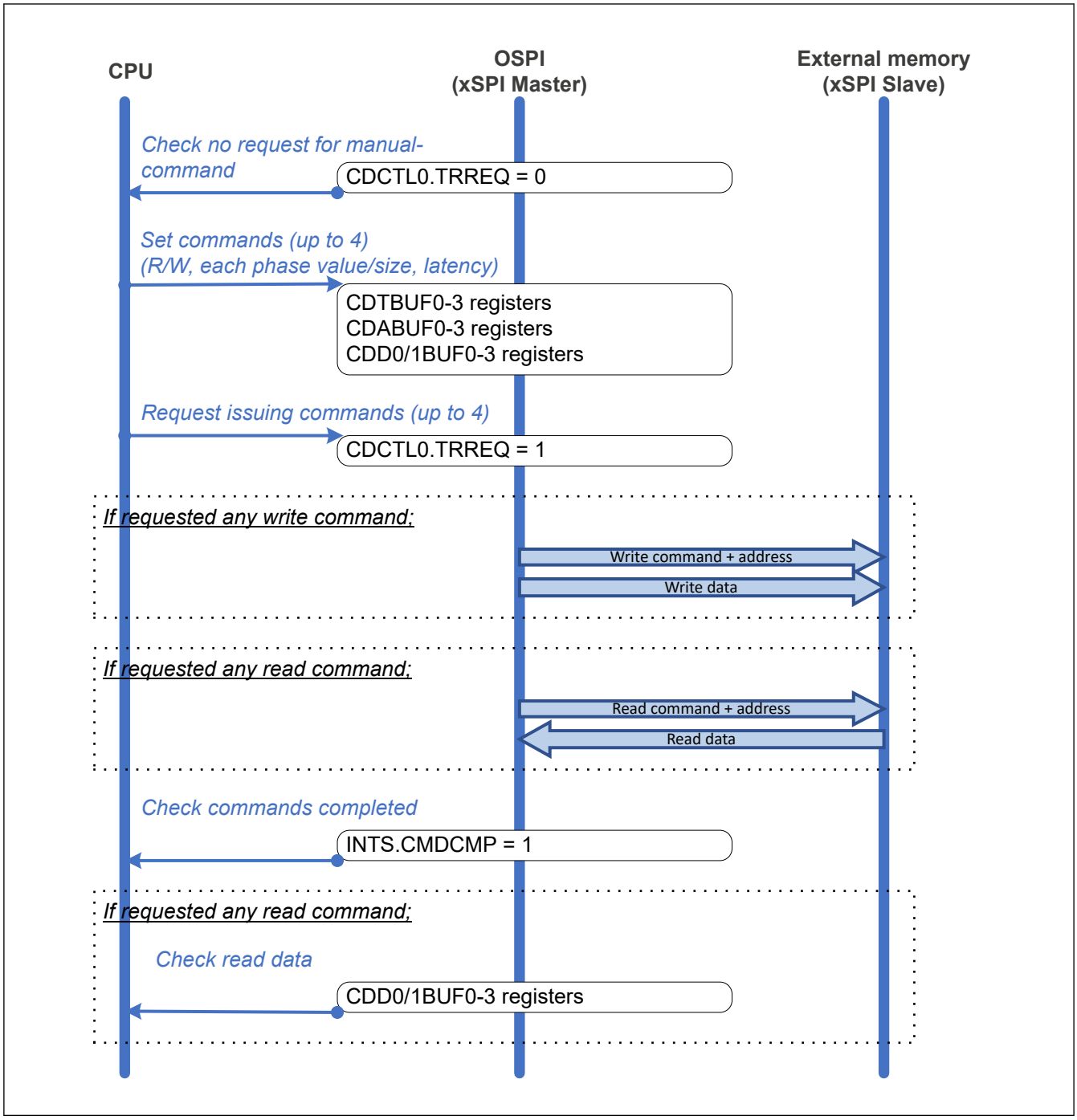


Figure 37.25 Flow of manual-command procedure for direct mode

Figure 37.26 shows manual-command procedure for periodic mode.

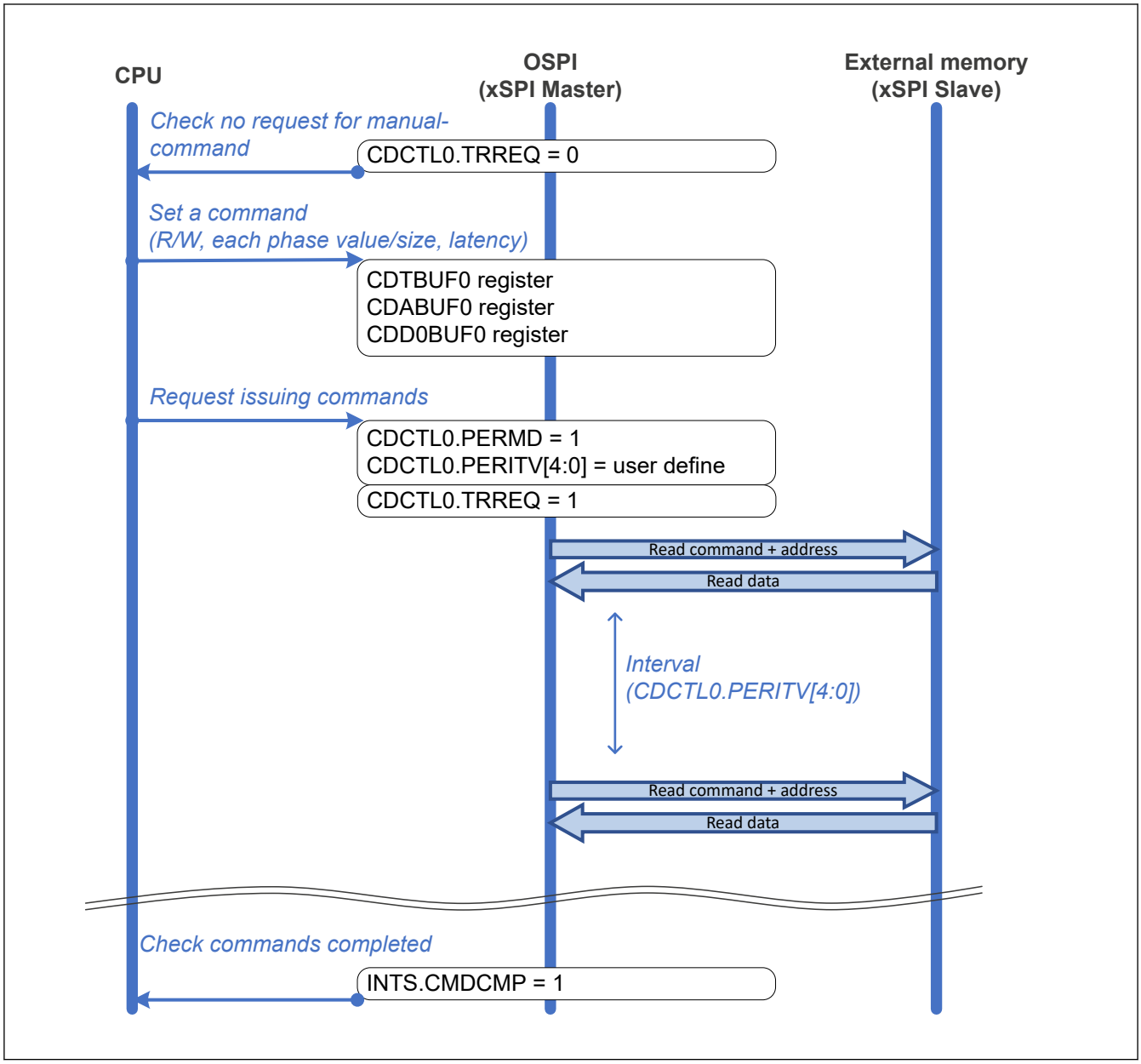


Figure 37.26 Flow of manual-command procedure for periodic mode

37.3.7.5 Flow of Memory-mapping

Figure 37.27 shows flow of memory-mapping.

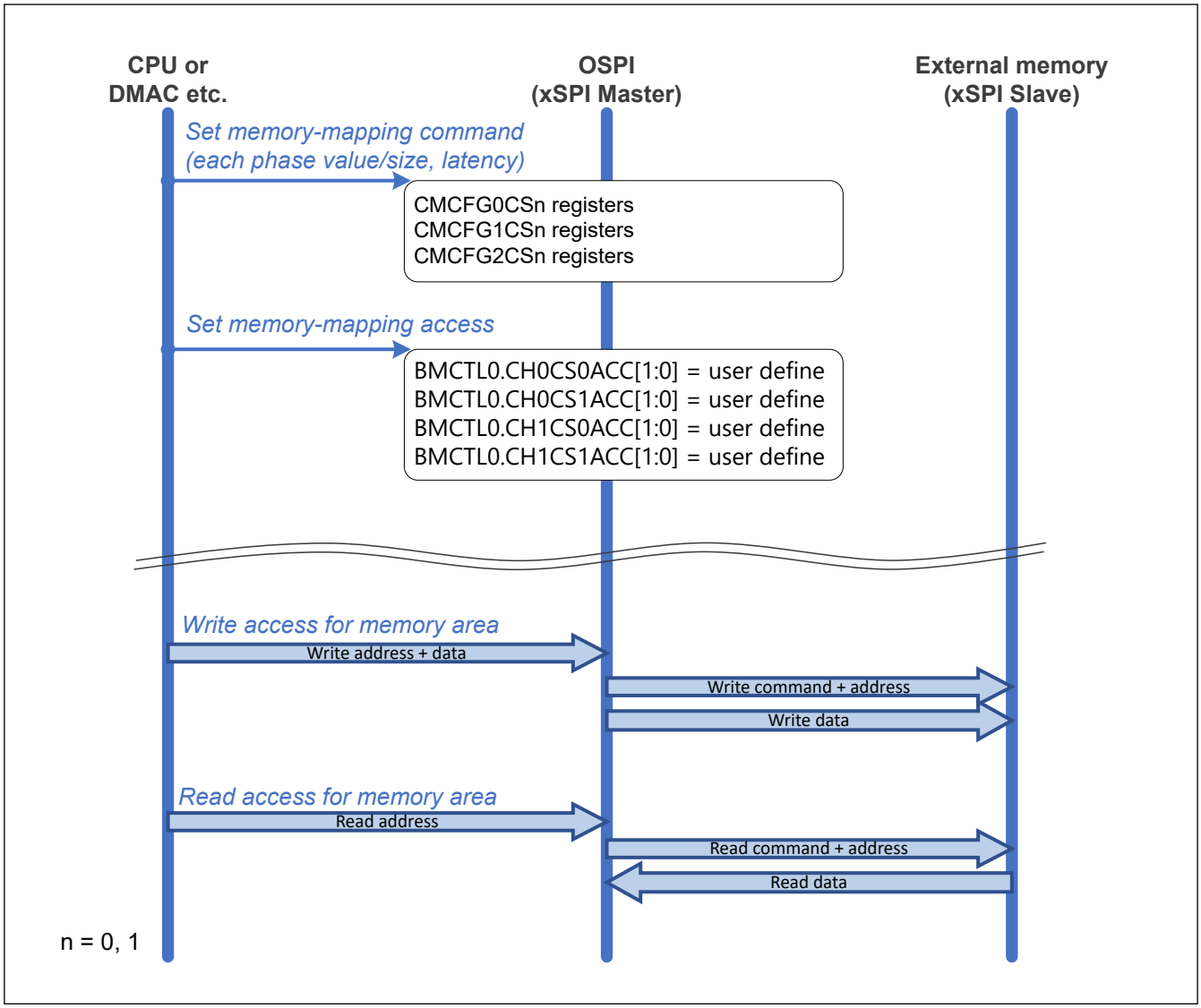


Figure 37.27 Flow of memory-mapping

37.3.7.6 Flow of Memory-mapping Stop

Figure 37.28 shows flow of memory-mapping stop.

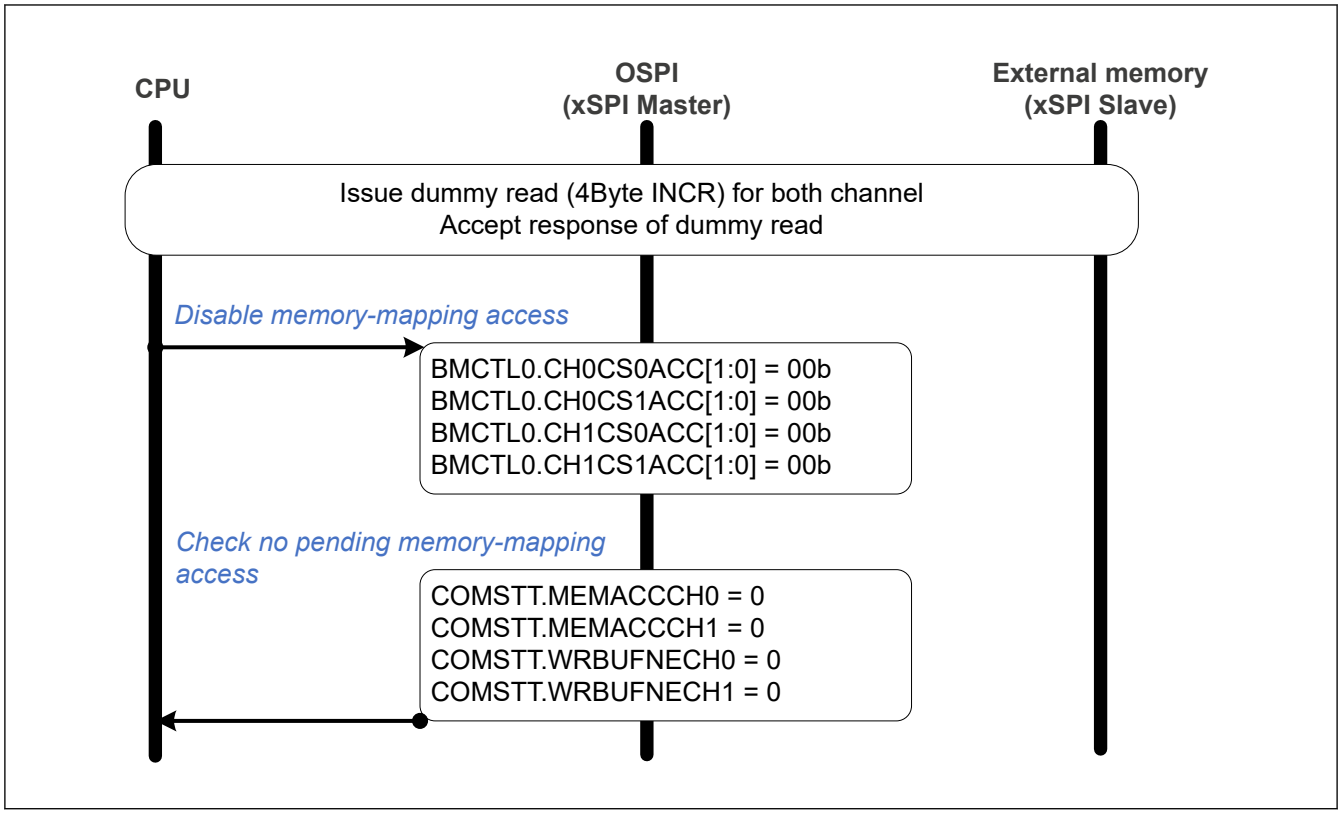


Figure 37.28 Flow of memory-mapping stop

37.3.7.7 Flow of Pattern Request

Figure 37.29 shows flow of pattern request. Before requesting any pattern, any ongoing commands should be completed or canceled.

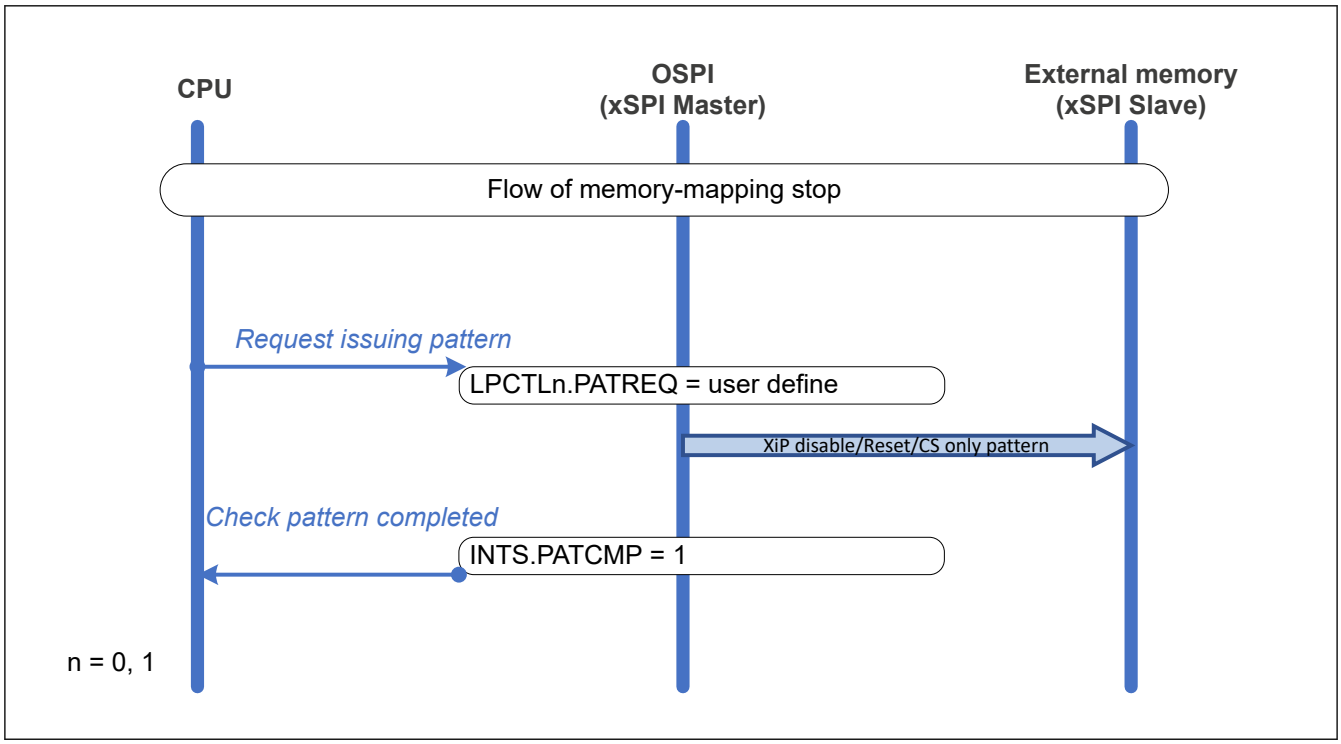


Figure 37.29 Flow of pattern request

37.3.7.8 Flow of XiP Mode

Figure 37.30 shows flow of XiP mode enable/disable.

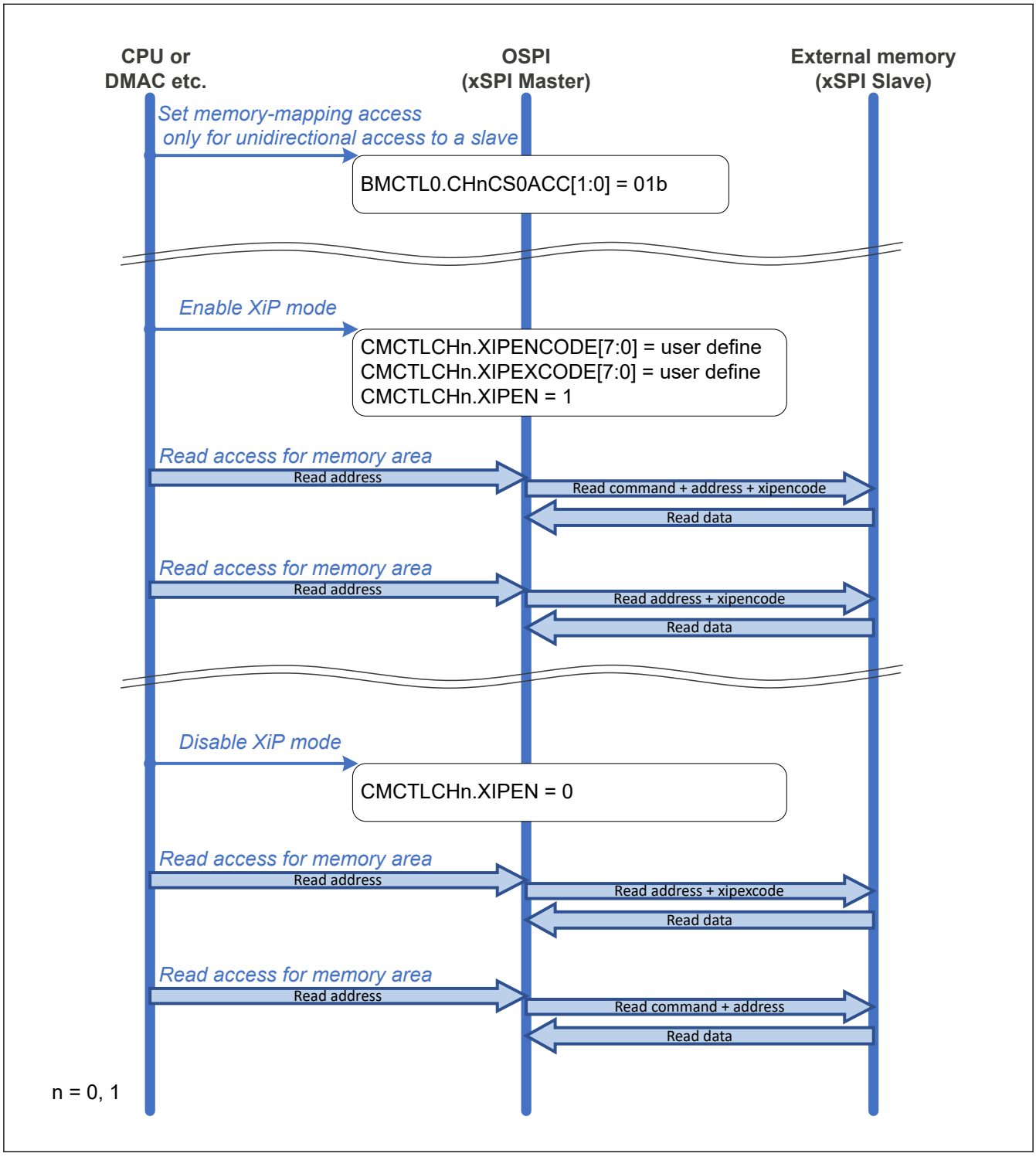


Figure 37.30 Flow of XiP mode enable/disable

37.3.7.9 Flow of Read While Write

Figure 37.31 shows example flow of read while write.

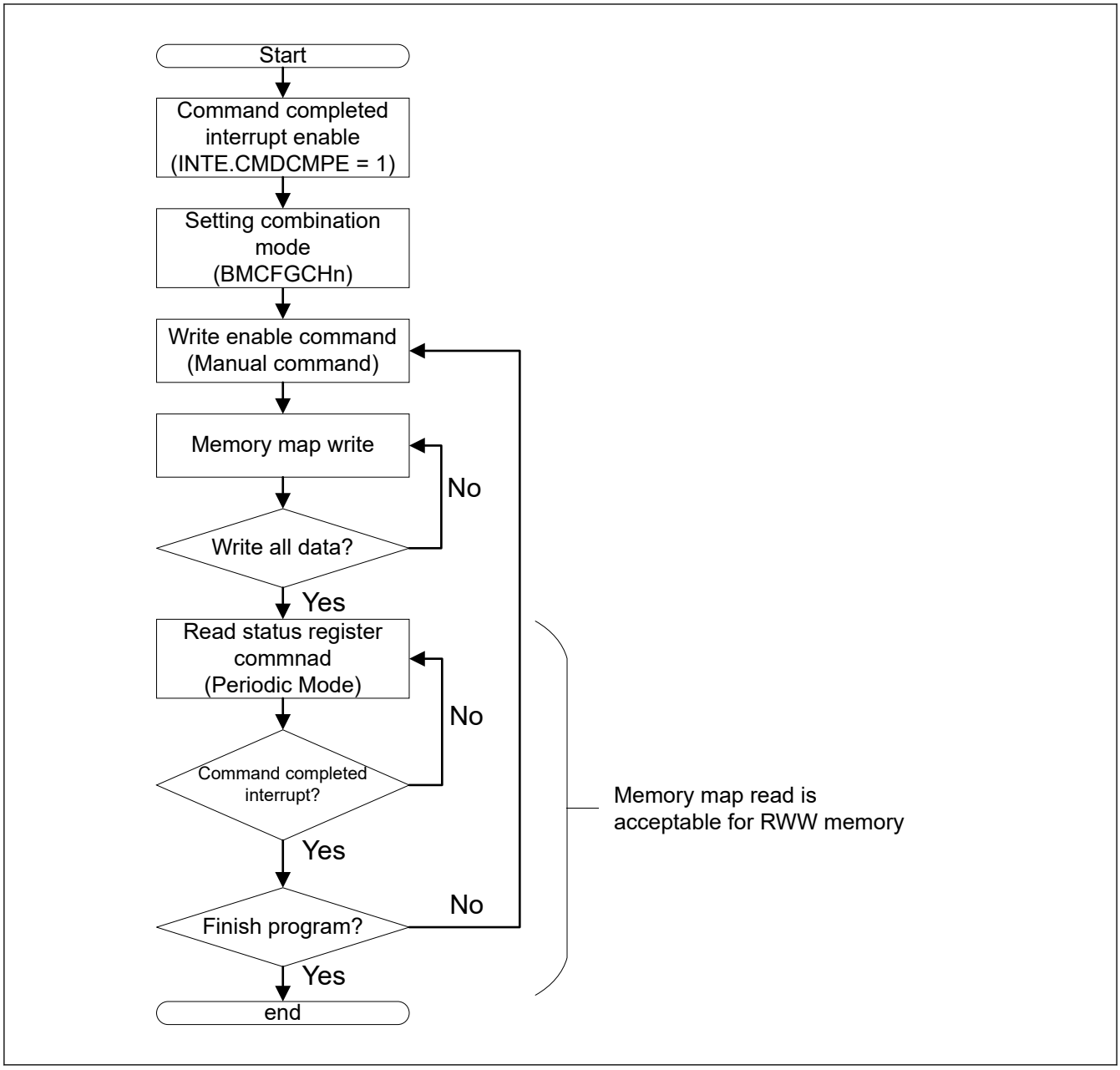


Figure 37.31 Flow of read while write

37.4 Usage Notes

37.4.1 Constraint on Memory Access

The memory access has the following constraints:

- Not using prefetch function if application access to the last 128 byte area
- Enable prefetch function when read HyperFlash by DRW.

37.4.2 Constraint on Burst Length

AXI bus master must have a Burst Length of 16 or less.

37.4.3 Memory Write Combination Mode

When BMCFGCHn.MWRCOMB is 1, memory write combination mode is enabled. However, combination does not work if the bus master is CPU, and OSPI data is under 32bit. Table 37.11 shows the possibility of data write for each bus master.

Table 37.11 Data write possibility for each bus master

Bus Master	Combination Enable	Combination Disable
CPU under 64 bit Access	Not possible	Possible
CPU 64 bit Access	Possible	Possible
DMAC/DTC	Possible	Possible
EDMAC	Possible	Possible
CEU	Possible	Possible
DRW	Possible	Possible

37.4.4 Module-stop function

OSPI operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The OSPI module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

37.4.5 Restriction in 8D-8D-8D profile 1.0 format

"When read or write access to the Octal Serial Peripheral Interface (OSPI) is performed from the following bus masters with 8-bit wide, access to odd addresses is prohibited as it will not be worked correctly. See Table 37.12 for the applicable conditions and their workarounds for each bus master"

Table 37.12 Workaround for the restriction in 8D-8D-8D profile 1.0 format

Bus master		Conditions that require workaround	Workaround
CPU	Normal memory	Debugger is connected	Access only even addresses.
	Device memory	Always	Access only even addresses.
DMAC/DTC	DMAC	DMTMD.SZ[1:0] = 00b	Access only even addresses.
	DTC	MRA.SZ[1:0] = 00b	Access only even addresses.
DRW		Settings of CONTROL2.WRITEFORMAT [1:0] bits and ORIGIN register	Set CONTROL2.WRITEFORMAT[1:0] only in 16 bpp or 32 bpp, and set even addresses only in the ORIGIN register.

37.4.6 Byte data order in 8pin-DDR mode

In 8pin-DDR mode, OSPI read and write data in the order of byte0 at rising edge and byte1 at falling edge.

Some memory devices read and write in the swapped order in 8pin-DDR mode. For these devices, if 8pin-DDR read the data which is written in other than 8pin-DDR mode, this data should be written in swapped order.

38. Decryption On The Fly (DOTF)

38.1 Overview

DOTF has a function to decode read data of the AXI bus using AES core.

Table 38.1 lists the DOTF specifications and for a system using DOTF, see section 37.1. Overview.

Table 38.1 DOTF Specification

Item	Description
Clock Source	Register clock : PCLKB AES core clock : PCLKA
AES core function	• Utilize for on-the-fly decryption of encrypted software stored in external memory. • Block size: 128-bit • Key size: 128-bit, 192-bit, 256-bit • Support the following block cipher mode. − Counter (CTR) mode following NIST SP800-38A • Support side channel counter measure function. • Supports self-test function. • Counter[127:0] = {IV[127:28], Address[31:4]}, where Address is the memory mapped address of the encrypted data, aligned to the AES block size of 128 bits (16 bytes). The IV shall be chosen according to the recommendations provided in appendix B of SP800-38A. See section 37, Octal Serial Peripheral Interface (OSPI) and section 4.1. Address Space for details. See NIST SP800-38A for details of AES counter mode operation.
Tamper Resistance	Countermeasures available for side-channel attacks, including SPA/DPA and timing attacks
Module-stop function	Module-stop state can be set to reduce power consumption. same as OSPI module stop
TrustZone Filter	Security and Privilege attribution can be set for each channel. same as TZF of OSPI

38.2 Block Diagram

The block diagram of DOTF is shown in Figure 38.1. See Figure 37.1 for a diagram including the OSPI.

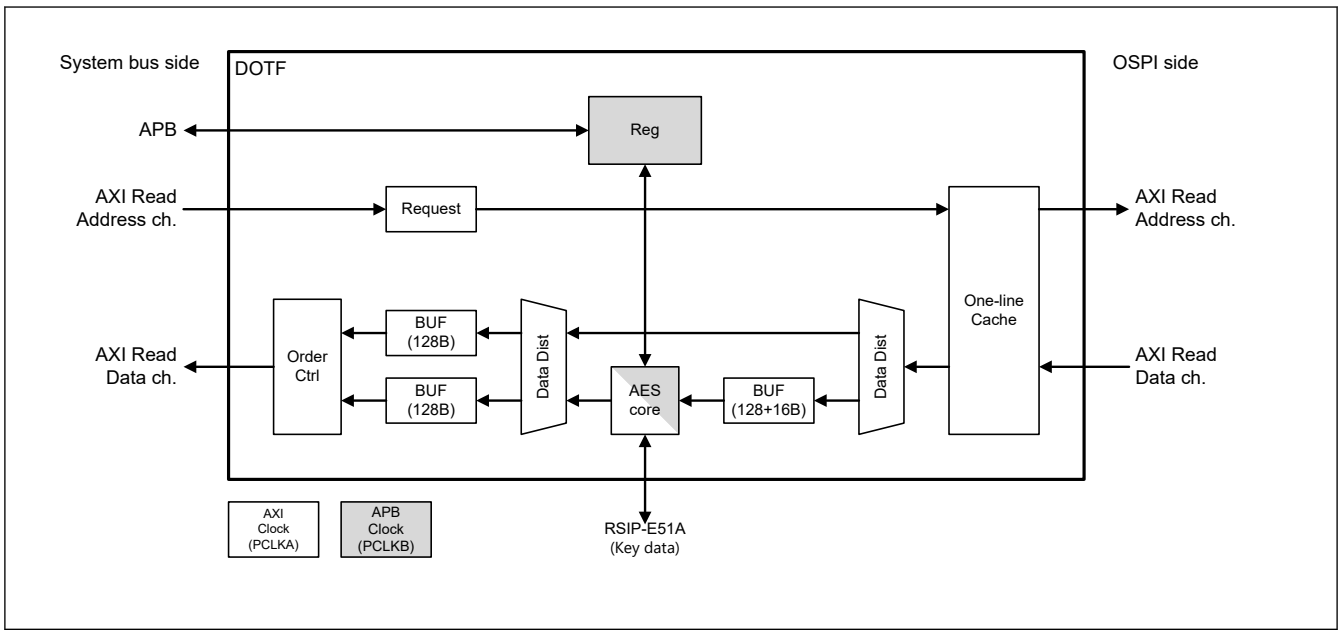


Figure 38.1 Block diagram

38.3 Register Descriptions

DOTF allows the entire area to be an accessible area, but the area where decryption processing is performed can be specified as the entire area or one contiguous area. The region can be specified by setting the start and end addresses to the CONVAREAST and CONAREAED registers, respectively.

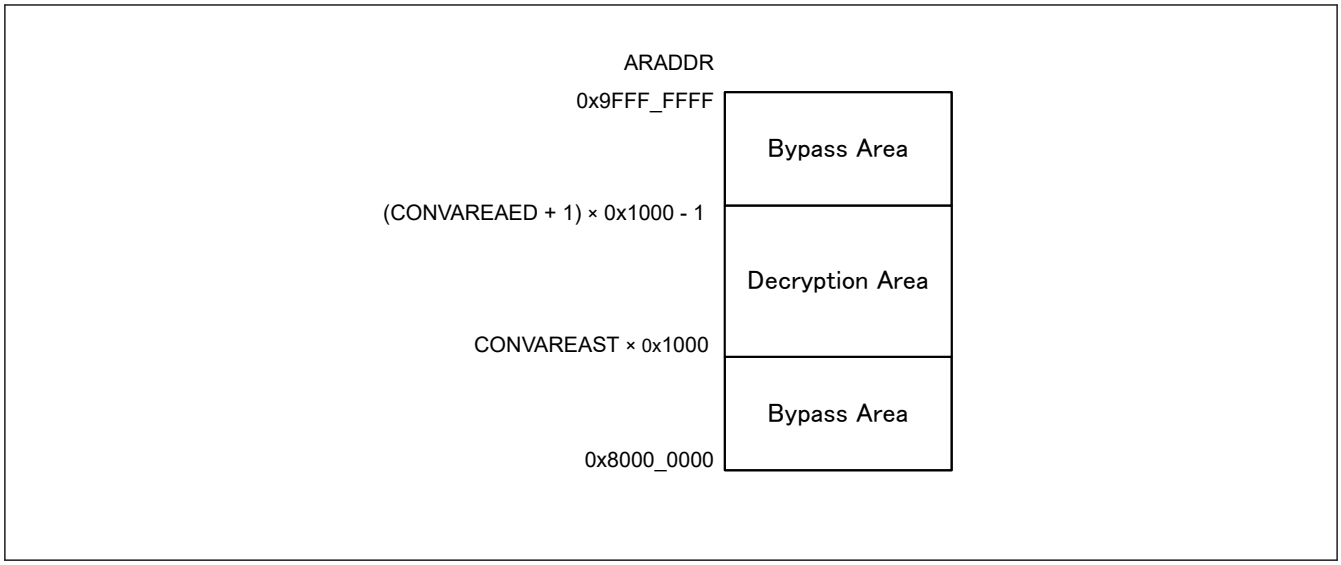


Figure 38.2 Image of decryption area setting

38.3.1 CONVAREAST : DOTF Conversion Area Start Address Register

Base address: DOTF0 = 0x4026_8800
DOTF0_NS = 0x5026_8800

Offset address: 0x00

Bit position:	31																			12												0
Bit field:	CONVAREAST[31:12]																			—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	—	These bits are read as 0.	R/W
31:12	CONVAREAST [31:12]	The first address of the decryption processing area. The actual address is CONVAREAST [31:12] × 0x1000.	R/W

Note: S-TYPE-3, P-TYPE-3

Specify the first address of the decryption processing area. Set before the AXI transfer request and do not make any further changes. Setting CONVAREAST[31:12] > CONVAREAED[31:12] is prohibited.

38.3.2 CONVAREAD : DOTF Conversion Area End Address Register

Base address: DOTF0 = 0x4026_8800
DOTF0_NS = 0x5026_8800

Offset address: 0x04

Bit position:	31																			12												0
Bit field:	CONVAREAED[31:12]																			—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	—	These bits are read as 0.	R/W
31:12	CONVAREAED [31:12]	The end address of the decryption processing area. The actual address is (CONVAREAED [31:12] + 1) × 0x1000 - 1.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Setting CONVAREAST[31:12] > CONVAREAED[31:12] is prohibited.

Specify the tail address of the decryption processing area. Set before the AXI transfer request and do not make any further changes. Setting CONVAREAST[31:12] > CONVAREAED[31:12] is prohibited.

38.4 Operation

DOTF has one AXI Read Slave Interface and one AXI Read Master Interface.

An AXI transfer request received on the AXI Slave Address channel is issued to the AXI Master's Read Address channel, and the Slave data is received from the Master's AXI Read data channel and sent to the AXI Slave Read data channel.

At this time, the necessity of decryption is determined by the requested address area, and the decrypted data is transmitted through AES core if necessary. When decryption is not required, the received response is sent without conversion.

38.5 System flow

The initialization flow after HW reset is shown in Figure 38.3.

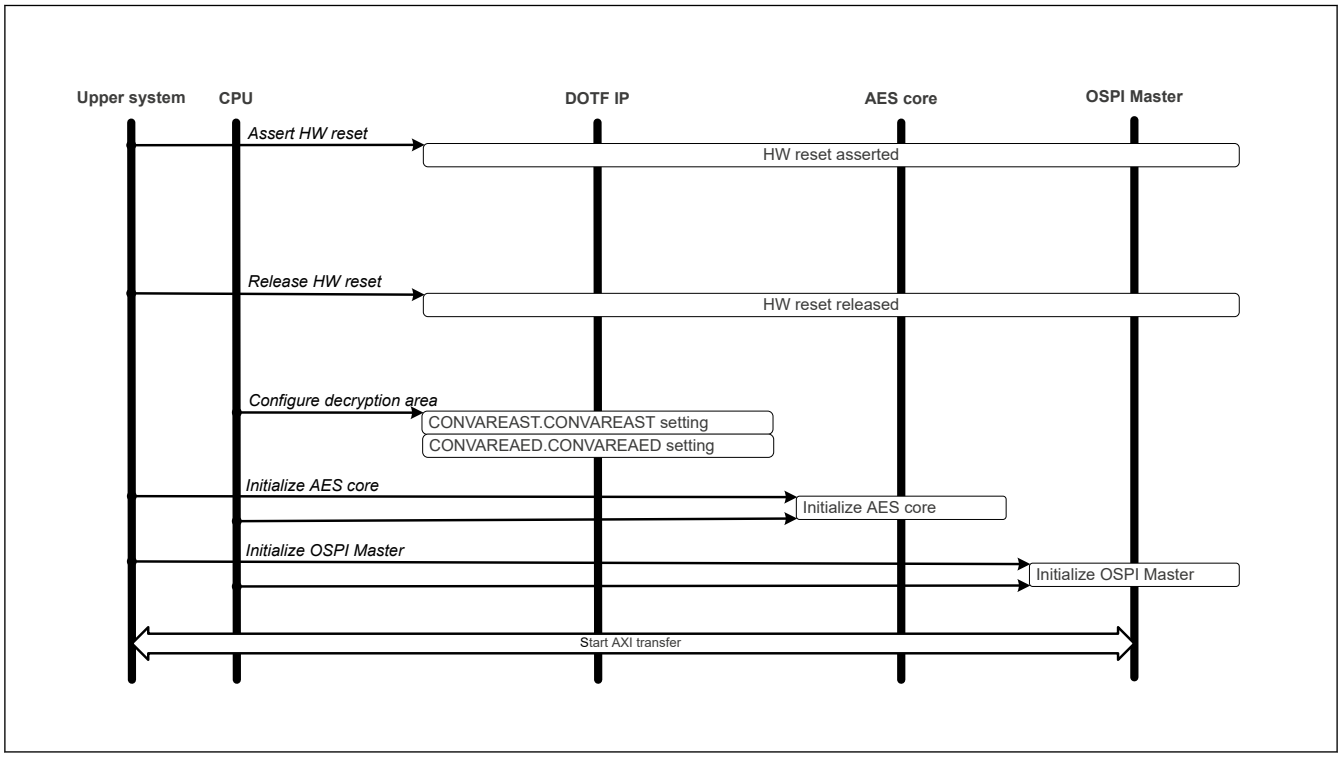


Figure 38.3 Initialization flow

38.6 Usage notes

38.6.1 Module-stop function

DOTF operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The DOTF module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

39. Serial Sound Interface Enhanced (SSIE)

39.1 Overview

The Serial Sound Interface Enhanced (SSIE) can transmit and receive audio data to and from various devices that support any of audio data formats, such as I^2S, monaural, and TDM.

Table 39.1 lists the SSIE specifications, Figure 39.3, Figure 39.4 and Figure 39.5 show a block diagram of SSIE, and Table 39.4 lists the I/O pins.

Table 39.1 SSIE specifications

Item		Description
Number of channels		Two channels, SSIE0 and SSIE1
Communication mode		• Master/slave • Transmission/reception (SSIE0:full duplex communication, SSIE1:half-duplex communication)
Communication format		• I^2S format • Monaural format • TDM format
Serial data		• MSB first • Data can be left-justified or right-justified. • Data delay (1 clock cycle) or no delay selectable for the period from SSILRCKn/SSIFSn (n = 0, 1) to SSITXD0/SSIRXD0/SSIDATA1 • System word length: 8, 16, 24, 32, 48, 64, 128, or 256 bits • Data word length: 8, 16, 18, 20, 22, 24, or 32 bits • Padding polarity: Low or high
Bit clock (SSIBCKn (n = 0, 1))	In master mode	• Two clock sources available (AUDIO_CLK/GPT output (GTIOC2A)) • Clock source division ratio: 1/1, 1/2, 1/4, 1/6, 1/8, 1/12, 1/16, 1/24, 1/32, 1/48, 1/64, 1/96, and 1/128. • Supply/stop is selectable while communication is halted.
	In master/slave mode	• Polarity (rising edge or falling edge) selectable
LR clock/frame synchronization (SSILRCKn/SSIFSn (n = 0, 1))	In master mode	• Polarity (low level or high level) selectable • Supply/stop is selectable while communication is halted.
Transmit data (SSITXD0/SSIDATA1) and receive data (SSIRXD0/SSIDATA1)	Transmission	• Muting method (transmission of transmit FIFO data or transmission of data fixed to 0) selectable
FIFO	Capacity	• Transmit FIFO/receive FIFO: 4 bytes × 32 stages
	Data alignment	• Data alignment method (left-justification or right-justification) selectable for the data transfer between FIFO and shift register
Interrupt	Interrupt output	• Communication error/idle mode • Receive data full • Transmit data empty
Low power consumption function		• Whether to supply the audio clock selectable in master mode
Module stop function		Module stop state can be set for each channels to reduce power consumption.
TrustZone Filter		Security and Privilege attribution can be set for each channels.

The following table lists and defines the terms used for the communication formats SSIE can use:

Table 39.2 Definition of terms (1 of 2)

Term	Definition
Start trigger	First edge of the signal on the SSILRCKn/SSIFSn (n = 0, 1) pin when the signal is set to the value specified in LRCKP to enable communication

Table 39.2 Definition of terms (2 of 2)

Term	Definition
Frame boundary	Point where SSIE starts transferring the first data of a frame or the point where SSIE ends transferring the last data of the frame
Frame word number	Number of sound channels per frame
System word length	Number of bits per channel
Data word length	Number of significant bits per channel
Control bits for communication formats	• SSICR register: FRM, DWL, SWL, LRCKP, SPDP, SDTA, PDTA, and DEL bits • SSIFCR register: BSW bit • SSIOFR register: OMOD bit • SSISCR register: TDES[4:0] and RDFS[4:0] bits

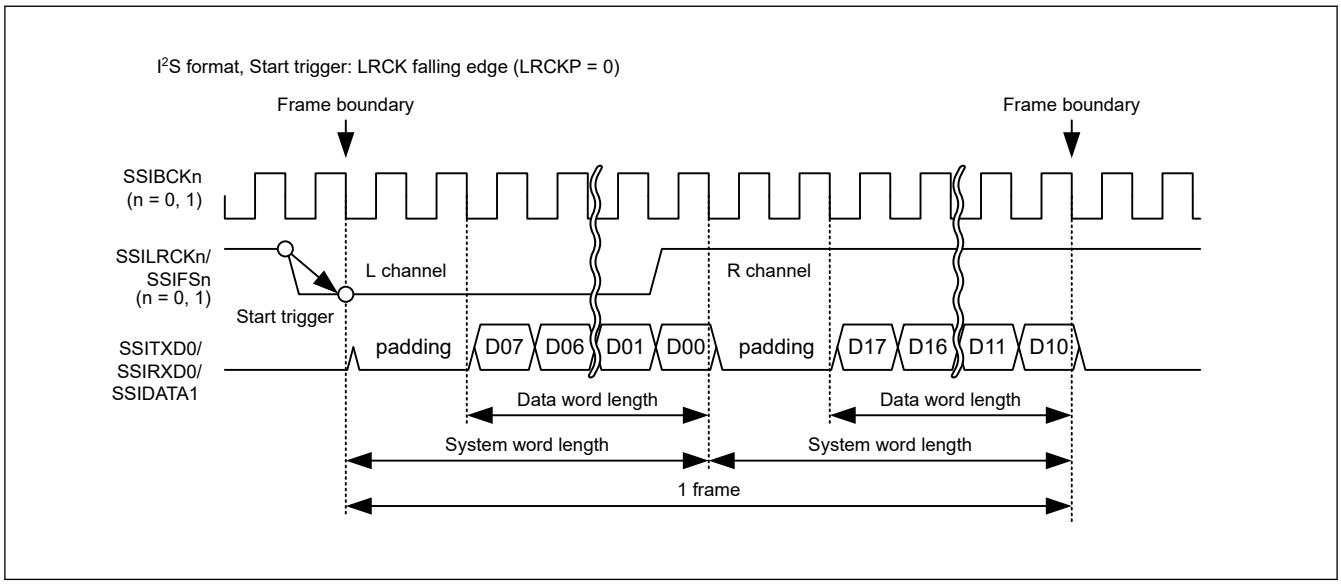


Figure 39.1 Definition of communication format

Figure 39.2 and Figure 39.3 shows a block diagram of SSIE.

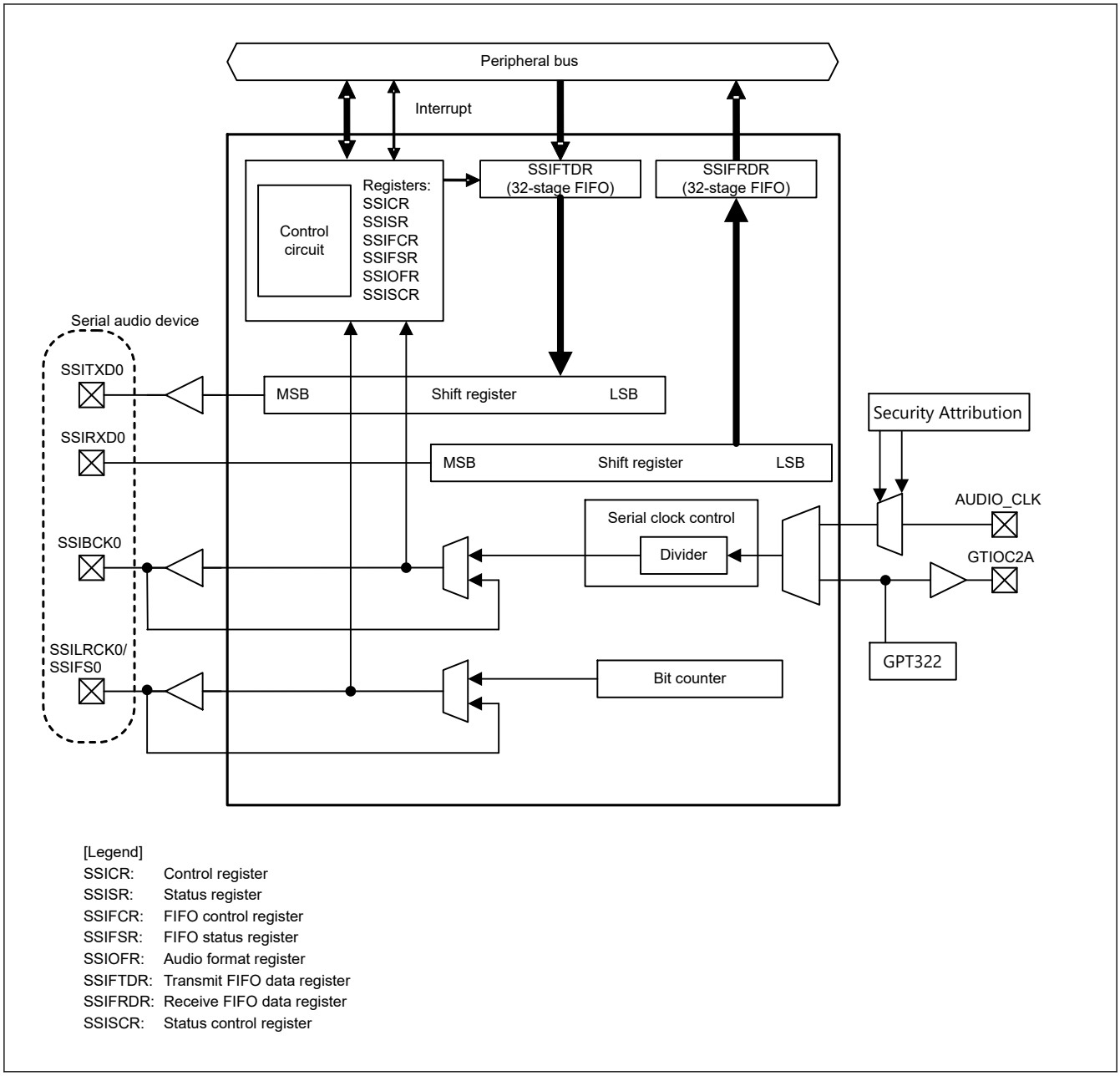


Figure 39.2 SSIE block diagram (SSIE0)

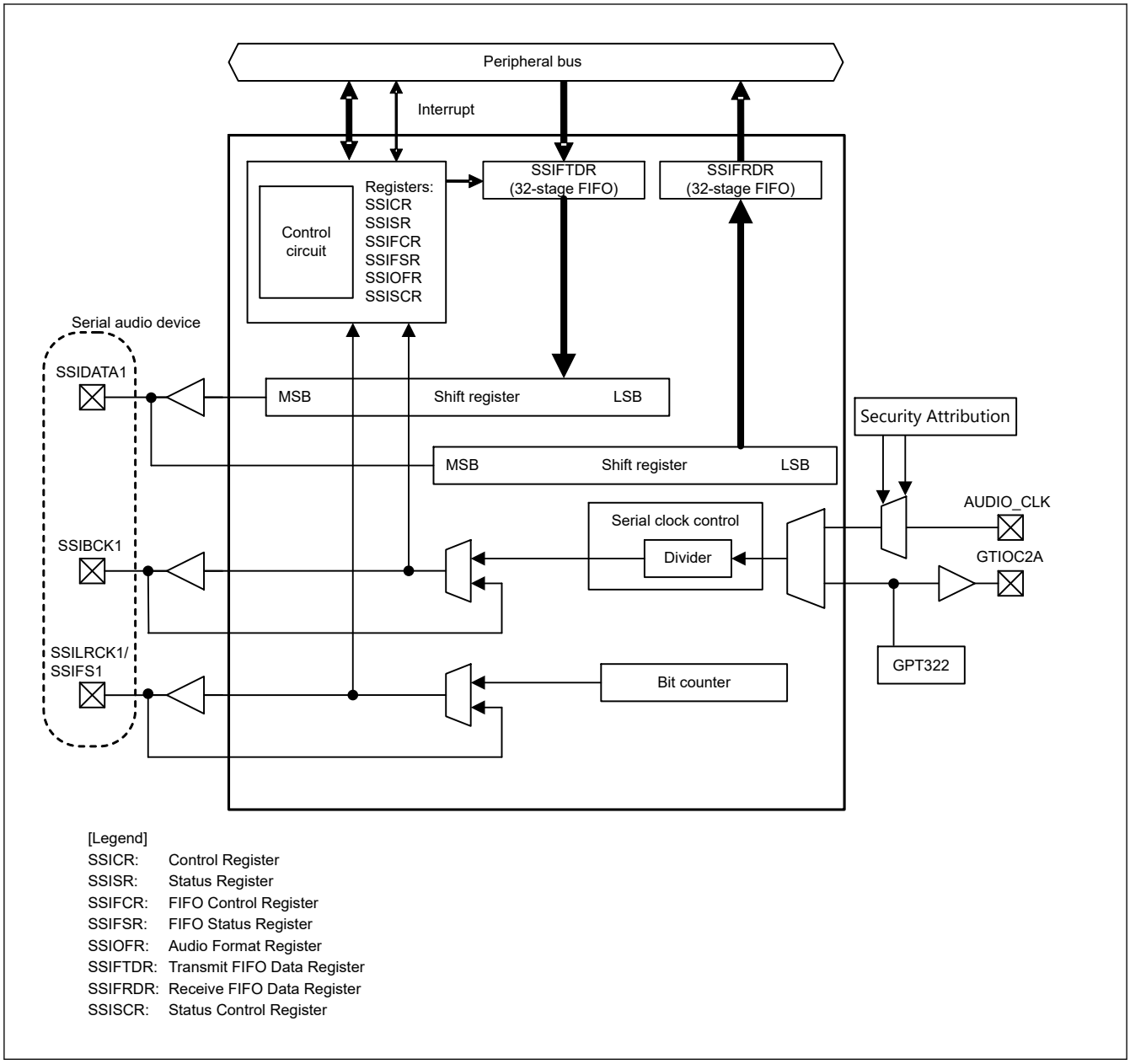


Figure 39.3 SSIE block diagram (SSIE1)

Figure 39.4 shows the clock configuration of SSIE.

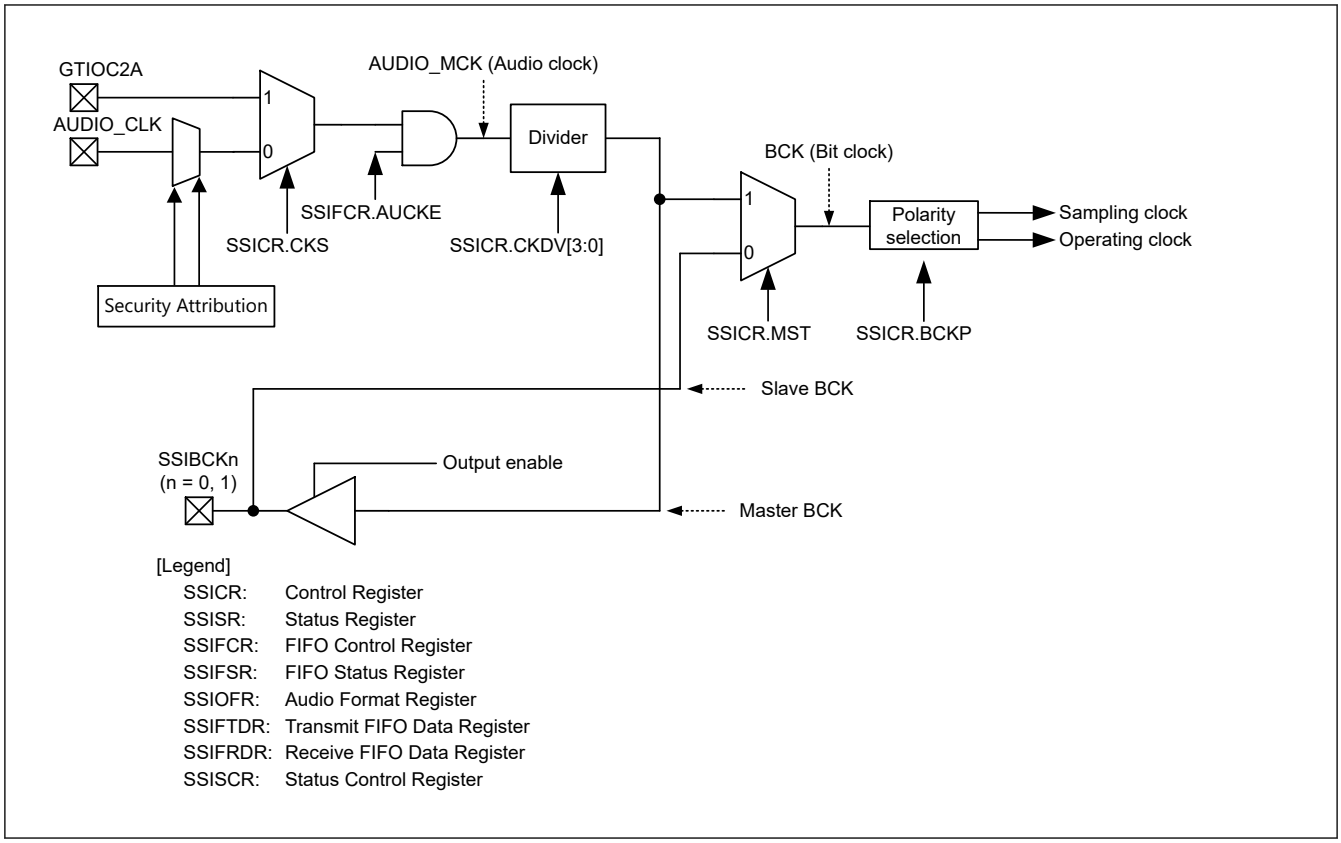


Figure 39.4 SSIE clock configuration

The AUDIO_CLK can be input according to the security attribution of Port and SSIE and SSICR.CKS bit setting as shown in the Table 39.3.

Table 39.3 The input condition of AUDIO_CLK by the security setting

PORT Security Attribution (PmSAR (m = 0 - 9, A, B))	SSIE0 Security Attribution (PSARm (B - E))	SSIE0's SSICR.CKS	SSIE1 Security Attribution (PSARm (B - E))	SSIE1's SSICR.CKS	AUDIO_CLK pin input
0 (secure)	0 (secure)	0 (AUDIO_CLK)	Don't care		AUDIO_CLK input enable
	Don't care		0 (secure)	0 (AUDIO_CLK)	
1 (non-secure)	1 (non-secure)	Don't care	0 (secure)	1 (GTIOC2A)	
			1 (non-secure)	Don't care	
	0 (secure)	1 (GTIOC2A)	1 (non-secure)	Don't care	
	1 (non-secure)	Don't care			
other than above					AUDIO_CLK input disable

Table 39.4 lists the I/O pins.

Table 39.4 SSIE I/O pins

Channel	Pin name	I/O	Function
Common	SSIBLKn	Input	Serial bit clock pins
	AUDIO_CLK	Input	External clock pin for audio (input oversampling clock)
	SSILRCKn/SSIFSn	Output	LR clock/frame synchronization pins
SSIE0	SSIRXD0	Input	Serial data input pin
	SSITXD0	Output	Serial data output pin
SSIE1	SSIDATA1	I/O	Serial data input/output pin

Note: n = 0, 1

39.2 Register Descriptions

39.2.1 SSICR : Control Register

Base address: SSIEn = 0x4025_D000 + 0x0100 × n (n = 0, 1)
SSIEn_NS = 0x5025_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	FRM[1:0]		DWL[2:0]			SWL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MST	BCKP	LRCKP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	REN	Reception Enable*2 0: Disables reception 1: Enables reception (starts reception)	R/W
1	TEN	Transmission Enable*2 0: Disables transmission 1: Enables transmission (starts transmission)	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	MUEN	Mute Enable 0: Disables muting on the next frame boundary 1: Enables muting on the next frame boundary	R/W
7:4	CKDV[3:0]	Selects Bit Clock Division Ratio*1 0x0: AUDIO_MCK 0x1: AUDIO_MCK/2 0x2: AUDIO_MCK/4 0x3: AUDIO_MCK/8 0x4: AUDIO_MCK/16 0x5: AUDIO_MCK/32 0x6: AUDIO_MCK/64 0x7: AUDIO_MCK/128 0x8: AUDIO_MCK/6 0x9: AUDIO_MCK/12 0xA: AUDIO_MCK/24 0xB: AUDIO_MCK/48 0xC: AUDIO_MCK/96 Others: Setting prohibited	R/W
8	DEL	Selects Serial Data Delay*1 In the monaural format, this bit controls the waveform of SSILRCKn/SSIFSn (n = 0, 1). For details, see section 39.3.2. Monaural Format. 0: Delay of 1 cycle of SSIBCKn (n = 0, 1) between SSILRCKn/SSIFSn (n = 0, 1) and SSITXD0/SSIRXD0/SSIDATA1 1: No delay between SSILRCKn/SSIFSn (n = 0, 1) and SSITXD0/SSIRXD0/ SSIDATA1	R/W
9	PDTA	Selects Placement Data Alignment*1 0: Left-justifies placement data (SSIFTDR, SSIFRDR) 1: Right-justifies placement data (SSIFTDR, SSIFRDR)	R/W
10	SDTA	Selects Serial Data Alignment*1 0: Transmits and receives serial data first and then padding bits 1: Transmit and receives padding bits first and then serial data	R/W

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>11</td><td>SPDP</td><td>Selects Serial Padding Polarity*1
0: Padding data is at a low level
1: Padding data is at a high level</td><td>R/W</td></tr>
<tr><td>12</td><td>LRCKP</td><td>Selects the Initial Value and Polarity of LR Clock/Frame Synchronization Signal*1
0: The initial value is at a high level.
The start trigger for a frame is synchronized with a falling edge of SSILRCKn/SSIFSn (n = 0, 1).
1: The initial value is at a low level.
The start trigger for a frame is synchronized with a rising edge of SSILRCKn/SSIFSn (n = 0, 1).</td><td>R/W</td></tr>
<tr><td>13</td><td>BCKP</td><td>Selects Bit Clock Polarity*1
0: SSILRCKn/SSIFSn (n = 0, 1) and SSITXD0/SSIRXD0/SSIDATA1 change at a falling edge (SSILRCKn/SSIFSn (n = 0, 1) and SSIRXD0/SSIDATA1 are sampled at a rising edge of SSIBCKn (n = 0, 1)).
1: SSILRCKn/SSIFSn (n = 0, 1) and SSITXD0/SSIRXD0/SSIDATA1 change at a rising edge (SSILRCKn/SSIFSn (n = 0, 1) and SSIRXD0/SSIDATA1 are sampled at a falling edge of SSIBCKn (n = 0, 1)).</td><td>R/W</td></tr>
<tr><td>14</td><td>MST</td><td>Master Enable*1
0: Slave-mode communication
1: Master-mode communication</td><td>R/W</td></tr>
<tr><td>15</td><td>—</td><td>This bit is read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>18:16</td><td>SWL[2:0]</td><td>Selects System Word Length*1
0 0 0: 8 bits
0 0 1: 16 bits
0 1 0: 24 bits
0 1 1: 32 bits
1 0 0: 48 bits
1 0 1: 64 bits
1 1 0: 128 bits
1 1 1: 256 bits</td><td>R/W</td></tr>
<tr><td>21:19</td><td>DWL[2:0]</td><td>Selects Data Word Length*1
0 0 0: 8 bits
0 0 1: 16 bits
0 1 0: 18 bits
0 1 1: 20 bits
1 0 0: 22 bits
1 0 1: 24 bits
1 1 0: 32 bits
1 1 1: Setting prohibited</td><td>R/W</td></tr>
<tr><td>23:22</td><td>FRM[1:0]</td><td>Selects Frame Word Number*1
<table>
<tr><th></th><th colspan="3">Communication format (SSIOFR.OMOD[1:0])</th></tr>
<tr><td>FRM[1:0]</td><td>I²S (00b)</td><td>Monaural (10b)</td><td>TDM (01b)</td></tr>
<tr><td>00b</td><td>2</td><td>1</td><td>Setting prohibited</td></tr>
<tr><td>01b</td><td>Setting prohibited</td><td>Setting prohibited</td><td>4</td></tr>
<tr><td>10b</td><td rowspan="2"></td><td rowspan="2"></td><td>6</td></tr>
<tr><td>11b</td><td>8</td></tr>
</table></td><td>R/W</td></tr>
<tr><td>24</td><td>—</td><td>This bit is read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>25</td><td>IIEN</td><td>Idle Mode Interrupt Output Enable
0: Disables idle mode interrupt output
1: Enables idle mode interrupt output</td><td>R/W</td></tr>
<tr><td>26</td><td>ROIEN</td><td>Receive Overflow Interrupt Output Enable
0: Disables receive overflow interrupt output
1: Enables receive overflow interrupt output</td><td>R/W</td></tr>
</table>

Bit	Symbol	Function	R/W
27	RUIEN	Receive Underflow Interrupt Output Enable 0: Disables receive underflow interrupt output 1: Enables receive underflow interrupt output	R/W
28	TOIEN	Transmit Overflow Interrupt Output Enable 0: Disables transmit overflow interrupt output 1: Enables transmit overflow interrupt output	R/W
29	TUIEN	Transmit Underflow Interrupt Output Enable 0: Disables transmit underflow interrupt output 1: Enables transmit underflow interrupt output	R/W
30	CKS	Selects an Audio Clock for Master-mode Communication*1 0: Selects the AUDIO_CLK input 1: Selects the GTIOC2A (GPT output)	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Writing to these bits while SSIE is in a communication state (SSISR.IIRQ = 0) is prohibited. If the value of these bits is changed by rewriting, subsequent operation is unpredictable.

Note 2. If the TEN bit or REN bit is rewritten, make sure that the SSISR.IIRQ bit is in the desired status. If the value of the TEN or REN bit is changed by rewriting, subsequent operation is unpredictable. For example, when transmission or reception is enabled, check that SSISR.IIRQ is 0; when transmission or reception is disabled, check that SSISR.IIRQ is 1.

With this register, select an audio clock, control interrupt requests, select data formats, and set an operation mode.

TEN and REN bits (Transmission and Reception Enable)

The TEN and REN bits enable/disable transmission and reception. When 1 is written to one of these bits, the corresponding communication operation starts in synchronization with a start trigger by the SSILRCKn/SSIFSn (n = 0, 1) signal. For details, see section 39.6.2. Transmission to section 39.6.4. Transmission and Reception. When 0 is written to this bit, the current communication operation stops at the next frame boundary. To use SSIE for both transmission and reception, always write 1 to these bits together. When stopping the communication using SSIE, always disable both transmission and reception (write 0 to the TEN and REN bits).

If you want to stop SSIE before a frame boundary is reached, perform a software reset procedure.

MUEN bit (Mute Enable)

The MUEN bit sets/clears the mute function for the data output from the SSITXD0/SSIDATA1 pin. When this bit is set to 1 in the middle of a frame, the SSITXD0/SSIDATA1 output changes to 0 at the next frame boundary. When this bit is set to 0 in the middle of a frame, the SSITXD0/SSIDATA1 output changes to the data of transmit FIFO data register at the next frame boundary. Note that this bit controls data only. Status flags and interrupt signals are normally generated.

Changing the value of this bit must be performed only after setting the communication format to be used.

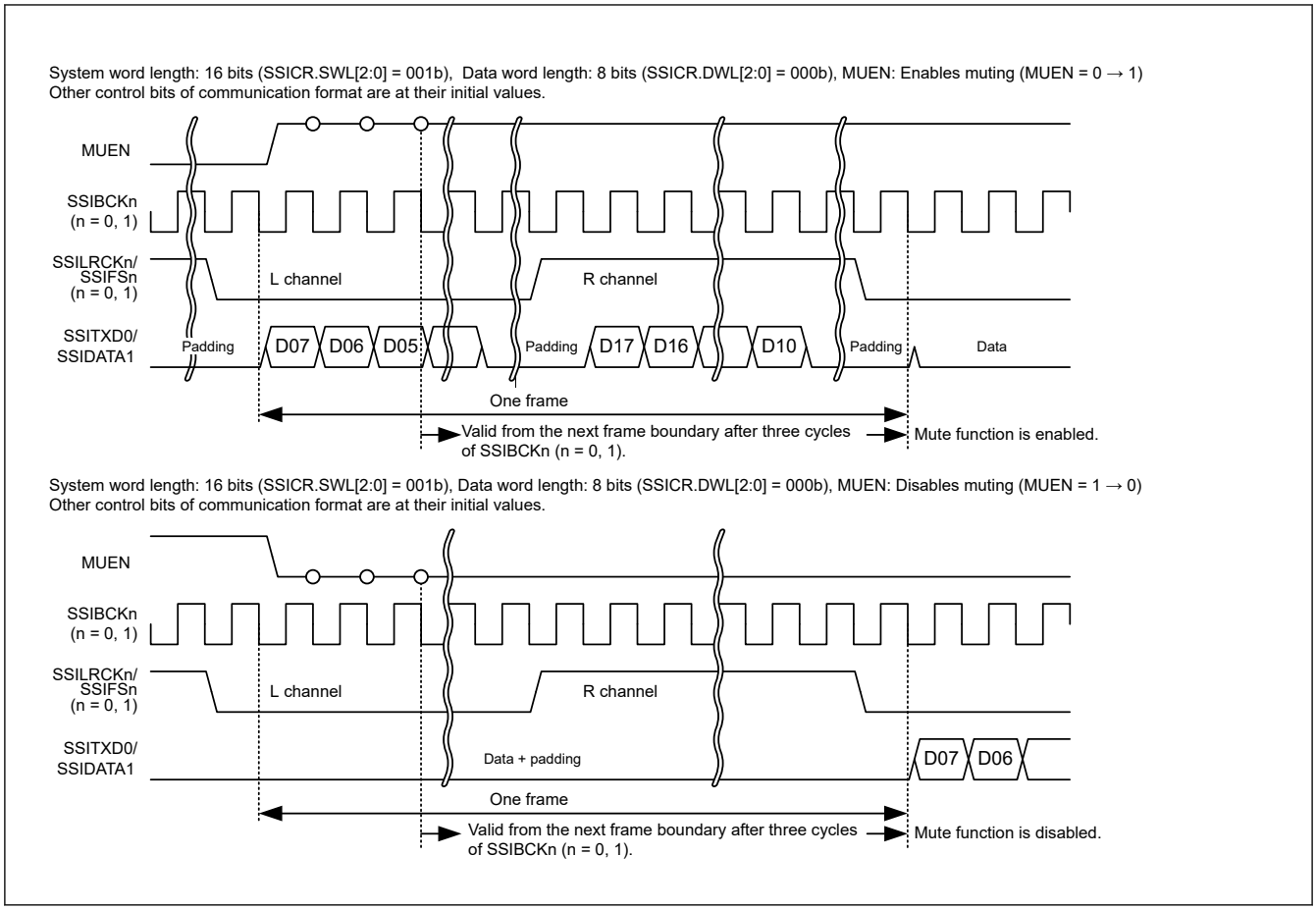


Figure 39.5 Transmit data with the mute function set

CKDV[3:0] bits (Selects Bit Clock Division Ratio)

The CKDV[3:0] bits set the division ratio of the bit clock based on AUDIO_MCK in master-mode communication (MST= 1). In slave-mode communication (MST = 0), setting of these bits are invalid.

Writing to this bit must be performed when the supply of AUDIO_MCK is stopped. For details about the timing, see the detailed description of the AUCKE bit in section 39.2.3. SSIFCR : FIFO Control Register.

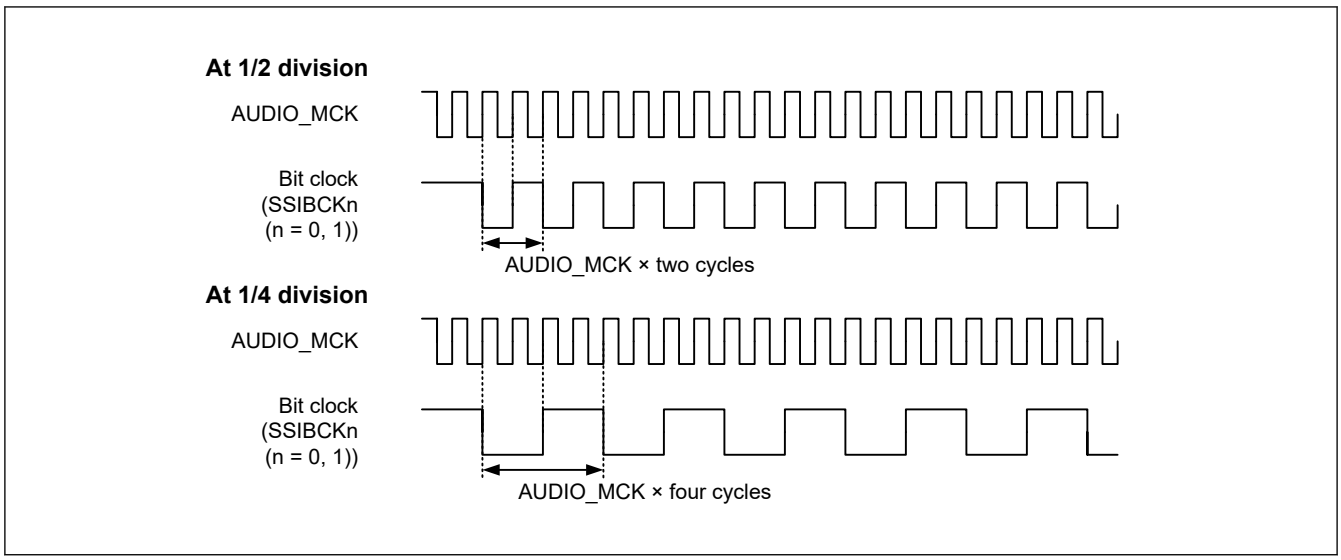


Figure 39.6 Sampling frequencies in master-mode communication

DEL bit (Selects Serial Data Delay)

The DEL bit sets whether or not there will be a delay between SSILRCKn/SSIFSn (n = 0, 1) and SSITXD0/SSIRXD0/SSIDATA1.

For the I^2S or TDM format, set the DEL bit to 0. When the monaural format is used, setting of this bit changes the high period width of SSILRCKn/SSIFSn (n = 0, 1). For details, see section 39.3.2. Monaural Format. When using a compatible communication format, specify a setting of this bit that enables communication.

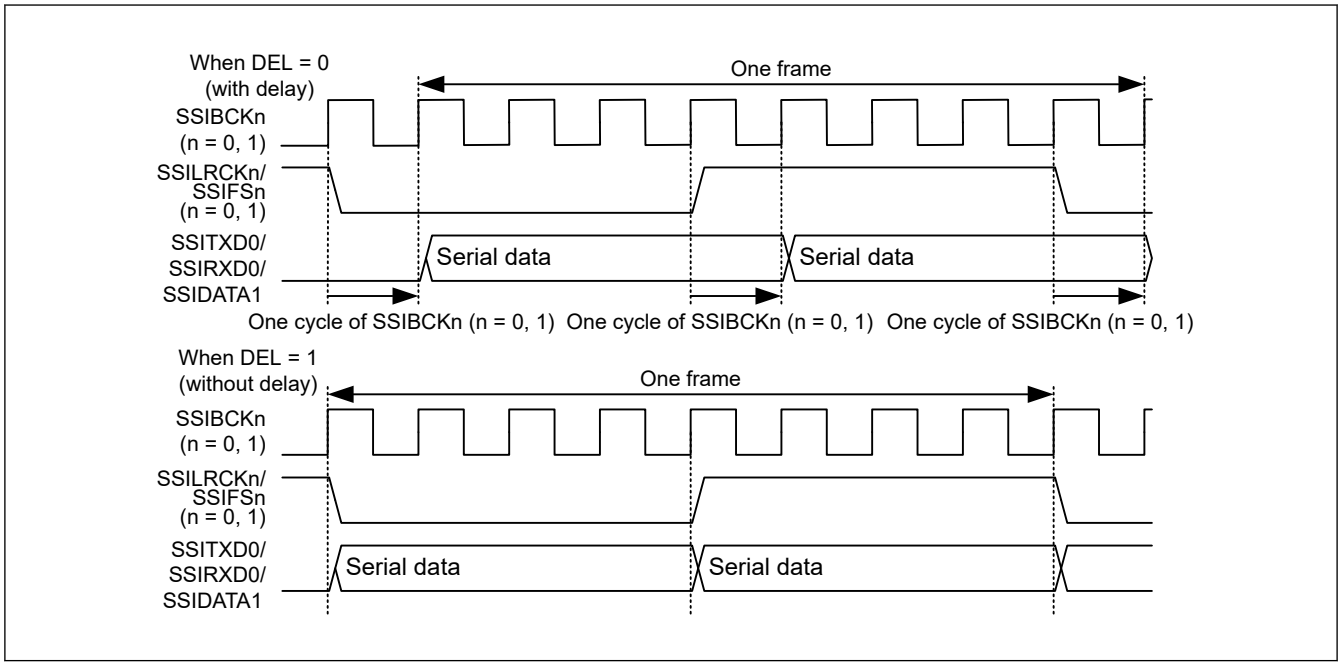


Figure 39.7 Setting of delay in serial data

PDTA bit (Selects Placement Data Alignment)

The PDTA bit sets how to align placement data. With the setting of data word length as 32 bits (SSICR.DWL[2:0] = 110b), this bit is invalid.

At transmission, see Figure 39.8.

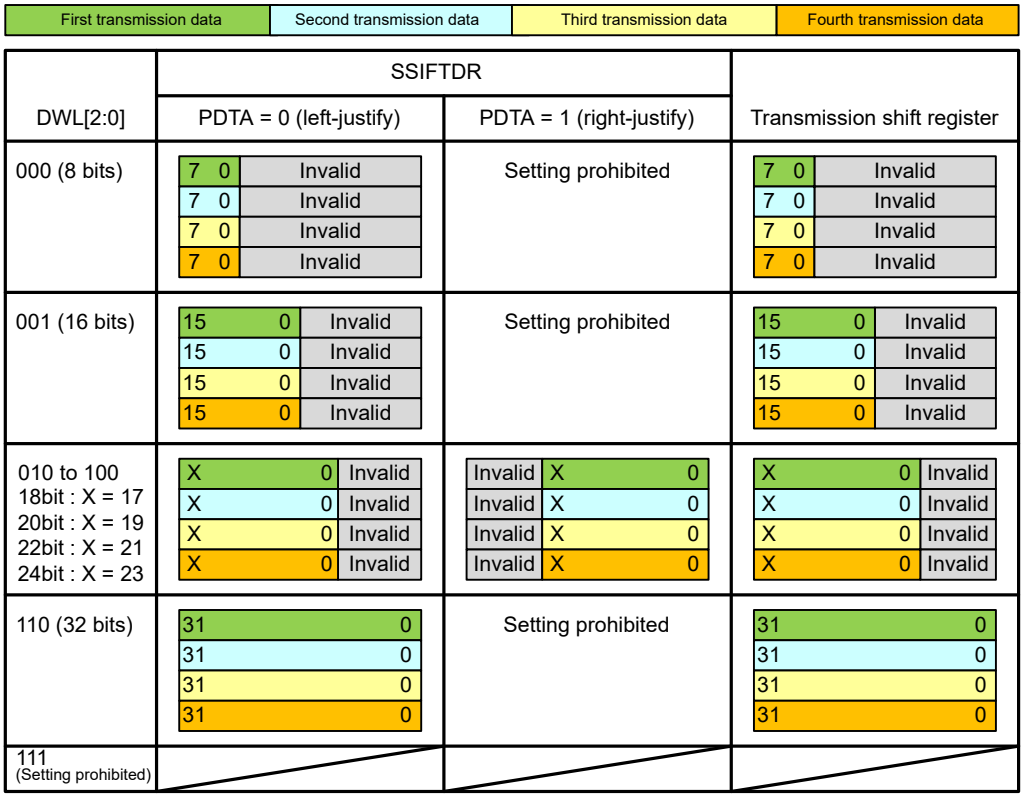


Figure 39.8 Alignment of placement data at transmission

At reception, see Figure 39.9.

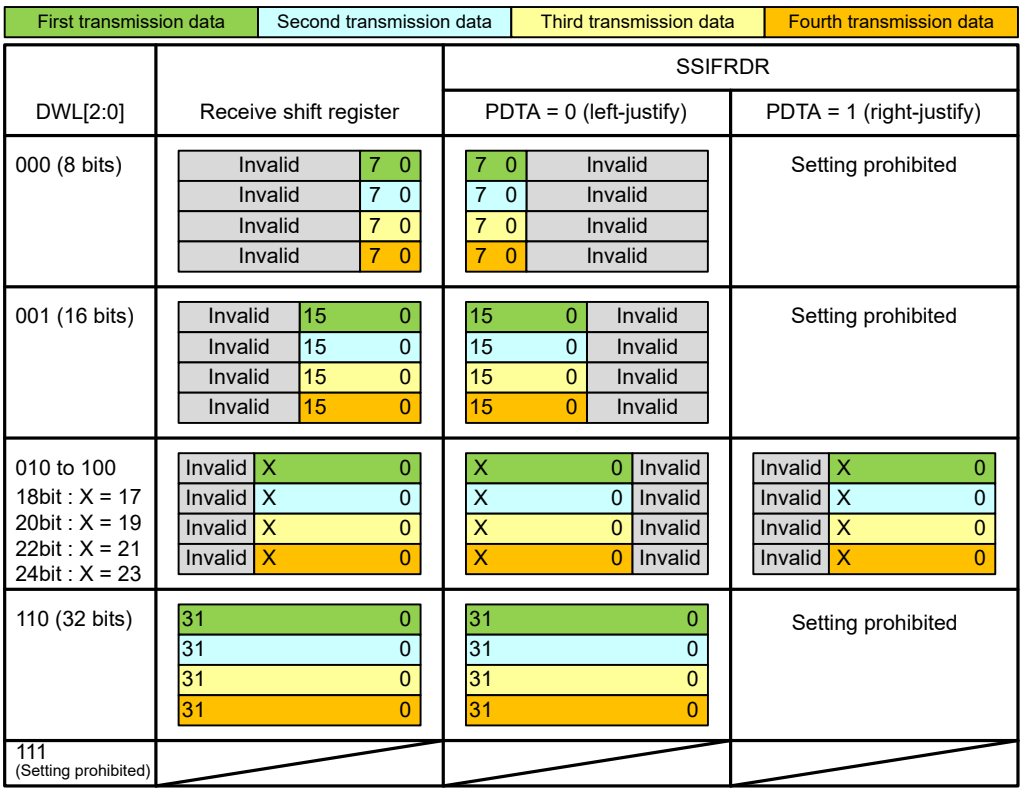


Figure 39.9 Alignment of placement data at reception

SDTA bit (Selects Serial Data Delay)

The SDTA bit sets how to align serial data and padding bits. For communication without padding bits, this bit is invalid.

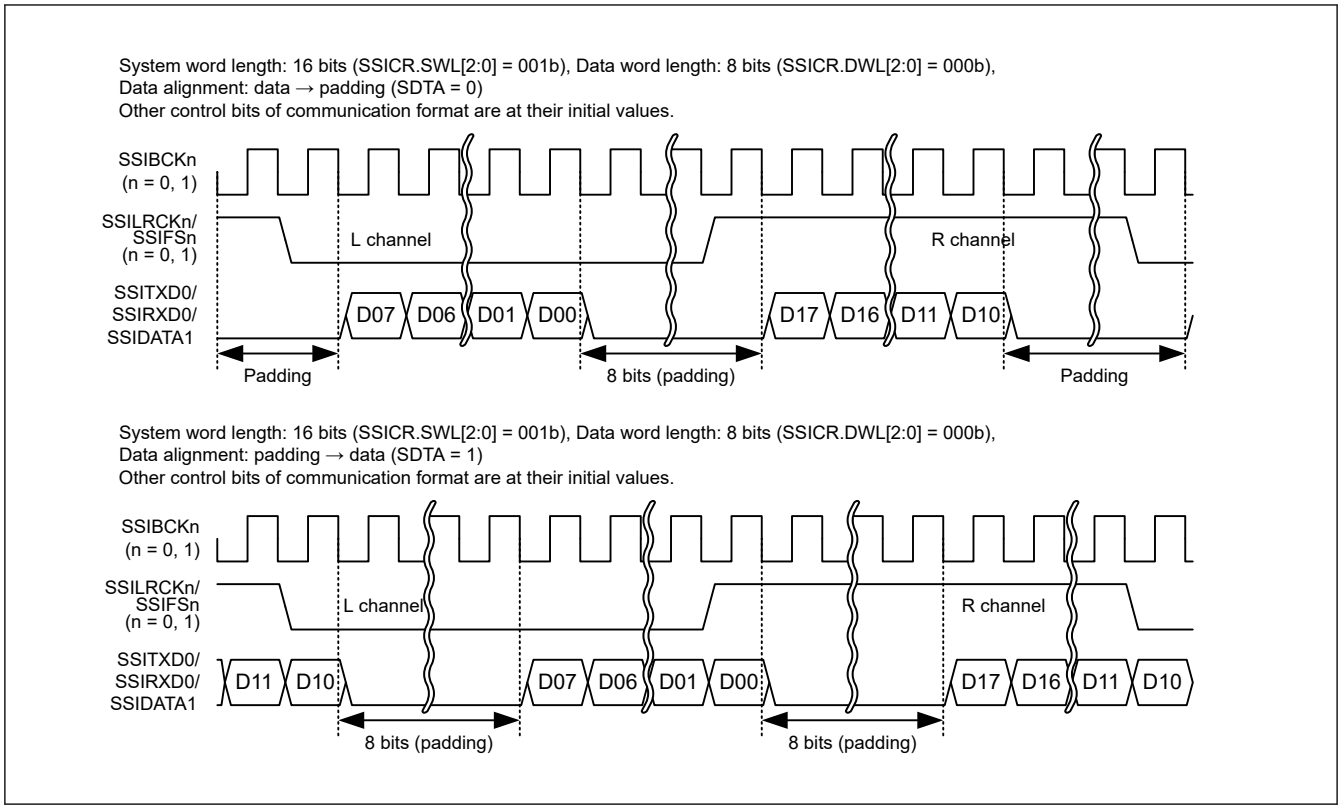


Figure 39.10 Alignment setting of serial data with padding bits

SPDP bit (Selects Serial Padding Polarity)

The SPDP bit sets polarity of padding bits.

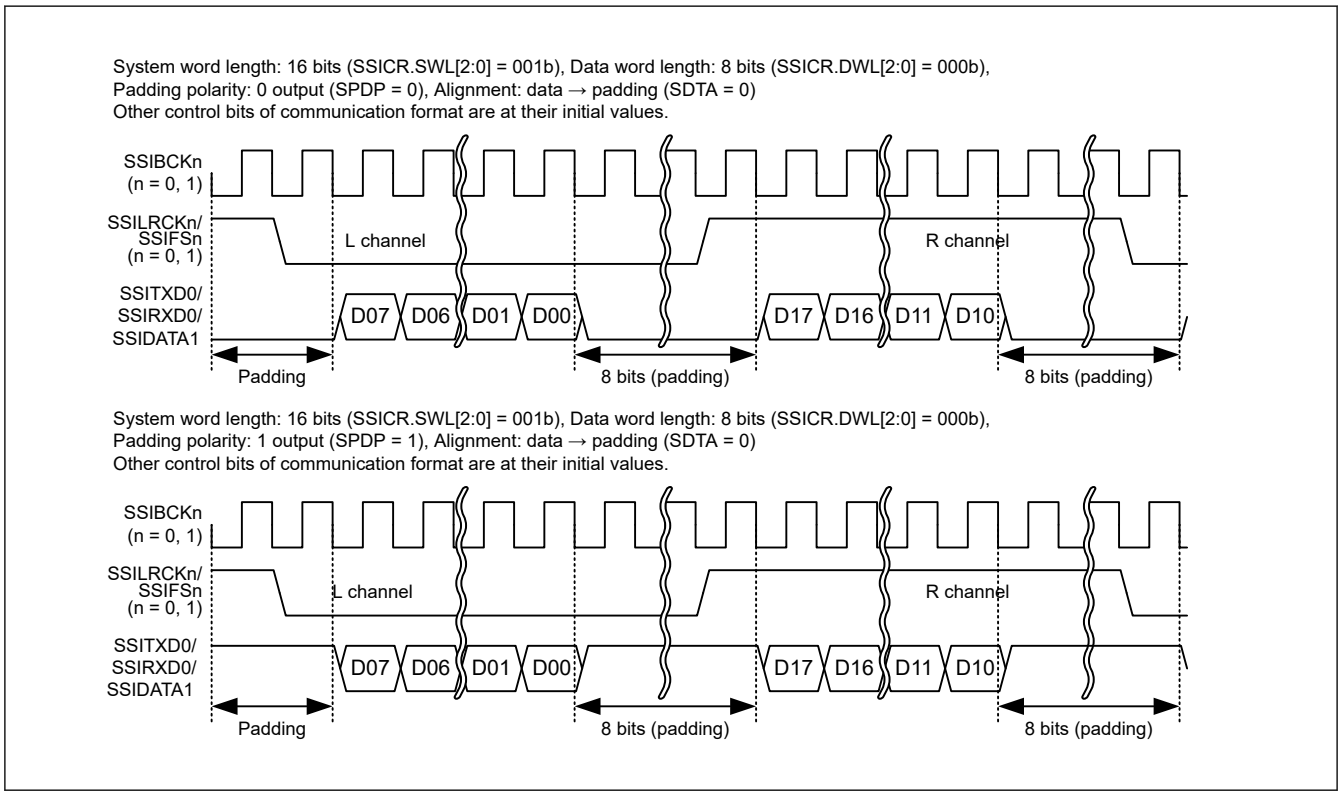


Figure 39.11 Padding bit polarity

LRCKP bit (Selects the Initial Value and Polarity of LR Clock/Frame Synchronization Signal)

The LRCKP bit sets the initial value and polarity of SSILRCKn/SSIFSn (n = 0, 1). Set this bit according to the communication format to be used in SSIE. See Table 39.5 Initial output value and polarity of SSILRCKn/SSIFSn (n = 0, 1) pin. For the slave-mode communication (MST = 0), only the start trigger is used.

Writing to these bits must be performed when the LR clock supply to the SSILRCKn/SSIFSn (n = 0, 1) pin is stopped. For details about the output of LR clock, see the detailed description of the LRCONT bit in section 39.2.7. SSIOFR : Audio Format Register.

Table 39.5 Initial output value and polarity of SSILRCKn/SSIFSn (n = 0, 1) pin

Communication Format	Expected Initial State	Setting Value of LRCKP
I^2S	High	0
Monaural	Low	1
TDM	Low	1

Note: When the format to be used is compatible with the I^2S, monaural, or TDM format, specify settings to enable communication with the respective formats.

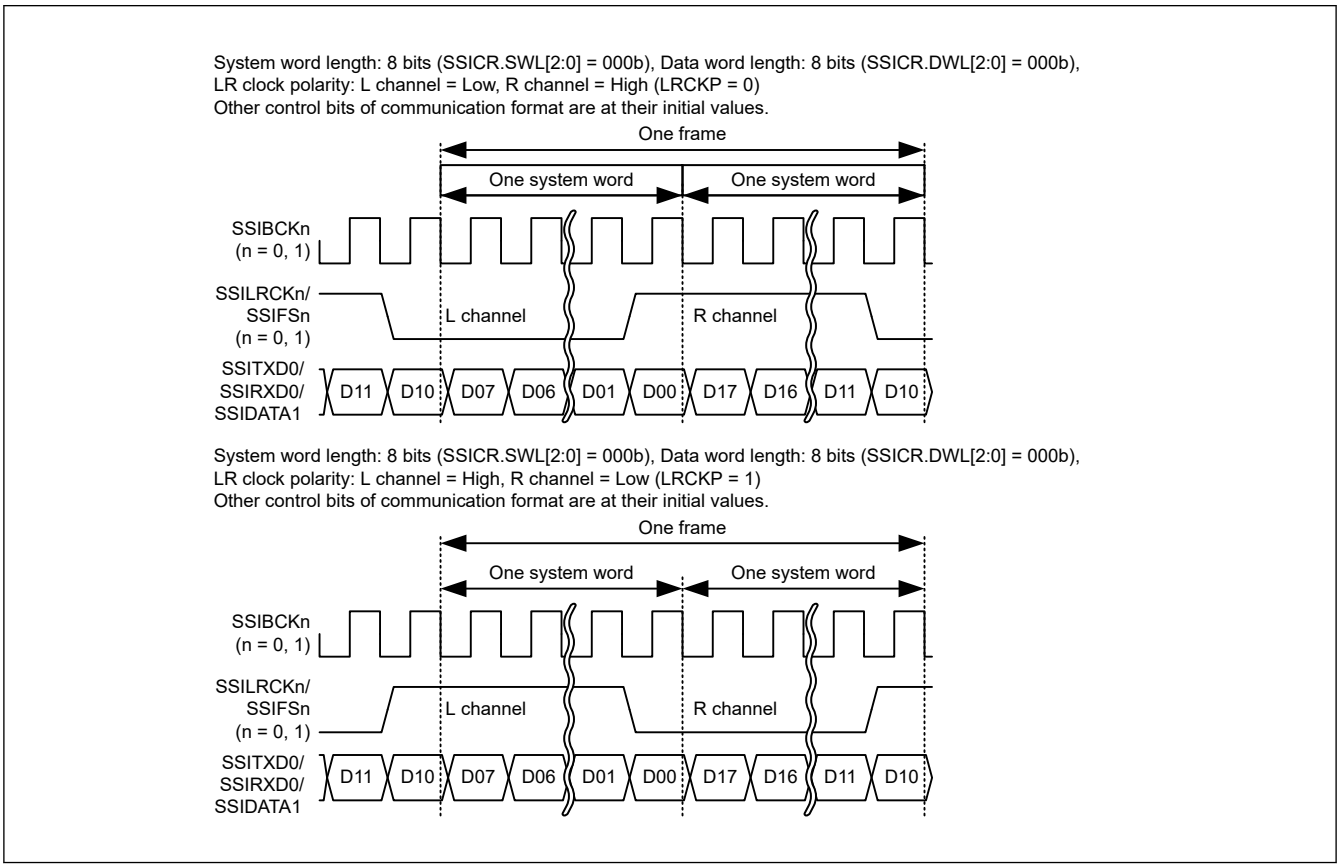


Figure 39.12 LR clock/frame synchronization polarity setting

BCKP bit (Selects Bit Clock Polarity)

The BCKP bit sets the bit clock polarity.

Writing to this bit must be performed when the supply of AUDIO_MCK is stopped. For details about the timing, see the detailed description of the AUCKE bit in section 39.2.3. SSIFCR : FIFO Control Register.

Table 39.6 Bit clock polarity

Communication	Master/Slave	Timing	BCKP = 0	BCKP = 1
Reception	Slave	At SSILRCKn/SSIFSn (n = 0, 1) sampling	SSIBCKn (n = 0, 1) rising edge	SSIBCKn (n = 0, 1) falling edge
	Master/slave	At SSIRXD0/SSIDATA1 sampling	SSIBCKn (n = 0, 1) rising edge	SSIBCKn (n = 0, 1) falling edge
Transmission	Master	At change of SSILRCKn/SSIFSn (n = 0, 1) output	SSIBCKn (n = 0, 1) falling edge	SSIBCKn (n = 0, 1) rising edge
	Master/slave	At change of SSITXD0/SSIDATA1 output	SSIBCKn (n = 0, 1) falling edge	SSIBCKn (n = 0, 1) rising edge

MST bit (Master Enable)

The MST bit sets master-/slave-mode communication.

Writing to this bit must be performed when the supply of AUDIO_MCK is stopped. For details about the timing, see the detailed description of the AUCKE bit in section 39.2.3. SSIFCR : FIFO Control Register.

SWL[2:0] bits (Selects System Word Length)

The SWL[2:0] bits set the number of bits in one system word. Padding bits are sent and received in relation with one data word set with DWL[2:0]. See Table 39.13 for details.

Writing to these bits must be performed when the LR clock supply to the SSILRCKn/SSIFSn (n = 0, 1) pin is stopped. For details about the output of LR clock, see the detailed description of the LRCONT bit in section 39.2.7. SSIOFR : Audio Format Register.

DWL[2:0] bits (Selects Data Word Length)

The DWL[2:0] bits set the number of bits in one data word. The data word length (number of bits per data word) must not exceed the system word length (number of bits per system word). For details, see Table 39.13.

FRM[1:0] bits (Selects Frame Word Number)

The FRM[1:0] bits set the frame word number in individual communication formats.

Writing to these bits must be performed when the LR clock supply to the SSILRCKn/SSIFSn (n = 0, 1) pin is stopped. For details about the output of LR clock, see the detailed description of the LRCONT bit in section 39.2.7. SSIOFR : Audio Format Register.

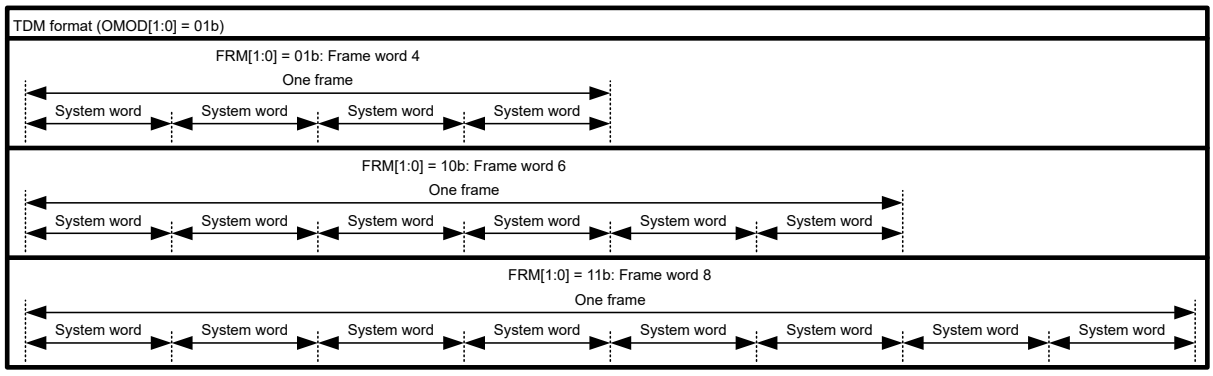


Figure 39.13 Frame word number

IIEN bit (Idle Mode Interrupt Output Enable)

The IIEN bit enables/disables output of idle mode interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.IIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.IIRQ = 1.

ROIEN bit (Receive Overflow Interrupt Output Enable)

The ROIEN bit enables/disables output of receive overflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.ROIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.ROIRQ = 1.

RUIEN bit (Receive Underflow Interrupt Output Enable)

The RUIEN bit enables/disables output of receive underflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.RUIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.RUIRQ = 1.

TOIEN bit (Transmit Overflow Interrupt Output Enable)

The TOIEN bit enables/disables output of transmit overflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.TOIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.TOIRQ = 1.

TUIEN bit (Transmit Underflow Interrupt Output Enable)

The TUIEN bit enables/disables output of transmit underflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.TUIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.TUIRQ = 1.

CKS bit (Selects an Audio Clock for Master-mode Communication)

The CKS bit sets the audio clock in master-mode communication (MST = 1). In slave-mode communication (MST = 0), setting of this bit is invalid.

Writing to this bit must be performed when the supply of AUDIO_MCK is stopped. For details about the timing, see the detailed description of the AUCKE bit in section 39.2.3. SSIFCR : FIFO Control Register.

39.2.2 SSISR : Status Register

Base address: SSIEn = 0x4025_D000 + 0x0100 × n (n = 0, 1)
SSIEn_NS = 0x5025_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
24:0	—	These bits are read as 0. The write value should be 0.	R/W
25	IIRQ	Idle Mode Status Flag 0: In the communication state 1: In the idle state	R
26	ROIRQ	Receive Overflow Error Status Flag 0: No receive overflow error is generated. 1: A receive overflow error is generated.	R/W
27	RUIRQ	Receive Underflow Error Status Flag 0: No receive underflow error is generated. 1: A receive underflow error is generated.	R/W
28	TOIRQ	Transmit Overflow Error Status Flag 0: No transmit overflow error is generated. 1: A transmit overflow error is generated.	R/W

Bit	Symbol	Function	R/W
29	TUIRQ	Transmit Underflow Error Status flag 0: No transmit underflow error is generated. 1: A transmit underflow error is generated.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register is configured with status flags that indicate SSIE operational state.

IIRQ flag (Idle Mode Status Flag)

The IIRQ flag is a status flag that indicates the idle state. It indicates whether SSIE is in the idle state or communication state.

For details, see Figure 39.14 and Figure 39.15.

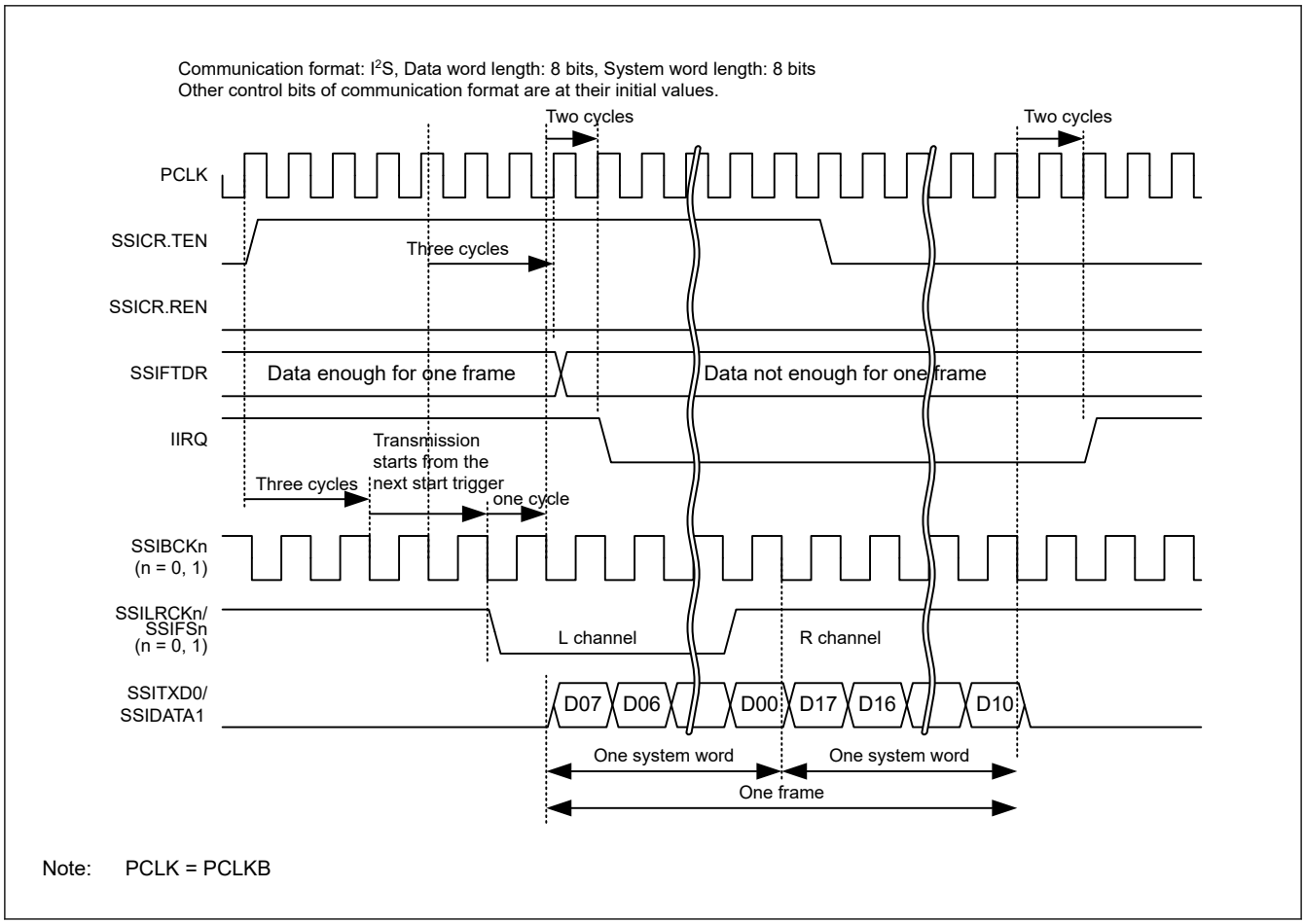


Figure 39.14 IIRQ setting timing (transmission)

- Transmitter (dedicated to transmission)

[Clearing condition]

While transmission was enabled (SSICR.TEN = 1 and SSICR.REN = 0), the transmit data for a transmission frame was written to the SSIFTDR register, and a start trigger was generated by the SSILRCKn/SSIFSn (n = 0, 1) signal.

[Clearing timing]

1 SSIBCKn (n = 0, 1) cycle + 2 PCLKB cycles after generation of the start trigger that is the clearing condition.

[Setting condition]

While transmission and reception were disabled (SSICR.TEN = 0 and SSICR.REN = 0), transmission of one frame was complete.

[Setting timing]

Two PCLKB cycles after the end of transmission (at a frame boundary) that is the setting condition.

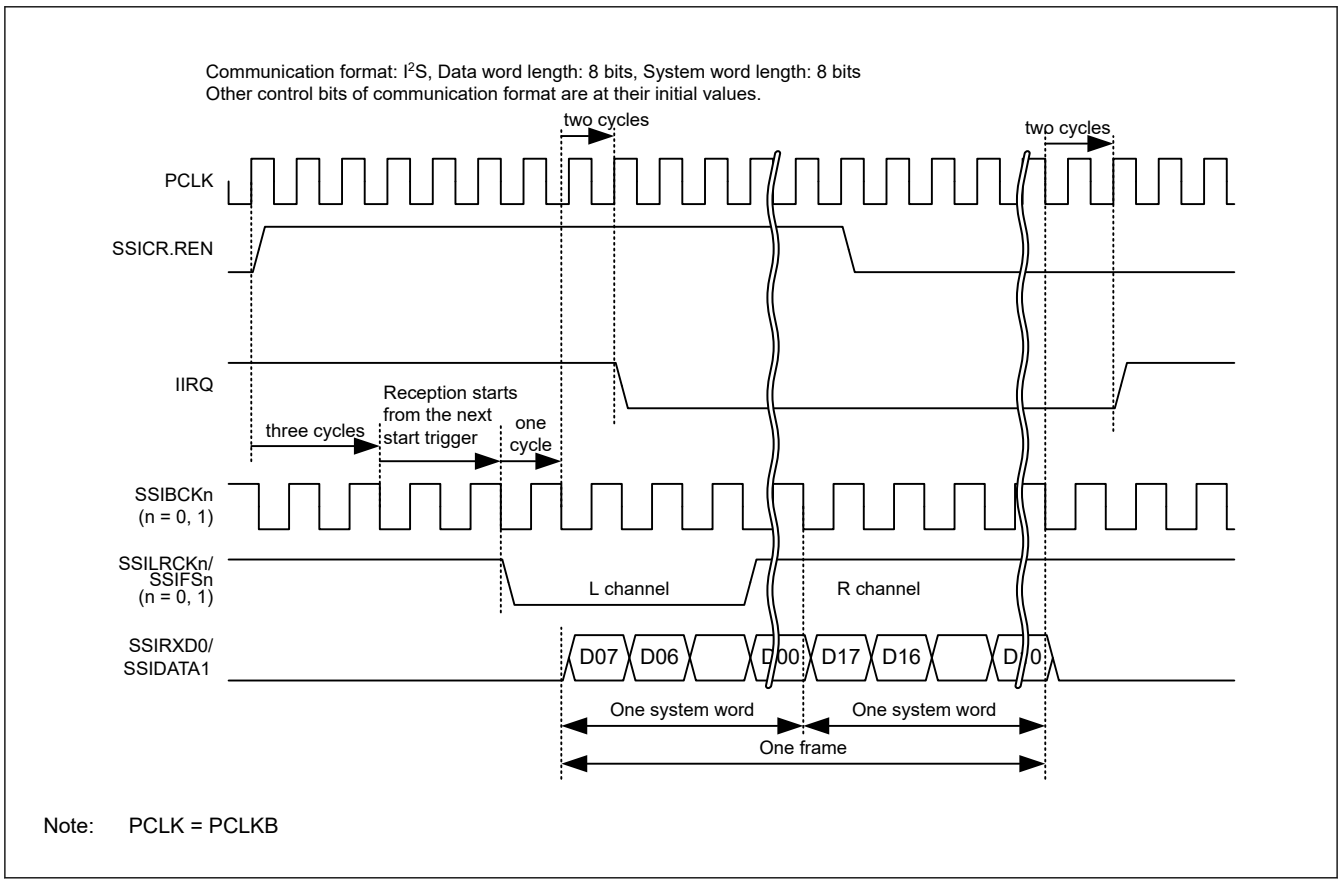


Figure 39.15 IIRQ setting timing (reception)

- Receiver (dedicated to reception)

[Clearing condition]

While reception was enabled (SSICR.TEN = 0 and SSICR.REN = 1, a start trigger was generated by the SSILRCKn/ SSIFSn (n = 0, 1) signal.

[Clearing timing]

1 SSIBCKn (n = 0, 1) cycle + 2 PCLKB cycles after generation of the start trigger that is the clearing condition.

[Setting condition]

While transmission and reception were disabled (SSICR.TEN = 0 and SSICR.REN = 0), reception of one frame was complete.

[Setting timing]

Two PCLKB cycles after the end of reception (at a frame boundary) that is the setting condition.

- Transceiver (transmission and reception)

[Clearing condition]

While transmission and reception were enabled (SSICR.TEN = 1 and SSICR.REN = 1), the transmit data for a transmission frame was written to the SSIFTDR register, and a start trigger is generated by the SSILRCKn/SSIFSn (n = 0, 1) signal.

[Clearing timing]

1 SSIBCKn (n = 0, 1) cycle + 2 PCLKB cycles after generation of the start trigger that is the clearing condition.

[Setting condition]

While transmission and reception were disabled (SSICR.TEN = 0 and SSICR.REN = 0), transmission of one frame was complete.

[Setting timing]

Two PCLKB cycles after the end of transmission (at a frame boundary) that is the setting condition.

ROIRQ flag (Receive Overflow Error Status Flag)

The ROIRQ flag is a status flag that indicates a receive overflow error. This flags is set by automatic determination but it must be cleared by register access. This flag indicates that received data is supplied at a higher rate than requested. Data is not transferred from the receive shift register to SSIFRDR where a receive overflow error is generated. For the procedure to recover from the overflow error, see section 39.6.6. Error Handling. This flag is not cleared by a receive FIFO data register reset (SSIFCR.RFRST).

[Priority order for setting and clearing]

Setting is prioritized.*1

[Clearing condition]

When either of the following operations is done:

1. Writing 0 to this bit after reading 1 from this bit*2
2. Enabling communication (changing SSICR.REN from 0 to 1).

[Clearing timing]

Clearing timing corresponding to the above clearing condition:

1. When 0 is written to this bit after reading 1 from this bit (same as the timing in Figure 39.19)
2. One PCLKB cycle after writing 1 to SSICR.REN.*3

[Setting condition]
At completion of receiving new data while SSIFRDR is full.

[Setting timing]
Three cycles of PCLKB after reception is completed.

Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following three conditions is met:
- A software reset (SSIFCR.SSIRST = 1) is done.
- After 1 has been read, writing of 0 is complete.
- One PCLKB cycle passes after 1 has been written to SSICR.REN.

Note 3. After communication is enabled (by changing the value of SSICR.REN bit from 0 to 1), the reception error flags (RUIRQ and ROIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the reception error flags might be unable to be read.

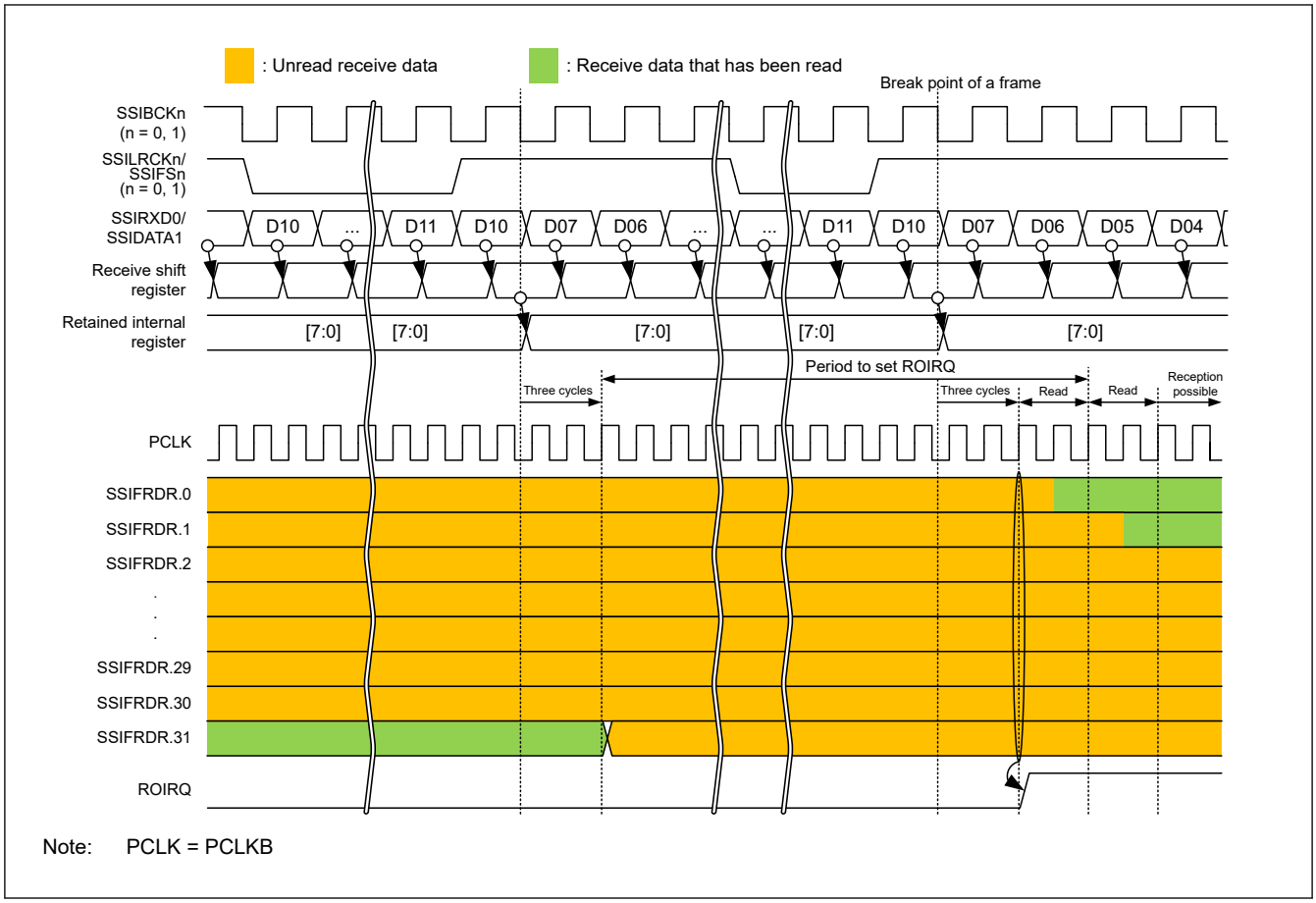


Figure 39.16 ROIRQ setting timing

RUIRQ flag (Receive Underflow Error Status Flag)

The RUIRQ flag is a status flag that indicates a receive underflow error. This flags is set by automatic determination but it must be cleared by register access. This flag indicates that SSIFRDR is read while it is empty. Data read from SSIFRDR where a receive underflow error is generated is invalid. See section 39.6.6. Error Handling for the error recovery procedure. This flag is not cleared by a receive FIFO data register reset (SSIFCR.RFRST). Note, however, that this flag is not set even if the SSIFRDR register is read while the receive FIFO data register is reset (by setting SSIFCR.RFRST to 1).

[Priority order for setting and clearing]

Setting is prioritized.*1

[Clearing condition]

When either of the following operations is done:

1. Writing 0 to this bit after reading 1 from this bit*2
2. Enabling communication (changing SSICR.REN from 0 to 1).

[Clearing timing]
Clearing timing corresponding to the above clearing condition

1. When 0 is written to this bit after reading 1 from this bit (same as the timing in Figure 39.19)
2. One PCLKB cycle after writing 1 to SSICR.REN.*3

[Setting condition]

Reading from SSIFRDR while it is empty.

[Setting timing]

At completion of reading from SSIFRDR. See Figure 39.17.

Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following three conditions is met:

- A software reset (SSIFCR.SSIRST = 1) is done.
- After 1 has been read, writing of 0 is complete.
- One PCLKB cycle passes after 1 has been written to SSICR.REN.

Note 3. After communication is enabled (by changing the value of SSICR.REN bit from 0 to 1), the reception error flags (RUIRQ and ROIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the reception error flags might be unable to be read.

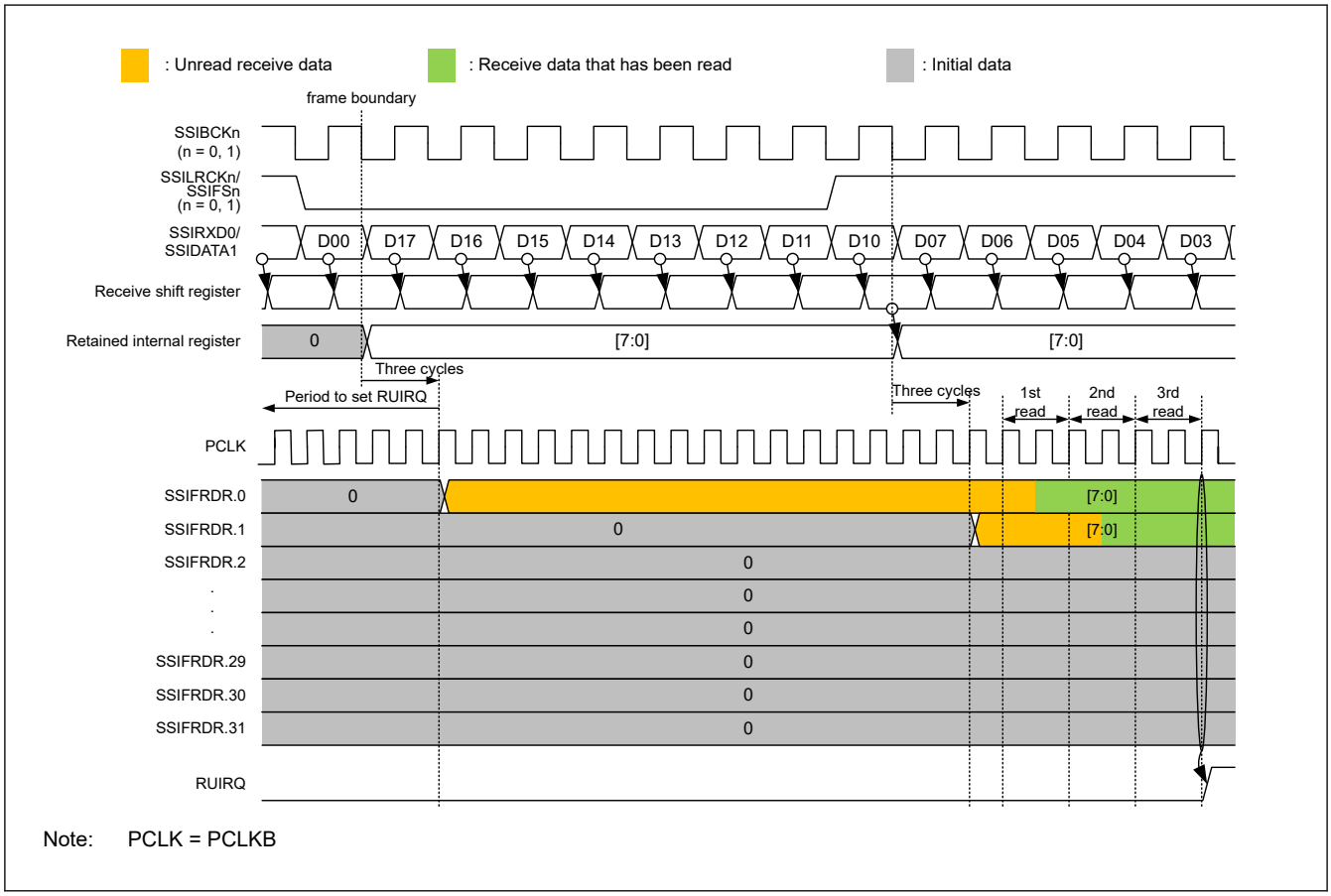


Figure 39.17 RUIRQ setting timing

TOIRQ flag (Transmit Overflow Error Status Flag)

The TOIRQ flag is a status flag that indicates a transmit overflow error. This flag is set by automatic determination but it must be cleared by register access. This flag indicates that an attempt has been made to write data to the SSIFTDR register when the register is full of data. The data writing that causes a transmit overflow is ignored. For the procedure to recover from the overflow error, see section 39.6.6. Error Handling. This flag is not cleared by a transmit FIFO data register reset (SSIFCR.TFRST).

[Priority order for setting and clearing]

Setting is prioritized.*1

[Clearing condition]

When either of the following operations is done:

1. Writing 0 to this bit after reading 1 from this bit*2
2. Enabling communication (changing SSICR.TEN from 0 to 1).

[Clearing timing]

Clearing timing corresponding to the above clearing condition

1. When 0 is written to this bit after reading 1 from this bit (same as the timing in Figure 39.19)
2. One PCLKB cycle after writing 1 to SSICR.TEN.*3

[Setting condition]

An attempt was made to write data to the SSIFTDR register when the register is full of data.

[Setting timing]

At completion of writing to SSIFTDR. For details, see Figure 39.18.

Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following three conditions is met:
- A software reset (SSIFCR.SSIRST = 1) is done.
- After 1 has been read, writing of 0 is complete.
- One PCLKB cycle passes after 1 has been written to SSICR.TEN.

Note 3. After communication is enabled (by changing the value of SSICR.TEN bit from 0 to 1), the transmission error flags (TOIRQ and TUIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the transmission error flags might be unable to be read.

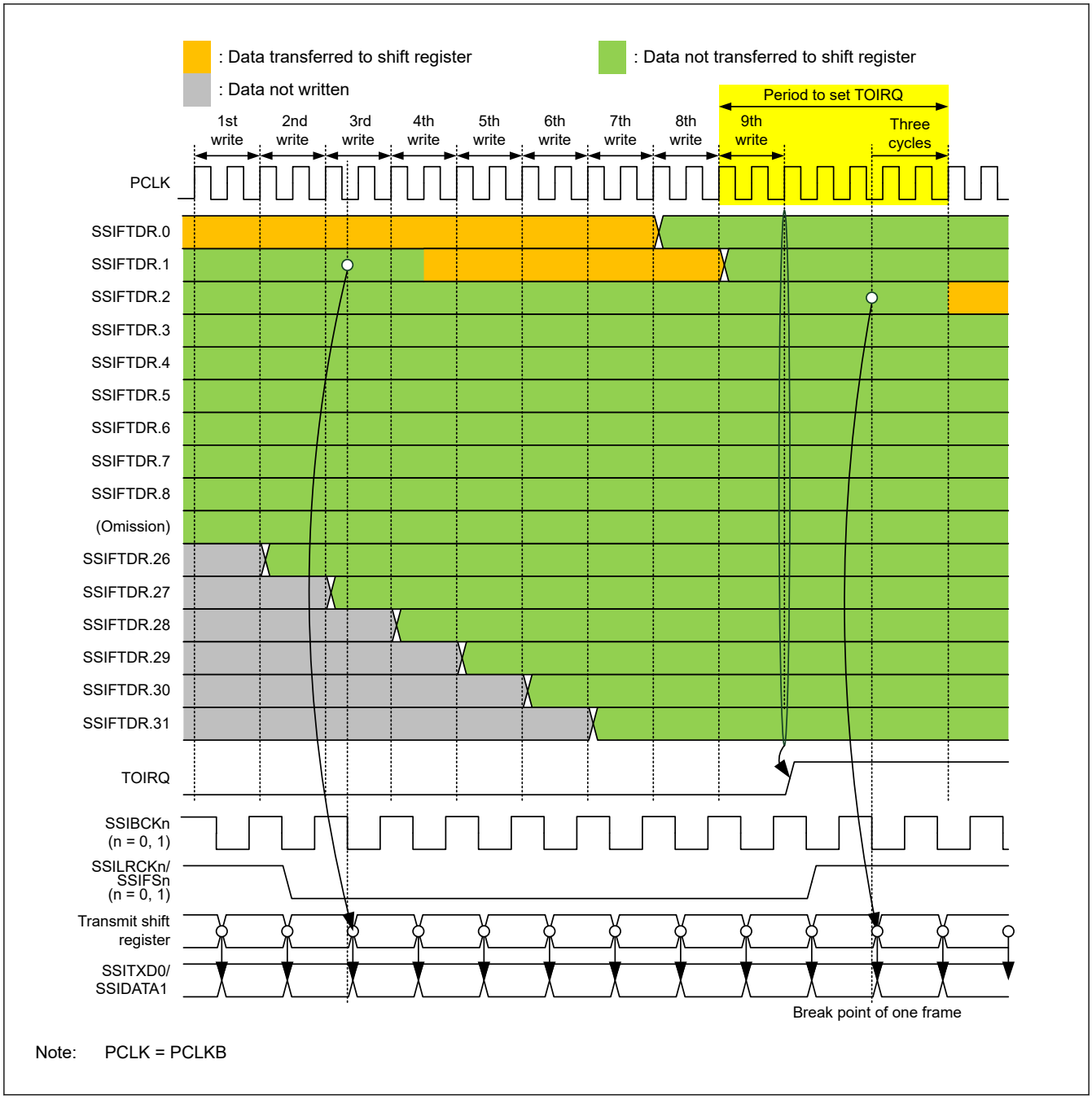


Figure 39.18 TOIRQ setting timing

TUIRQ flag (Transmit Underflow Error Status flag)

The TUIRQ flag is a status flag that indicates a transmit underflow error. This flag is set by automatic determination but it must be cleared by register access. This flag indicates that writing the serial data required for a frame to SSIFTDR did not catch up with transmission of the frame. Even if this flag is cleared after it has been set, the SSITXD0/SSIDATA1 output remains to be 0. To output the data written to the transmit FIFO data register (SSIFTDR) to the SSITXD0/SSIDATA1 pin, follow the communication stop procedure in Figure 39.56 and error-handling procedure in Figure 39.57. For the procedure to recover from an error, see section 39.6.6. Error Handling. This flag is not cleared by a reset of transmit FIFO data register (by the SSIFCR.TFRST signal).

[Priority order for setting and clearing]

Setting is prioritized.*1

[Clearing condition]

When either of the following operations is done:

1. Writing 0 to this bit after reading 1 from this bit*2
2. Enabling communication (changing SSICR.TEN from 0 to 1).

[Clearing timing]

Clearing timing corresponding to the above clearing condition

1. When 0 is written to this bit after reading 1 from this bit
2. One PCLKB cycle after writing 1 to SSICR.TEN.*3

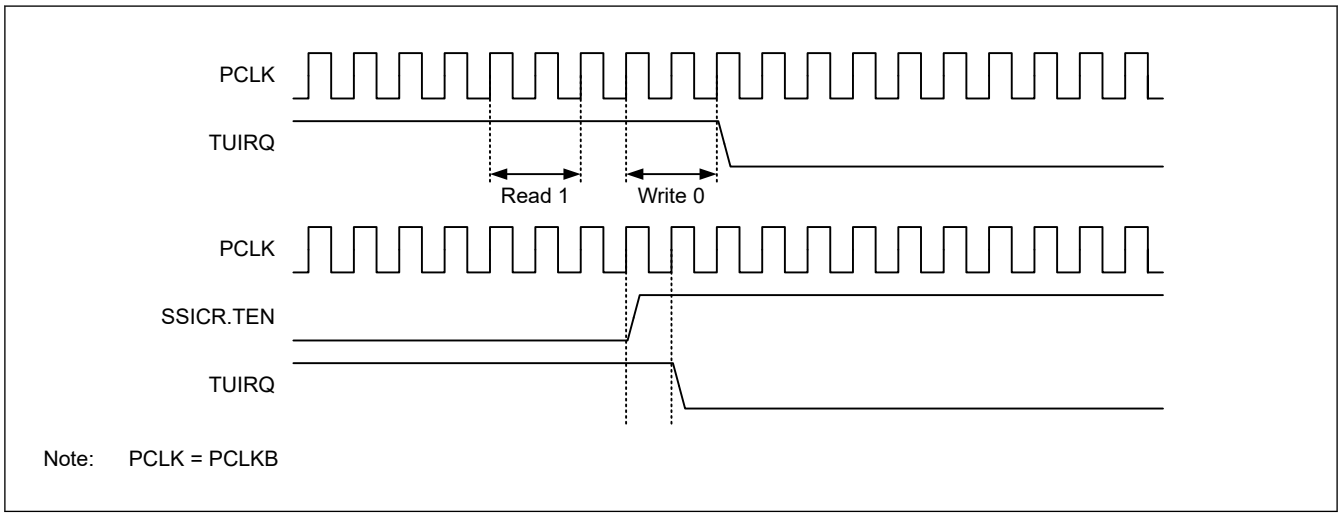


Figure 39.19 TUIRQ clearing timing

Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following three conditions is met:
- A software reset (SSIFCR.SSIRST = 1) is done.
- After 1 has been read, writing of 0 is complete.
- One PCLKB cycle passes after 1 has been written to SSICR.TEN.

Note 3. After communication is enabled (by changing the value of SSICR.TEN bit from 0 to 1), the transmission error flags (TOIRQ and TUIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the transmission error flags might be unable to be read.

[Setting condition]

When communication continues over a frame boundary, the transmit data required for the next frame has not been written to SSIFTDR. For details, see Figure 39.20 and Figure 39.21.

[Setting timing]

Three PCLKB cycles after the frame boundary. For details, see Figure 39.20.

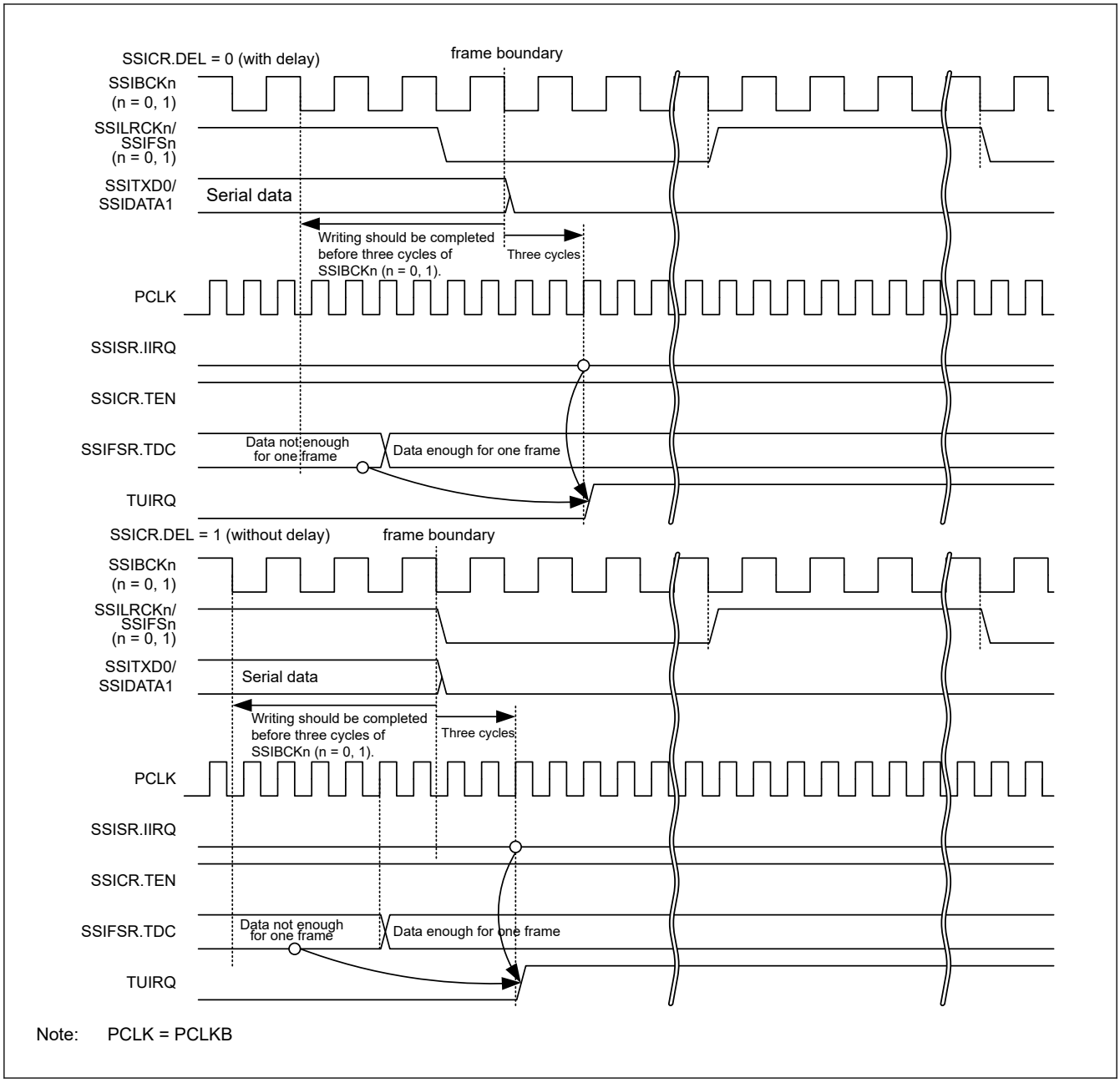


Figure 39.20 TUIRQ setting timing (when communication continues)

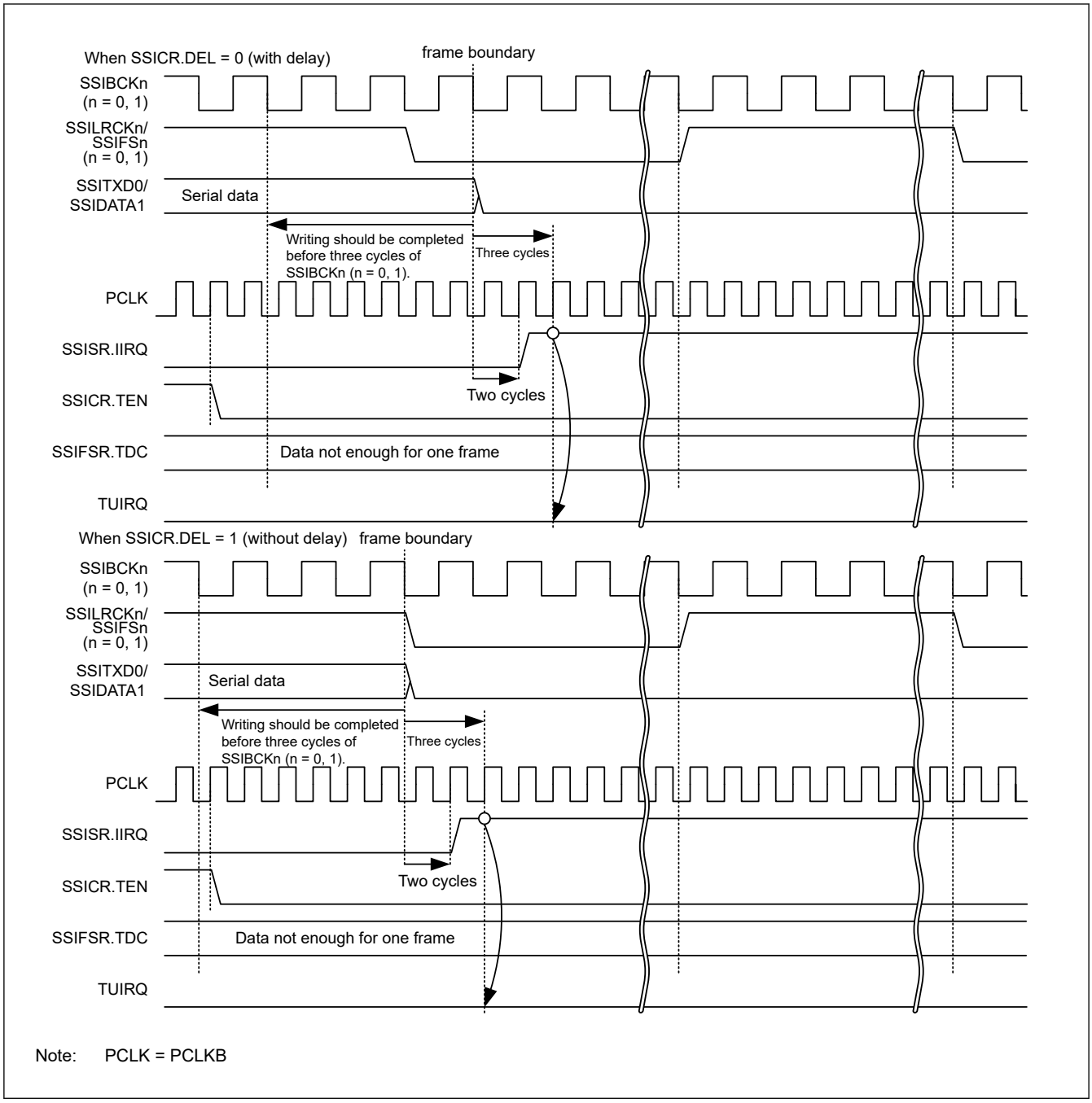


Figure 39.21 TUIRQ setting timing (when communication stops)

39.2.3 SSIFCR : FIFO Control Register

Base address: SSIEn = 0x4025_D000 + 0x0100 × n (n = 0, 1)
SSIEn_NS = 0x5025_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	AUCKE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	BSW	—	—	—	—	—	—	—	TIE	RIE	TFRST	RFRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFRST	Receive FIFO Data Register Reset*1 0: Clears a receive data FIFO reset condition 1: Sets a receive data FIFO reset condition	R/W
1	TFRST	Transmit FIFO Data Register Reset*1 0: Clears a transmit data FIFO reset condition 1: Sets a transmit data FIFO reset condition	R/W
2	RIE	Receive Data Full Interrupt Output Enable 0: Disables receive data full interrupts 1: Enables receive data full interrupts	R/W
3	TIE	Transmit Data Empty Interrupt Output Enable 0: Disables transmit data empty interrupts 1: Enables transmit data empty interrupts	R/W
10:4	—	These bits are read as 0. The write value should be 0.	R/W
11	BSW	Byte Swap Enable*1 0: Disables byte swap 1: Enables byte swap	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	SSIRST	Software Reset 0: Clears a software reset condition 1: Sets a software reset condition	R/W
30:17	—	These bits are read as 0. The write value should be 0.	R/W
31	AUCKE	AUDIO_MCK Enable in Mastermode Communication*1 0: Disables supply of AUDIO_MCK 1: Enables supply of AUDIO_MCK	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Writing to these bits while SSIE is in a communication state (SSISR.IIRQ = 0) is prohibited. If the value of these bits is changed by rewriting, subsequent operation is unpredictable.

This register sets a software reset, byte swap, and enable/disable of interrupt requests.

RFRST bit (Receive FIFO Data Register Reset)

The RFRST bit sets a software reset of the receive FIFO data register (SSIFRDR). Writing 1 to this bit initializes the internal state related to SSIFRDR. The register bits subject to the software reset triggered by this bit are indicated by shading in Table 39.7. Because this bit is not automatically cleared after it has been set, write 0 to this bit to release the register bits from the software reset. After writing 0 to this bit, be sure to check that this bit is 0 before starting the next procedural step.

This bit is subject to the software reset by the SSIRST bit. Because the software reset by the SSIRST bit has priority over the reset by this bit, setting this bit is ignored when the SSIRST bit is set.

Table 39.7 Bits subject to software reset by the RFRST bit

Symbol	Address (BASE+)		+0								+1							
			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSICR	0x00	+0	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	FRM[1:0]		DWL[2:0]			SWL[2:0]		
		+2	—	MST	BCKP	LRCKP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
SSISR	0x04	+0	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SSIFCR	0x10	+0	AUCKE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRST
		+2	—	—	—	—	BSW	—	—	—	—	—	—	—	TIE	RIE	TFRST	RFRST
SSIFSR	0x14	+0	—	—	TDC[5:0]						—	—	—	—	—	—	—	TDE
		+2	—	—	RDC[5:0]						—	—	—	—	—	—	—	RDF
SSIFTDR	0x18	+0	SSIFTDR[31:16]															
		+2	SSIFTDR[15:0]															
SSIFRDR	0x1C	+0	SSIFRDR[31:16]															
		+2	SSIFRDR[15:0]															
SSIOFR	0x20	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	BCKASTP	LRCONT	—	—	—	—	—	—	OMOD[1:0]	
SSISCR	0x24	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]				

TFRST bit (Transmit FIFO Data Register Reset)

The TFRST bit sets a software reset of the transmit FIFO data register (SSIFTDR). Writing 1 to this bit initializes the internal state related to SSIFTDR. The register bits subject to the software reset triggered by this bit are indicated by shading in Table 39.8. Because this bit is not automatically cleared after it has been set, write 0 to this bit to release the register bits from the software reset. After writing 0 to this bit, be sure to check that this bit is 0 before starting the next procedural step.

This bit is subject to the software reset by the SSIRST bit. Because the software reset by the SSIRST bit has priority over the reset by this bit, setting this bit is ignored when the SSIRST bit is set.

Table 39.8 Bits subject to software reset by the TFRST bit (1 of 2)

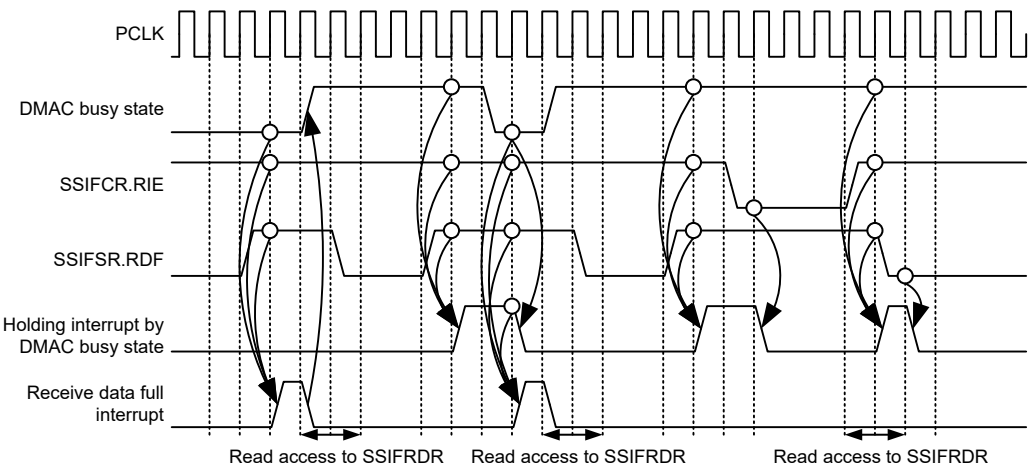
Symbol	Address (BASE+)		+0								+1							
			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSICR	0x00	+0	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	FRM[1:0]		DWL[2:0]			SWL[2:0]		
		+2	—	MST	BCKP	LRCKP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
SSISR	0x04	+0	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 39.8 Bits subject to software reset by the TFRST bit (2 of 2)

Symbol	Address (BASE+)		+0								+1							
			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSIFCR	0x10	+0	AUCKE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRST
		+2	—	—	—	—	BSW	—	—	—	—	—	—	—	TIE	RIE	TFRST	RFRST
SSIFSR	0x14	+0	—	—	TDC[5:0]						—	—	—	—	—	—	—	TDE
		+2	—	—	RDC[5:0]						—	—	—	—	—	—	—	RDF
SSIFTDR	0x18	+0	SSIFTDR[31:16]															
		+2	SSIFTDR[15:0]															
SSIFRDR	0x1C	+0	SSIFRDR[31:16]															
		+2	SSIFRDR[15:0]															
SSIOFR	0x20	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	BCKASTP	LRCONT	—	—	—	—	—	—	OMOD[1:0]	
SSISCR	0x24	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]				

RIE bit (Receive Data Full Interrupt Output Enable)

The RIE bit enables/disables output of receive data full interrupts. Use a receive data full interrupt as an interrupt to trigger data reading from the receive FIFO data register. Write 1 to this bit after specifying the setting condition for receive data full interrupt (by using the SSISCR.RDFS bit). Figure 39.22 shows the timing of generating the receive data full interrupt.

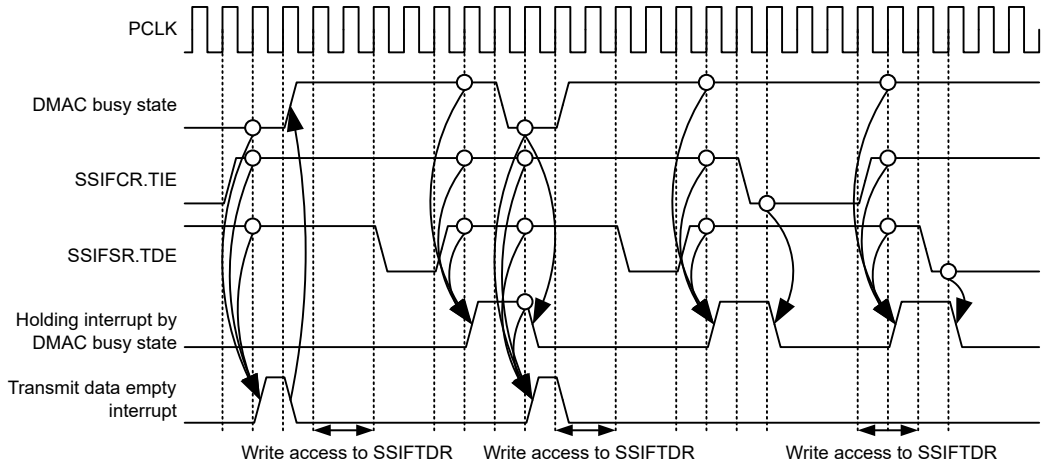


Note: PCLK = PCLKB

Figure 39.22 Timing of receive data full interrupt

TIE bit (Transmit Data Empty Interrupt Output Enable)

The TIE bit enables/disables output of transmit data empty interrupts. Use a transmit data empty interrupt as an interrupt to trigger data writing to the transmit FIFO data register. Write 1 to this bit after specifying the setting condition for transmit data empty interrupt (by using the SSISCR.TDES bit). Figure 39.23 shows the timing of generating the transmit data empty interrupt.



Note: PCLK = PCLKB

Figure 39.23 Timing of transmit data empty interrupt

BSW bit (Byte Swap Enable)

The BSW bit enables/disables byte swap of register access for the transmit FIFO data register (SSIFTDR) and the receive FIFO data register (SSIFRDR). This bit is valid only with 16-bit access or 32-bit access to SSIFTDR and SSIFRDR. For details, see Figure 39.24.

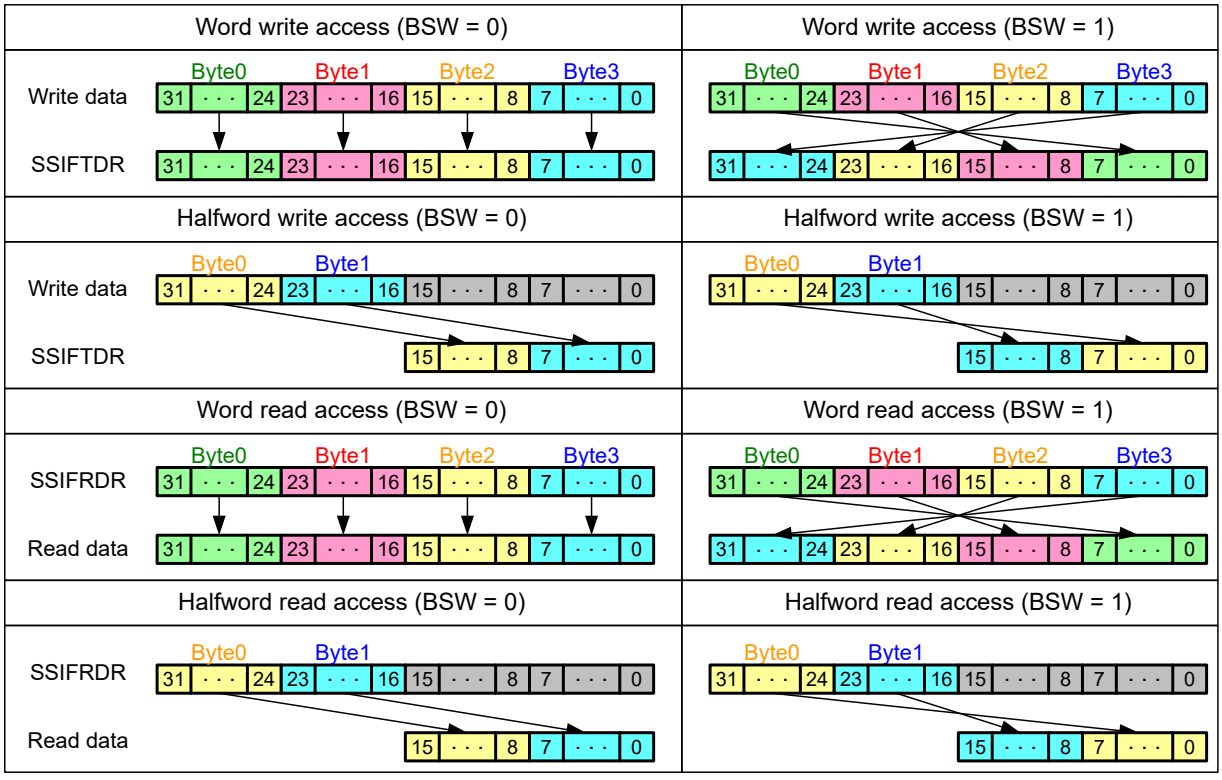


Figure 39.24 Operation example of byte swap

SSIRST bit (Software Reset)

The SSIRST bit sets a software reset of SSIE. Writing 1 to this bit initializes the internal state of SSIE. The register bits subject to the software reset triggered by this bit are indicated by shading in Table 39.9. Because this bit is not automatically

cleared after it has been set, write 0 to this bit to release the register bits from the software reset. After writing 0 to this bit, be sure to check that this bit is 0 before starting the next procedural step.

To stop communication of SSIE immediately, after turning off the peripheral functions, write 1 to this bit. Initialization by a software reset is performed without any relation with the bit clock.

Table 39.9 Bits subject to software reset by the SSIRST bit

Symbol	Address (BASE+)		+0								+1							
			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSICR	0x00	+0	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	FRM[1:0]		DWL[2:0]			SWL[2:0]		
		+2	—	MST	BCKP	LRCKP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
SSISR	0x04	+0	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SSIFCR	0x10	+0	AUCKE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRST
		+2	—	—	—	—	BSW	—	—	—	—	—	—	—	TIE	RIE	TFRST	RFRST
SSIFSR	0x14	+0	—	—	TDC[5:0]						—	—	—	—	—	—	—	TDE
		+2	—	—	RDC[5:0]						—	—	—	—	—	—	—	RDF
SSIFTDR	0x18	+0	SSIFTDR[31:16]															
		+2	SSIFTDR[15:0]															
SSIFRDR	0x1C	+0	SSIFRDR[31:16]															
		+2	SSIFRDR[15:0]															
SSIOFR	0x20	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	BCKASTP	LRCONT	—	—	—	—	—	—	OMOD[1:0]	
SSISCR	0x24	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]				

AUCKE bit (AUDIO_MCK Enable in Mastermode Communication)

The AUCKE bit enables/disables supply to AUDIO_MCK while in master-mode communication (MST = 1).

Changing the value of this bit must be performed only after specifying the settings related to AUDIO_MCK (by using the CKS, MST, BCKP, and CKDV bits in the SSICR register).

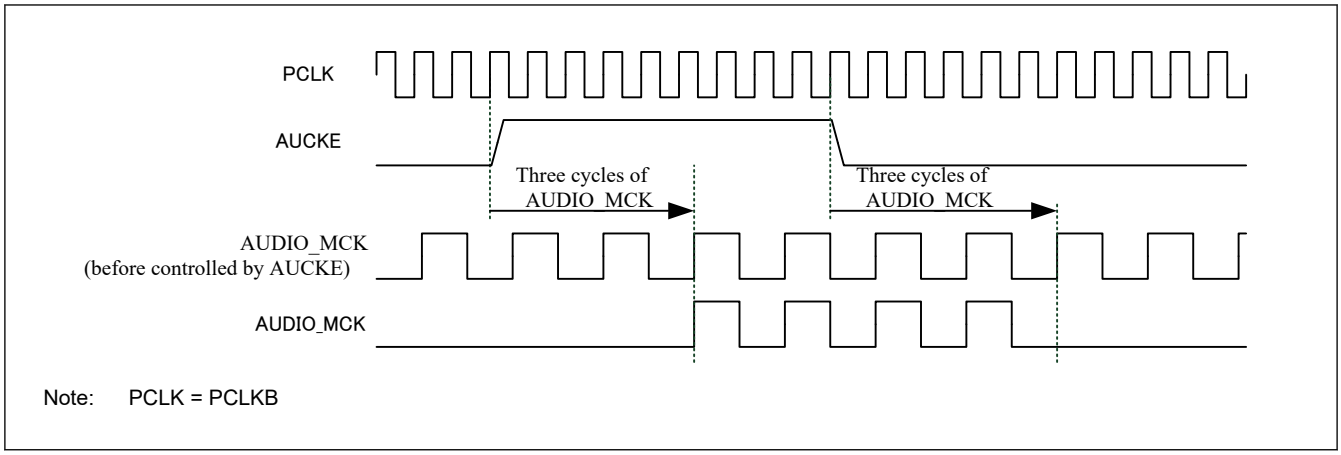


Figure 39.25 Stop/resume of AUDIO_MCK

Note: In slave-mode communication (SSICR.MST = 0), SSIE needs supply of SSIBCKn (n = 0, 1). To stop BCK on the master side, make sure that SSIE is in the idle state (SSISR.IIRQ = 1). If BCK is stopped before SSIE becomes idle, take the procedure to start communication in Figure 39.52 or wait for an idle state by taking the procedure to resume communication in Figure 39.58.

In master-mode communication (SSICR.MST = 1), SSIE operates with the audio clock (AUDIO_MCK). To stop SSIE completely, make sure that SSIE is in the idle state (SSISR.IIRQ = 1) and then write 0 to SSIFCR.AUCKE. If 0 is written to SSIFCR.AUCKE before SSIE becomes idle, take the procedure to start communication in Figure 39.52.

Figure 39.26 and Figure 39.27 show the timings of signal operation in the period from setting this bit to 1 to the output to the SSIBCKn (n = 0, 1) pin.

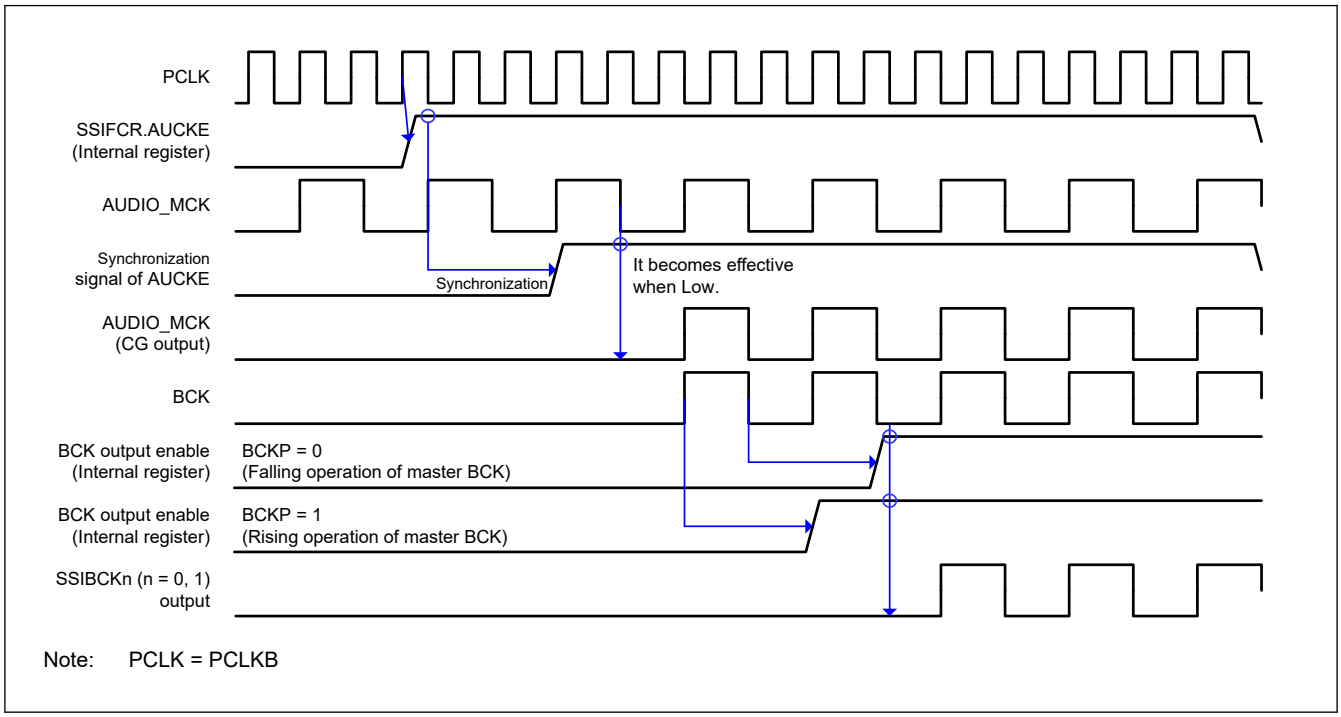


Figure 39.26 Timing diagram for the operation from system reset to start of master-mode communication

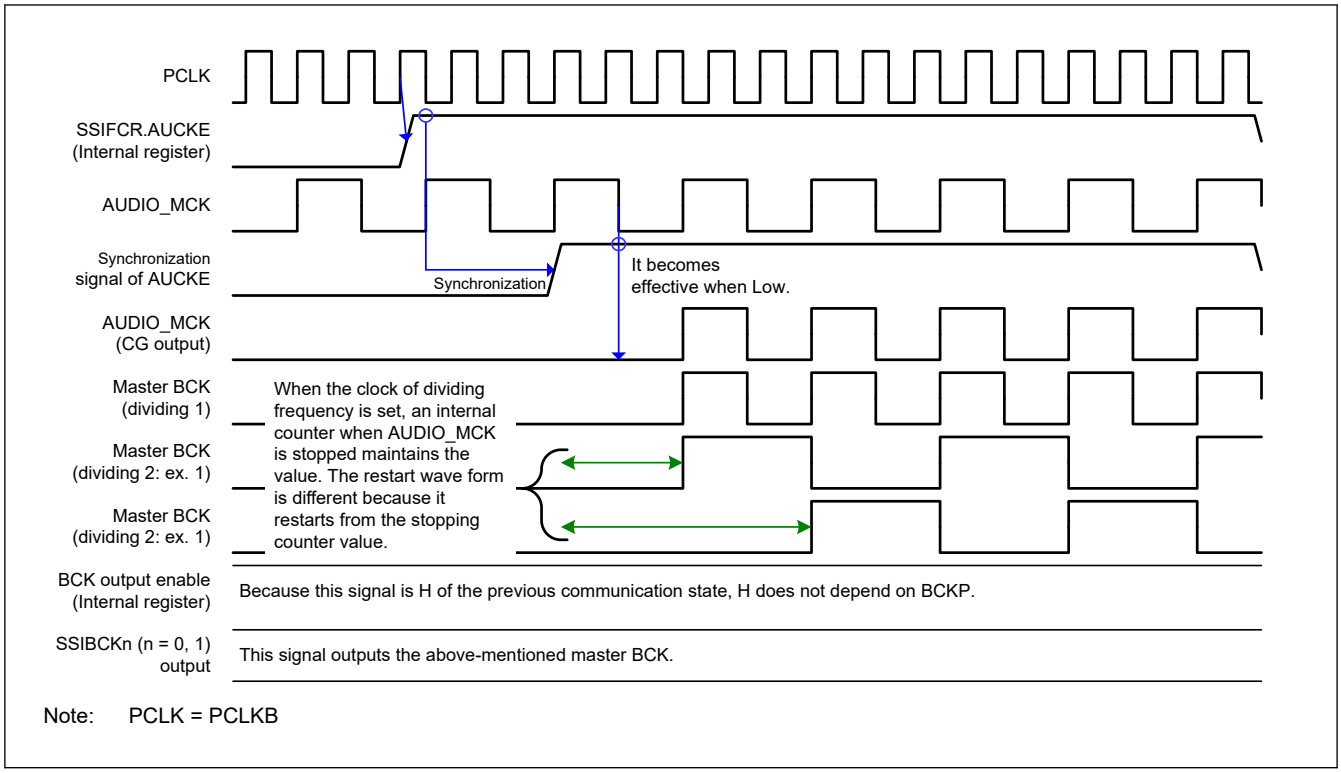


Figure 39.27 Timing diagram for the operation from stop of communication to start of master-mode communication

Note: If the supply of AUDIO_MCK stops, the value of the SSIBCKn (n = 0, 1) pin is held. Therefore, the SSIBCKn (n = 0, 1) signal might stop in the H (high level) state.

39.2.4 SSIFSR : FIFO Status Register

Base address: SSIEn = 0x4025_D000 + 0x0100 × n (n = 0, 1)
SSIEn_NS = 0x5025_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	TDC[5:0]						—	—	—	—	—	—	—	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	RDC[5:0]						—	—	—	—	—	—	—	RDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RDF	Receive Data Full Flag 0: The size of received data in SSIFRDR is not more than the value of SSISCR.RDFS. 1: The size of received data in SSIFRDR is not less than the value of SSISCR.RDFS plus one.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
13:8	RDC[5:0]	Receive Data Count Number of valid data stored in the receive FIFO data register	R
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	TDE	Transmit Data Empty Flag 0: The free space of SSIFTDR is not more than the value of SSISCR.TDES. 1: The free space of SSIFTDR is not less than the value of SSISCR.TDES plus one.	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
29:24	TDC[5:0]	Transmit Data Count Number of valid data stored in the transmit FIFO data register	R
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register is configured with status flags that indicate the status of the transmit FIFO data register and the receive FIFO data register.

RDF flag (Receive Data Full Flag)

The RDF flag indicates that the receive FIFO data register (SSIFRDR) has unread received data not less than the amount set with the SSISCR.RDFS bits plus one. This flag is set by automatic determination but it must be cleared by register access.

[Priority order for setting and clearing]

Clearing is prioritized.

[Clearing condition]

Either of the following two:[*1]

1. Writing 0 to this bit after reading 1 from this bit (CPU operation)[*2]
2. Last access (DTC/DMAC operation) to read data from SSIFRDR by an interrupt routine using the DTC and DMAC.

[Clearing timing]

Clearing timing corresponding to the above clearing condition

1. When 0 is written to this bit after reading 1 from this bit (same as the timing in Figure 39.19)
2. After the PCLKB cycle in which the last access instruction is issued to read data from SSIFRDR by an interrupt routine using the DTC and DMAC.

[Setting condition]

SSIFRDR has free space not less than the amount set with the SSISCR.RDFS bits plus one.

[Setting timing]

At completion of transfer from the shift register that results in SSIFRDR having data not less than the amount set with the SSISCR.RDFS bits plus one.

Note 1. These bits are cleared by a software reset (SSIFCR.SSIRST = 1) and receive FIFO data register reset (SSIFCR.RFRST = 1). Reset conditions available for these bits are the software reset and receive FIFO data register reset as well as the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following four conditions is met:
- A software reset is done (SSIFCR.SSIRST = 1).
- A receive FIFO data register reset is done (SSIFCR.RFRST = 1).
- After 1 has been read, writing of 0 is complete.
- Last access is performed to read data from SSIFRDR by an interrupt routine using the DTC and DMAC.

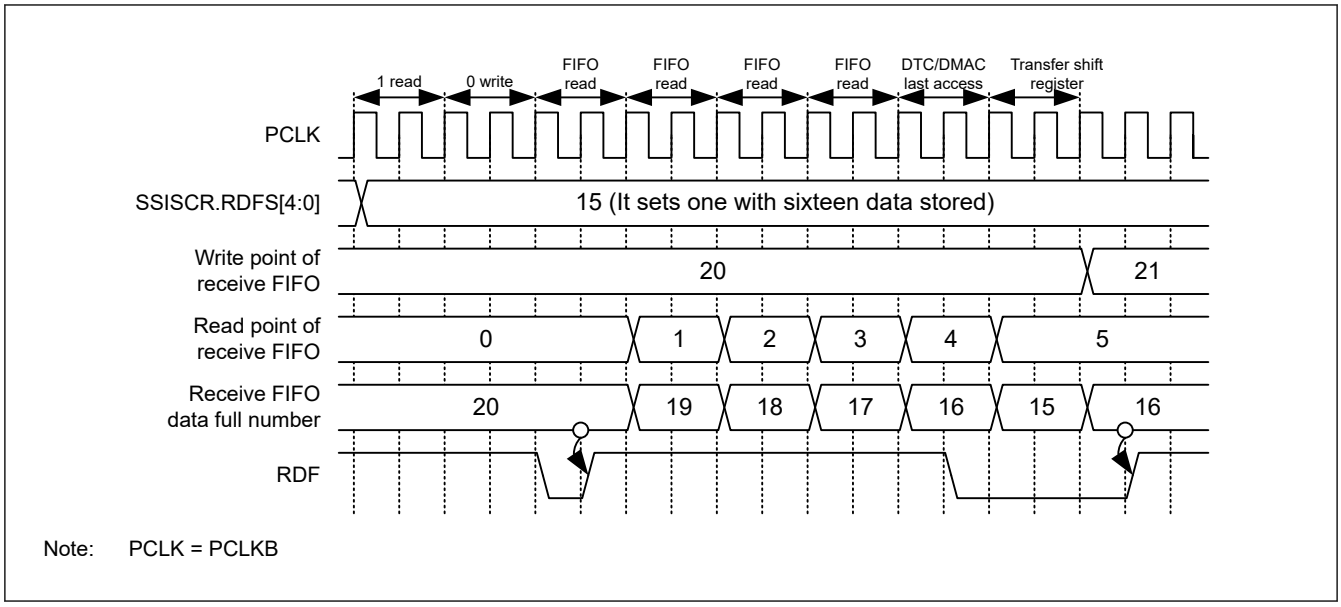


Figure 39.28 Timing diagram for setting and clearing RDF

RDC[5:0] bits (Receive Data Count)

The RDC[5:0] bits indicate the number of valid data that are stored in the receive FIFO data register (SSIFRDR). With these bits as 0x00, there is no received data. With 0x20, the register is filled with received data and there is no free space.

TDE flag (Transmit Data Empty Flag)

The TDE flag indicates that the transmit FIFO data register (SSIFTDR) has free space not less than the amount set with the SSISCR.TDES bits plus one. This flag is set by automatic determination but it must be cleared by register access.

[Priority order for setting and clearing]

Clearing is prioritized.*1

[Clearing condition]

Either of the following two:

1. Writing 0 to this bit after reading 1 from this bit (CPU operation)*2
2. Last access (DTC/DMAC operation) to write data to SSIFTDR by an interrupt routine using the DTC and DMAC.

[Clearing timing]

Clearing timing corresponding to the above clearing condition

1. When 0 is written to this bit after reading 1 from this bit (same as the timing in Figure 39.19)
2. Last access (DTC/DMAC operation) to write data to SSIFTDR by an interrupt routine using the DTC and DMAC.

[Setting condition]

SSIFTDR has free space not less than the amount set with the SSISCR.TDES bits plus one.

[Setting timing]

While operating on PCLKB, SSIFTDR is found to have free space not less than “size set in the SSISCR.TDES bits + 1.”

Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1) and transmit FIFO data register reset (SSIFCR.TFRST = 1). The software reset and transmit FIFO data register reset have priority over all the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following four conditions is met:
- A software reset is done (SSIFCR.SSIRST = 1).
- A transmit FIFO data register reset is done (SSIFCR.TFRST = 1).
- After 1 has been read, writing of 0 is complete.
- Last access is performed to write data to SSIFTDR by an interrupt routine using the DTC and DMAC.

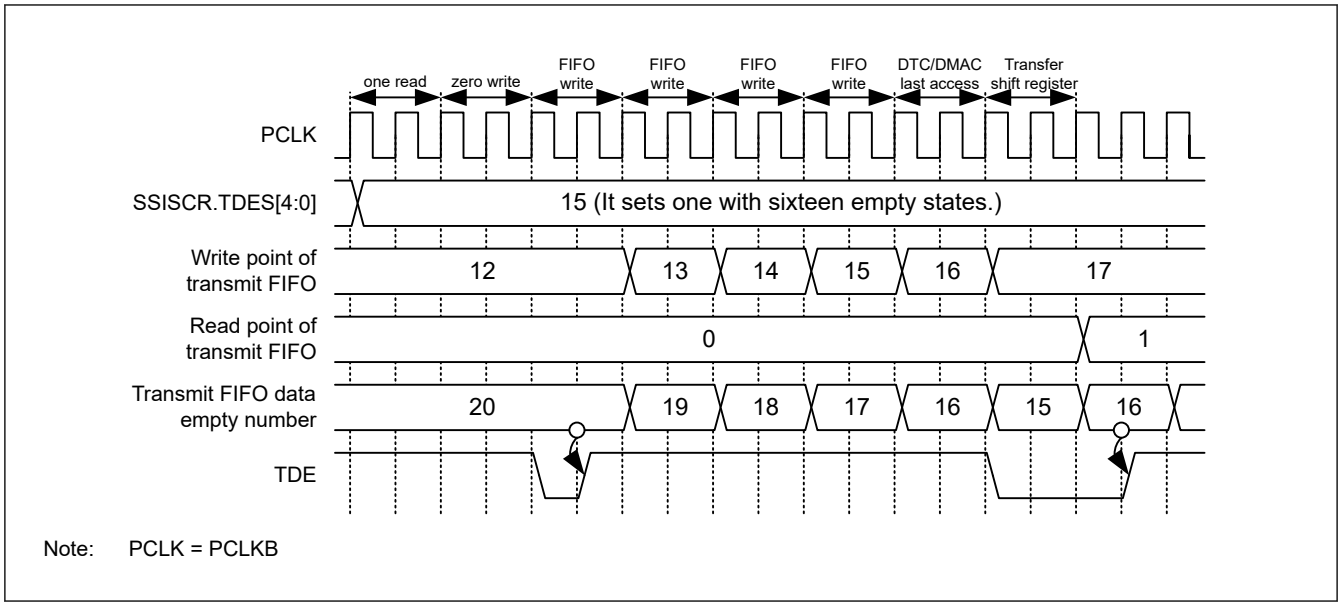


Figure 39.29 Timing diagram for setting and clearing TDE

TDC[5:0] bits (Transmit Data Count)

The TDC[5:0] bits indicate the number of valid data that are stored in the transmit FIFO data register (SSIFTDR). With these bits as 0x00, there is no data to be transmitted. With 0x20, there is no space to write data.

39.2.5 SSIFTDR : Transmit FIFO Data Register

Base address: SSIEn = 0x4025_D000 + 0x0100 × n (n = 0, 1)
SSIEn_NS = 0x5025_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x18

Bit position:	31																													0
Bit field:	SSIFTDR[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SSIFTDR[31:0]	Transmit FIFO Data	W

Note: S-TYPE-3, P-TYPE-3

This register stores data to be serially transmitted. 0 is returned when this register is read.

When you use this register for transmission, specify data writing to this register as the DTC/DMAC operation that is triggered by a transmit data empty interrupt. Determine the access size to this register according to the data word length to be communicated in Table 39.10.

Table 39.10 Register access restriction to FIFOs (1 of 2)

Access Size				
SSICR.DWL[2:0]	Data Word Length	Byte	Halfword	Word
000b	8	✓	—	—
001b	16	—	✓	—
010b	18	—	—	✓
011b	20	—	—	✓
100b	22	—	—	✓
101b	24	—	—	✓

Table 39.10 Register access restriction to FIFOs (2 of 2)

Access Size				
SSICR.DWL[2:0]	**Data Word Length**	**Byte**	**Halfword**	**Word**
110b	32	—	—	✓
111b	Setting prohibited	—	—	—

Figure 39.30 shows register access to the transmit FIFO data register.

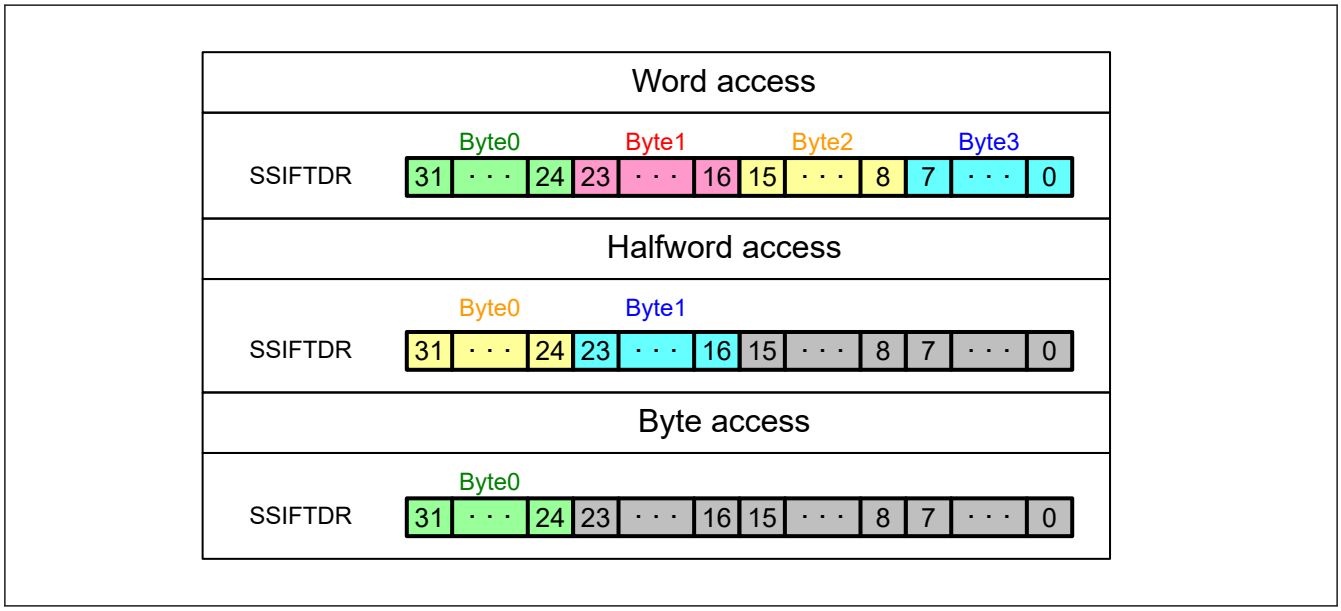


Figure 39.30 Example of register access to the transmit FIFO data register

Figure 39.31 shows the configurations and operation examples of the transmit FIFO data register and transmit shift register. The configurations are for storing data to FIFO and not related with communication.

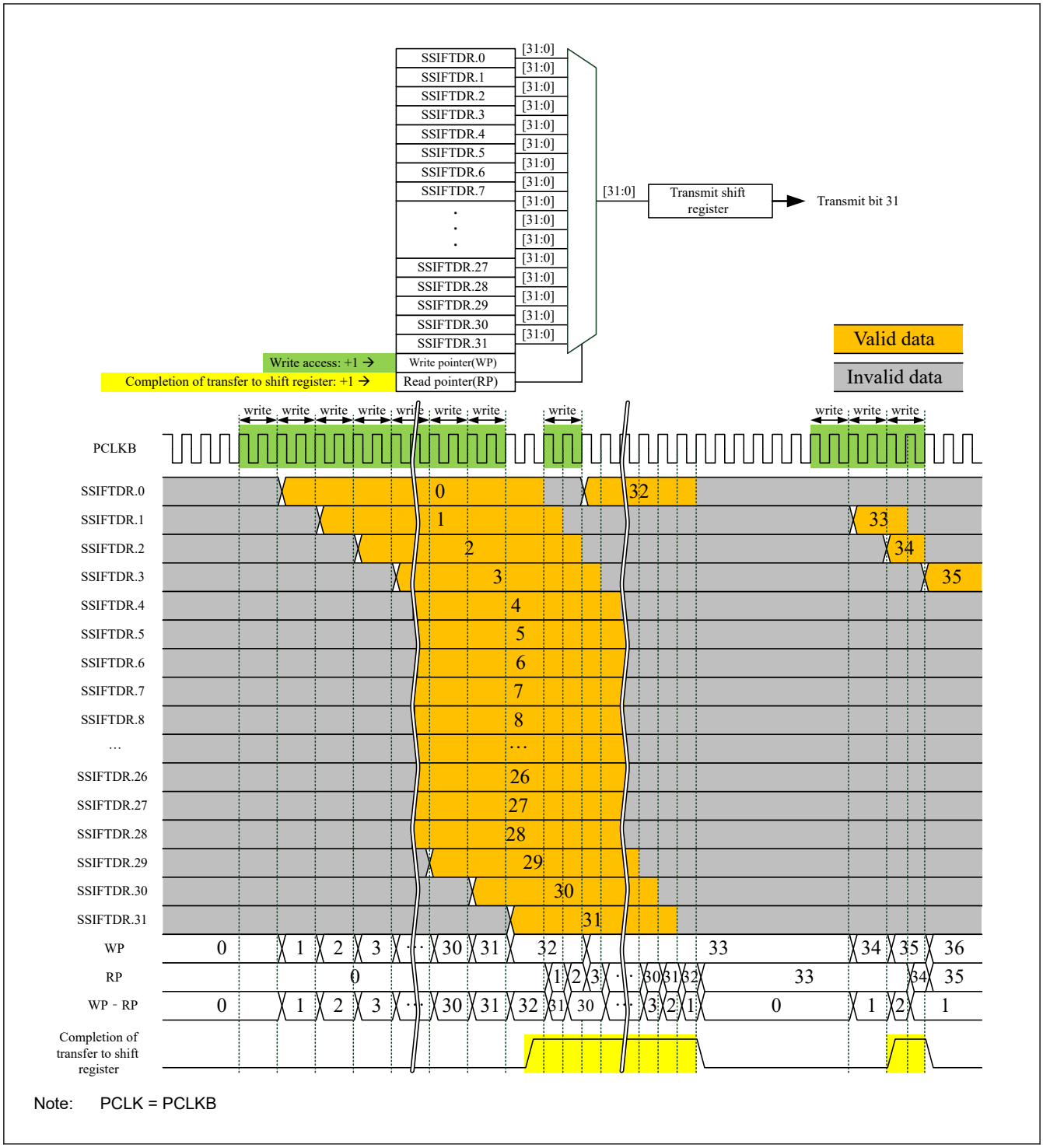


Figure 39.31 Configuration of the transmit FIFO data register and transmit shift register, and FIFO operation example

39.2.6 SSIFRDR : Receive FIFO Data Register

Base address: SSIEn = 0x4025_D000 + 0x0100 × n (n = 0, 1)
SSIEn_NS = 0x5025_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x1C

Bit position:	31																												0
Bit field:	SSIFRDR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SSIFRDR[31:0]	Receive FIFO Data	R

Note: S-TYPE-3, P-TYPE-3

When you use this register for reception, specify data reading from this register as the DTC/DMAC operation that is triggered by a transmit data empty interrupt. Determine the access size to this register according to the data word length to be communicated in Table 39.10.

Register access to the receive FIFO data register is same as for the transmit FIFO data register.

Figure 39.32 shows the configurations and operation examples of the receive FIFO data register and receive shift register.

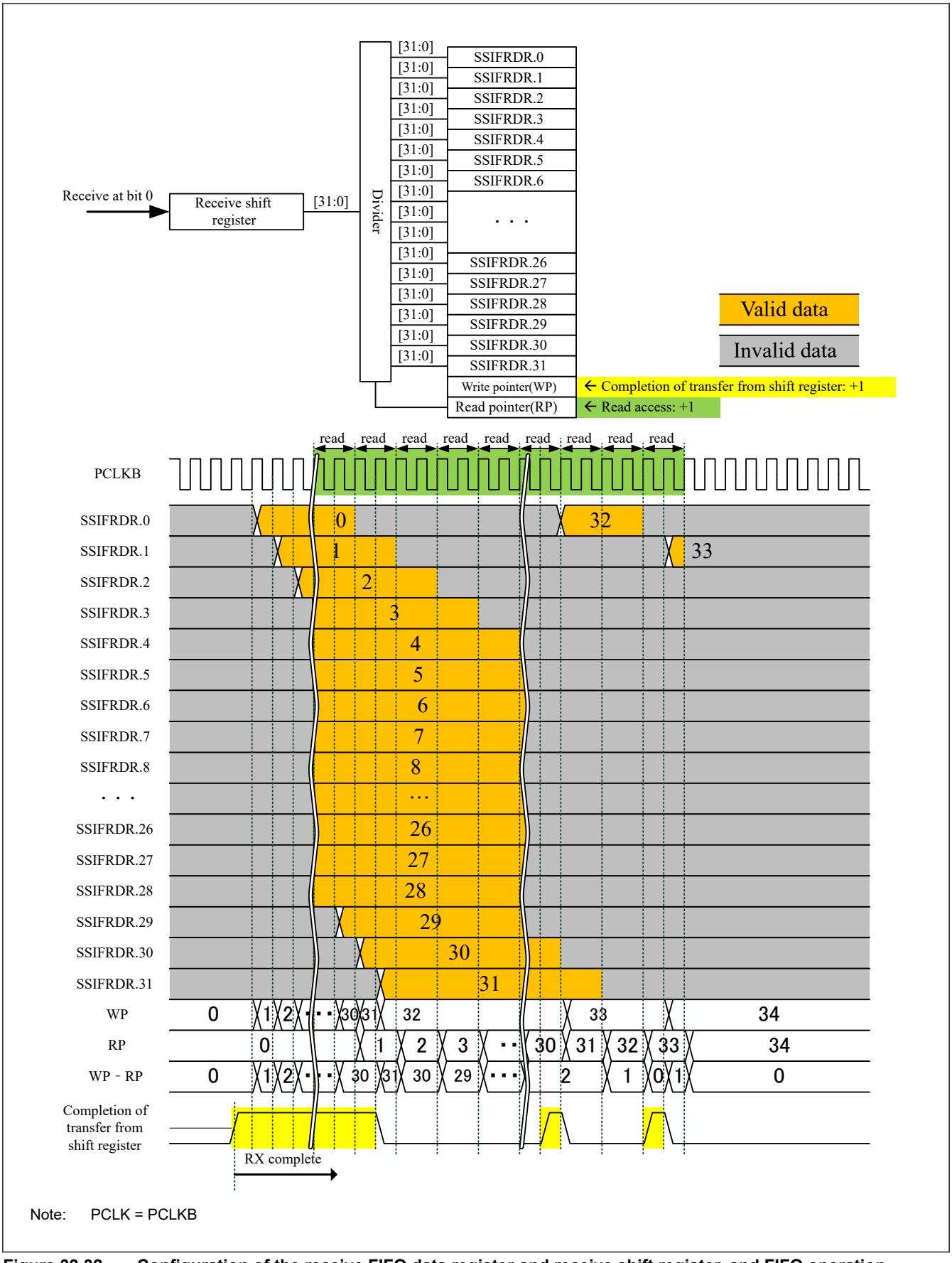


Figure 39.32 Configuration of the receive FIFO data register and receive shift register, and FIFO operation example

39.2.7 SSIOFR : Audio Format Register

Base address: SSIEn = 0x4025_D000 + 0x0100 × n (n = 0, 1)
SSIEn_NS = 0x5025_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	BCKASTP	LRCONT	—	—	—	—	—	—	OMOD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	OMOD[1:0]	Audio Format Select*3 *4 0 0: I²S format 0 1: TDM format 1 0: Monaural format 1 1: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	LRCONT	Whether to Enable LRCK/FS Continuation*1 *2 0: Disables LRCK/FS continuation 1: Enables LRCK/FS continuation	R/W
9	BCKASTP	Whether to Enable Stopping BCK Output When SSIE is in Idle Status*1 *2 0: Always outputs BCK to the SSIBCKn (n = 0, 1) pin 1: Automatically controls output of BCK to the SSIBCKn (n = 0, 1) pin	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is valid only in master-mode communication (SSICR.MST = 1). The setting is invalid in slave-mode communication (SSICR.MST = 0).
Note 2. The BCKASTP and LRCONT bits must not be set to 1 together.
Note 3. While SSIE is communicating (SSISR.IIRQ = 0), writing to these bits is prohibited. If the value of these bits is changed by writing, subsequent operation is unpredictable.
Note 4. If the communication format of other-party device is compatible with a communication format of SSIE, specify and use the communication format that enables communication with the other-party device.

This register is used to set an audio format (which involves the settings of communication format, LR clock/frame synchronization continuation mode, and BCK output stop).

OMOD[1:0] bits (Audio Format Select)

The OMOD[1:0] bits set an audio format. Writing to these bits must be performed when the LR clock supply to the SSILRCKn/SSIFSn (n = 0, 1) pin is stopped. For details about the output of LR clock, see the detailed description of the LRCONT bit in section 39.2.7. SSIOFR : Audio Format Register.

LRCONT bit (Whether to Enable LRCK/FS Continuation)

The LRCONT bit enables or disables the output from SSILRCKn/SSIFSn (n = 0, 1) pin when the communication mode is master-mode communication (SSICR.MST = 1) and SSIE is in the idle state (SSISR.IIRQ = 1).

Even in the idle state, a signal can output from the SSILRCKn/SSIFSn (n = 0, 1) pin when this bit is set to 1 (to enable LR clock/frame synchronization continuation) in master mode (SSICR.MST = 1).

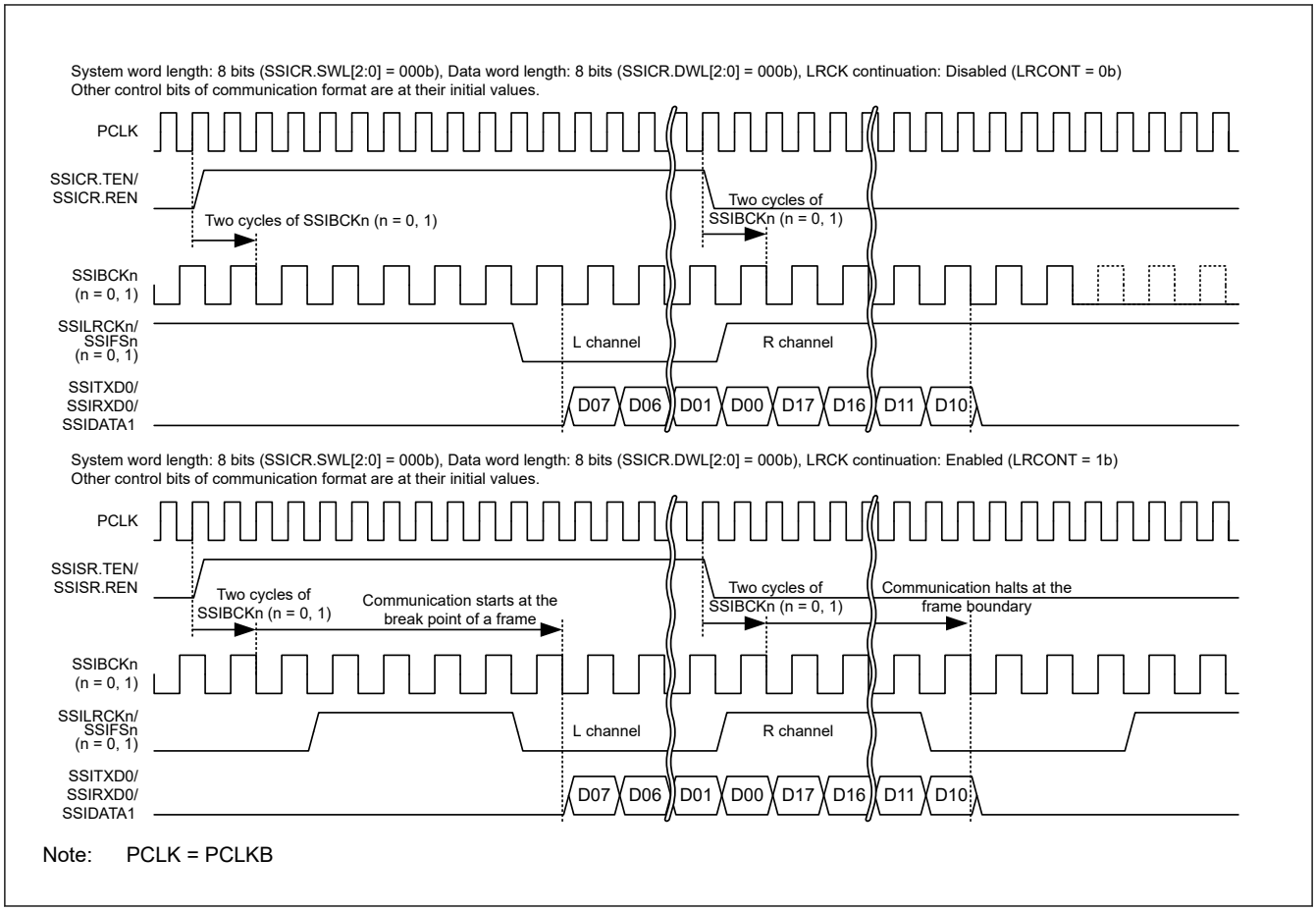


Figure 39.33 Example of LR clock/frame synchronization continuation operation

BCKASTP bit (Whether to Enable Stopping BCK Output When SSIE is in Idle Status)

The BCKASTP bit turns on or off the function to output BCK to the SSIBCKn (n = 0, 1) pin according to the communication shown in Figure 39.34 and Figure 39.35 in master-mode communication (SSICR.MST = 1).

Changing the value of this bit must be performed only after setting the communication format to be used.

This bit must be used in the following way:

Write 0 to the BCKASTP bit, and then start communication. During the communication, write 1 to the BCKASTP bit. By this operation, the bit clock output to the SSIBCKn (n = 0, 1) pin stops automatically when the communication stops. To resume the communication, set SSIE to the idle state (SSICR.IIRQ = 1), enable the supply of AUDIO_MCK (SSIFCR.AUCKE = 1), and then write 0 to the BCKASTP bit.

When the communication mode is master-mode communication (SSICR.MST = 1) and SSIE is in the idle state (SSICR.IIRQ = 1):

Table 39.11 BCKASTP bit status and SSIBCKn (n = 0, 1) pin output

BCKASTP Bit	SSIBCKn (n = 0, 1) Pin Output Status
0	Output
1	Stopped

Note: The BCKASTP bit cannot be used when the other-party device (which is a slave) requires the clock output from the SSIBCKn (n = 0, 1) pin before and during communication. In such a case, use the BCKASTP bit to stop the clock only after communication. For the timing of enabling the clock stop function, see Figure 39.34.

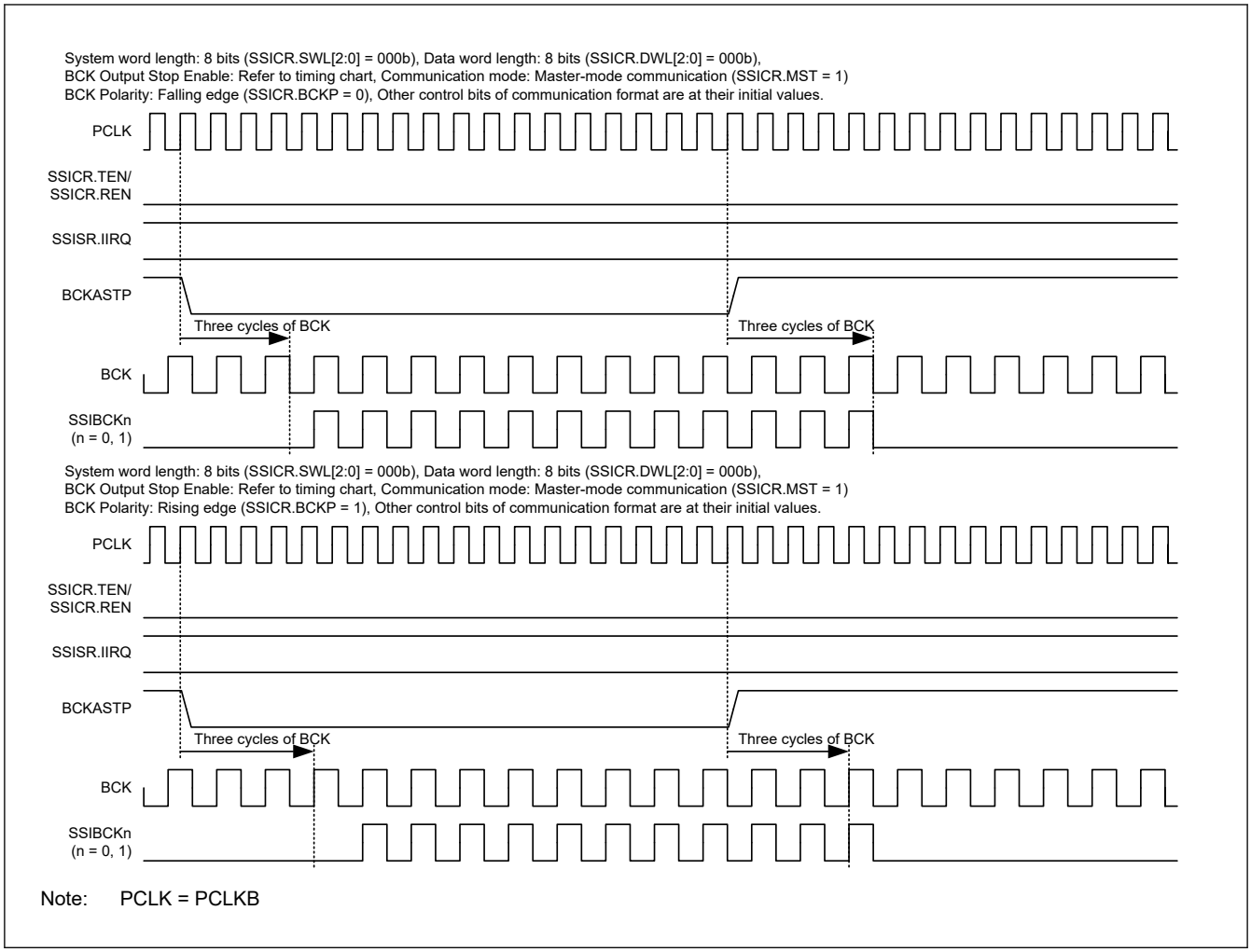


Figure 39.34 Example operation of the BCKASTP bit (idle state)

When the communication mode is master-mode communication (SSICR.MST = 1) and the BCK output stop function is enabled (BCKASTP = 1):

Details of the BCK output to the SSIBCKn (n = 0, 1) pin are as follows:

Output start timing: BCK is output in appropriate timing so that a valid edge is generated when the LR clock/frame synchronization signal shifts to a valid value.

Output stop timing: 1 to 1.5 clock cycles after a frame boundary.

For details about the timings, see the timing diagram in Figure 39.35.

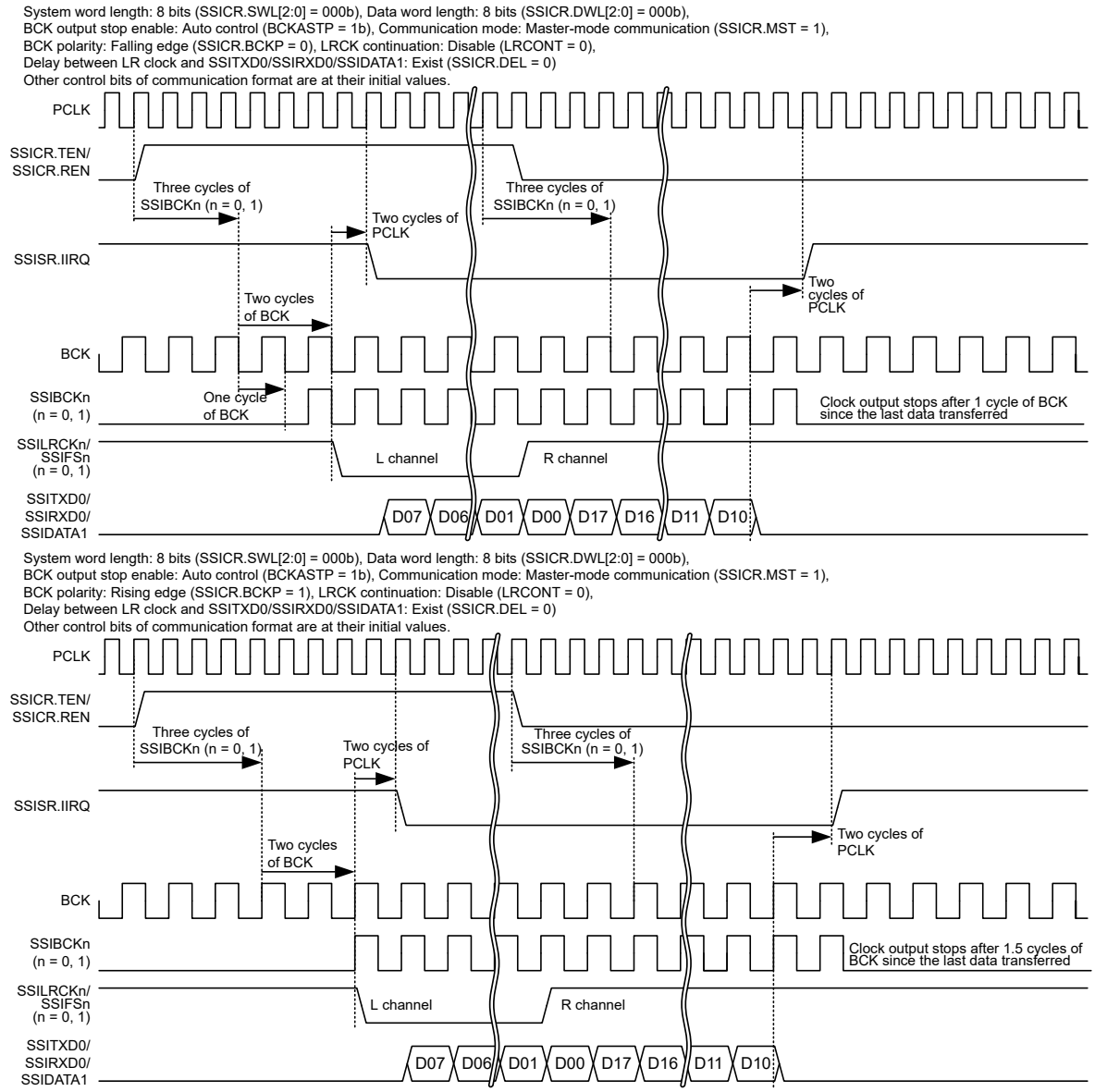


Figure 39.35 Example operation of the BCKASTP bit (communication operation with BCKASTP = 1)

39.2.8 SSISCR : Status Control Register

Base address: SSIEn = 0x4025_D000 + 0x0100 × n (n = 0, 1)
SSIEn_NS = 0x5025_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	RDFS[4:0]	RDF Setting Condition Select*1 0x00: SSIFRDR has one stage or more data size. 0x01: SSIFRDR has two stages or more data size. ⋮ 0x1E: SSIFRDR has thirty-one stages or more data size. 0x1F: SSIFRDR has thirty-two stages or more data size.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
12:8	TDES[4:0]	TDE Setting Condition Select*1 0x00: SSIFTDR has one stage or more free space. 0x01: SSIFTDR has two stages or more free space. ⋮ 0x1E: SSIFTDR has thirty-one stages or more free space. 0x1F: SSIFTDR has thirty-two stages or more free space.	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Writing to these bits while SSIE is in a communication state (SSISR.IIRQ = 0) is prohibited. If written, the operation performed immediately after writing is not guaranteed.

RDFS[4:0] bits (RDF Setting Condition Select)

The RDFS[4:0] bits set the setting condition of the receive data full flag (RDF).

TDES[4:0] bits (TDE Setting Condition Select)

The TDES[4:0] bits set the setting condition of the transmit data empty flag (TDE).

39.3 Communication Formats

SSIE supports three communication formats. Table 39.12 shows supported communication formats.

Table 39.12 Supported communication formats

Communication Format	SSIOFR.OMOD[1:0]
I²S format	00
TDM format	01
Monaural format	10

The following describes the serial data structure shared by communication formats. A serial data structure is defined by the system word length (set in SSICR.SWL[2:0]) and the data word length (set in SSICR.DWL[2:0]). If the data word length is shorter than the system word length, padding bits are transferred in the serial data. For details, see Figure 39.36.

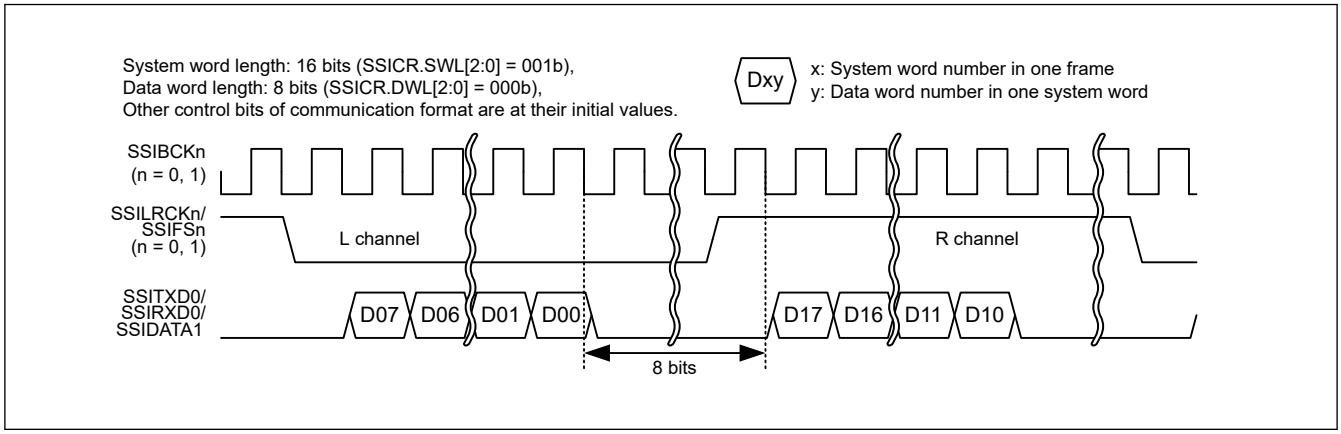


Figure 39.36 Example of padding bit transfer (I²S format: system word length > data word length)

Table 39.13 lists the number of padding bits to be transferred with each combination of system word length (SSICR.SWL[2:0]) and data word length (SSICR.DWL[2:0]). "-" indicates that the setting is prohibited.

Table 39.13 Number of padding bits

	SSICR.DWL[2:0]	000b	001b	010b	011b	100b	101b	110b	111b
SSICR.SWL[2:0]	System Word Length	8	16	18	20	22	24	32	Setting prohibited
000b	8	0	—	—	—	—	—	—	—
001b	16	8	0	—	—	—	—	—	—
010b	24	16	8	6	4	2	0	—	—
011b	32	24	16	14	12	10	8	0	—
100b	48	40	32	30	28	26	24	16	—
101b	64	56	48	46	44	42	40	32	—
110b	128	120	112	110	108	106	104	96	—
111b	256	248	240	238	236	234	232	224	—

39.3.1 I²S Format

The I²S format is a communication format used for connection with I²S-compatible serial devices. With this format setting (SSIOFR.OMOD[1:0] = 00b), one frame is configured with two system words, one for the channel L and the other for channel R. The SSILRCKn/SSIFSn (n = 0, 1) signals are at a low level for the channel L and at a high level for the channel R. Set the polarity of the signals with the SSICR.LRCKP bit. Figure 39.37 shows the I²S format without padding. See Figure 39.36 for the format with padding.

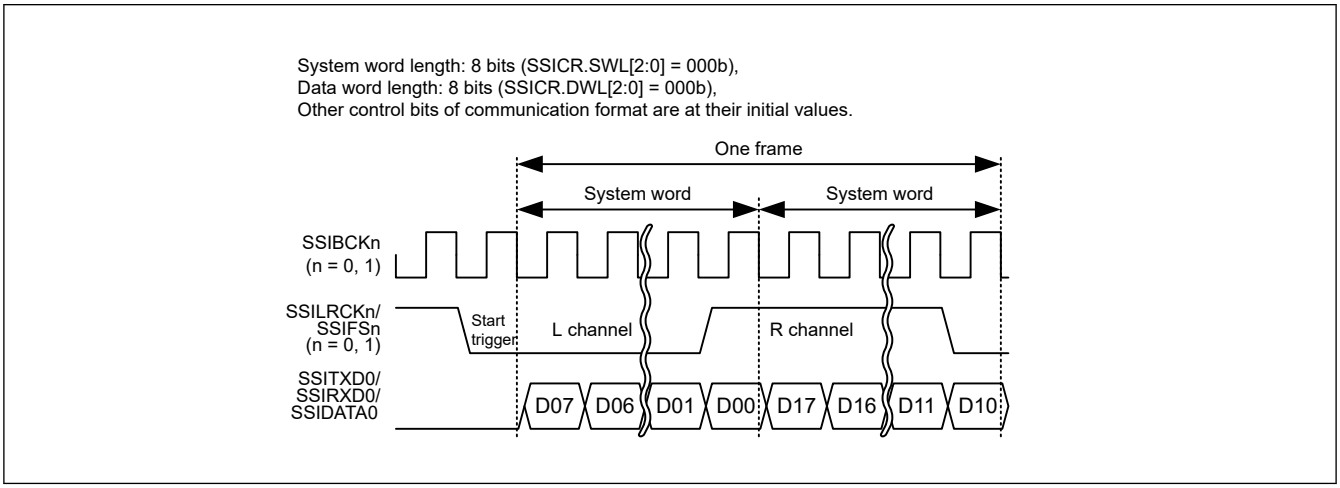


Figure 39.37 I²S format (without padding: system word length = data word length)

For the state of external pins when SSIE is in the idle state, see section 39.5.1. Idle State.

Note: SSIE has the SSILRCKn/SSIFSn (n = 0, 1) pin, which indicates the synchronization of communication. When SSIE is in slave mode (SSICR.MST = 0), the communication format SSIE uses must match that of the other-party device to communicate. SSIE uses the signal input by the SSILRCKn/SSIFSn (n = 0, 1) pin only as a trigger to start communication.

39.3.2 Monaural Format

The monaural format is a communication format used for connection with monaural-compatible serial devices. When the monaural format is specified (SSIOFR.OMOD[1:0] = 10b) for use, one frame consists of one system word. Also, a rising edge of the SSILRCKn/SSIFSn (n = 0, 1) signal indicates a communication start trigger. Figure 39.38 and Figure 39.39 respectively show the monaural formats without and with padding.

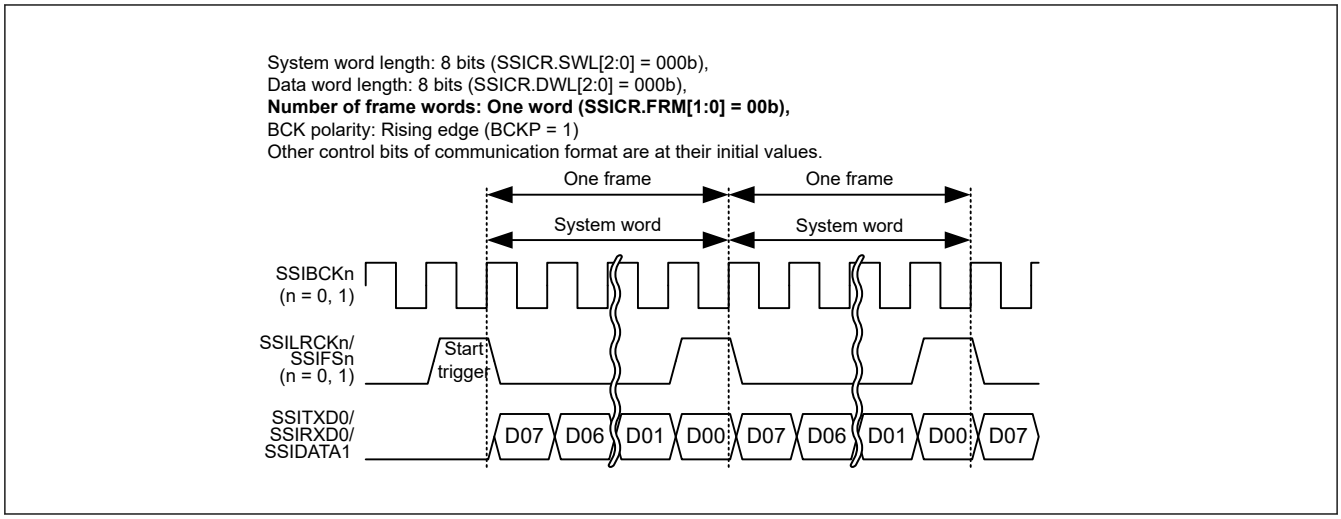


Figure 39.38 Short frame in monaural format (without padding: system word length = data word length)

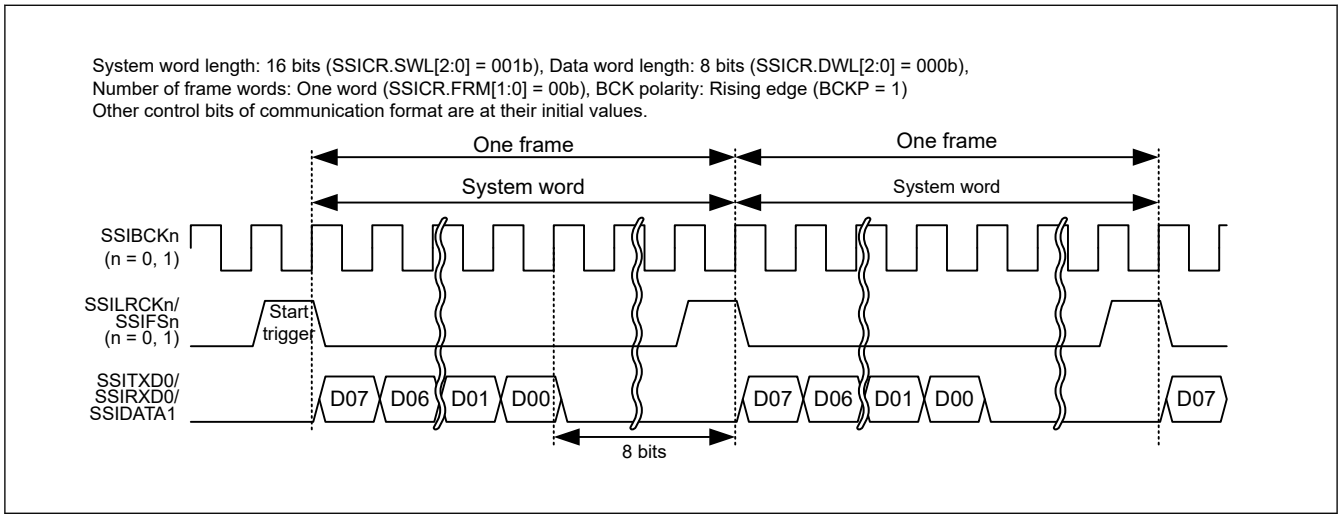


Figure 39.39 Short frame in monaural format (with padding: system word length > data word length)

The monaural formats supported by SSIE consist of short frames and long frames. See section 39.3.2.1. Short frame and section 39.3.2.2. Long frame for the difference between these two frames.

For the state of external pins state when SSIE is in the idle state, see section 39.5.1. Idle State.

Note: SSIE has the SSILRCKn/SSIFSn (n = 0, 1) pin, which indicates the synchronization of communication. When SSIE is in slave mode (SSICR.MST = 0), the communication format SSIE uses must match that of the other-party device to communicate. SSIE uses the signal input by the SSILRCKn/SSIFSn (n = 0, 1) pin only as a trigger to start communication.

39.3.2.1 Short frame

When a short frame is used (SSICR.DEL = 0), the SSILRCKn/SSIFSn (n = 0, 1) signal indicating the start of serial data is set to high level only for 1 cycle of SSIBCKn (n = 0, 1). Data transfer starts at the falling edge of the signal.

39.3.2.2 Long frame

When a long frame is used (SSICR.DEL = 1), the SSILRCKn/SSIFSn (n = 0, 1) signal indicating the start of serial data is set to high level only for 2 cycles of SSIBCKn (n = 0, 1). See Figure 39.40. Data transfer starts at the rising edge of the signal.

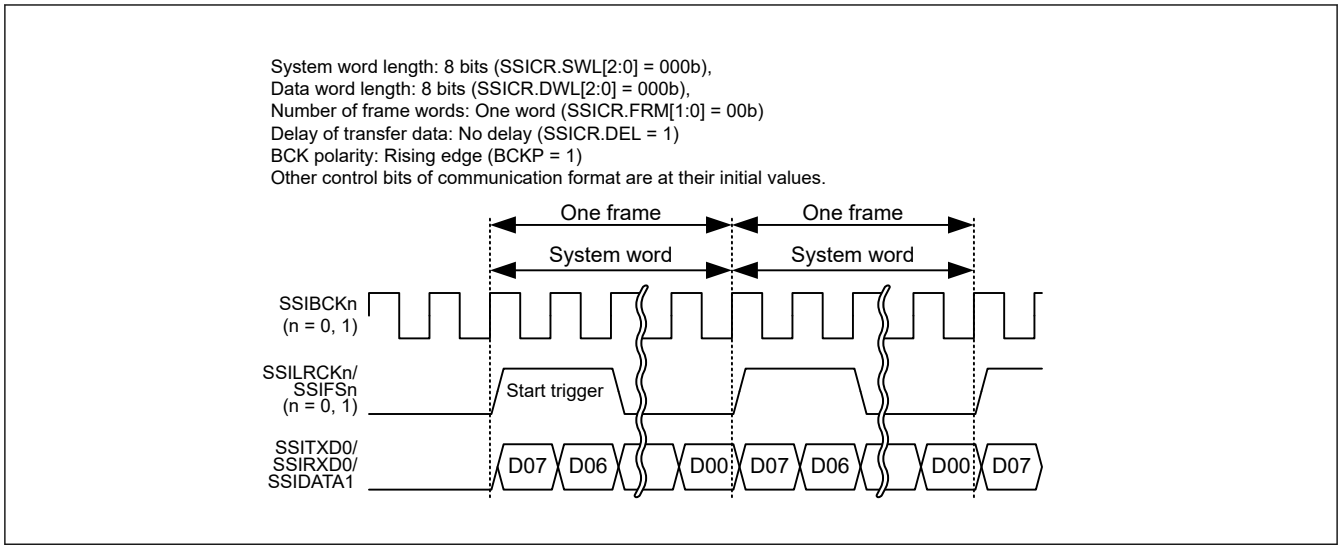


Figure 39.40 Long frame in monaural format (without padding)

39.3.3 TDM Format

The TDM format is a communication format used for connection with TDM-compatible multi-channel devices. With this format setting (SSIOFR.OMOD[1:0] = 01b), one frame is configured with four to eight system words set with the SSICR.FRM[1:0] bits. With this format, the SSILRCKn/SSIFSn (n = 0, 1) signal is at a high level for the first one system word and at a low level for the rest. The pulse generated on the SSILRCKn/SSIFSn (n = 0, 1) signal is defined as the SYNC pulse and its rising edge means a start of one frame. Figure 39.41 and Figure 39.42 respectively show the TDM formats without and with padding.

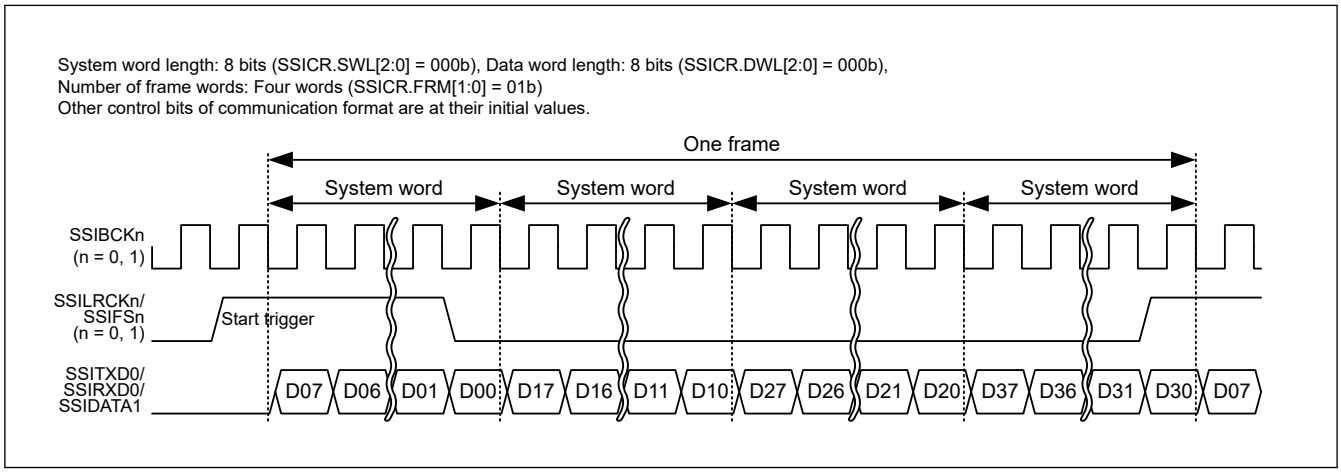


Figure 39.41 TDM format (without padding: system word length = data word length)

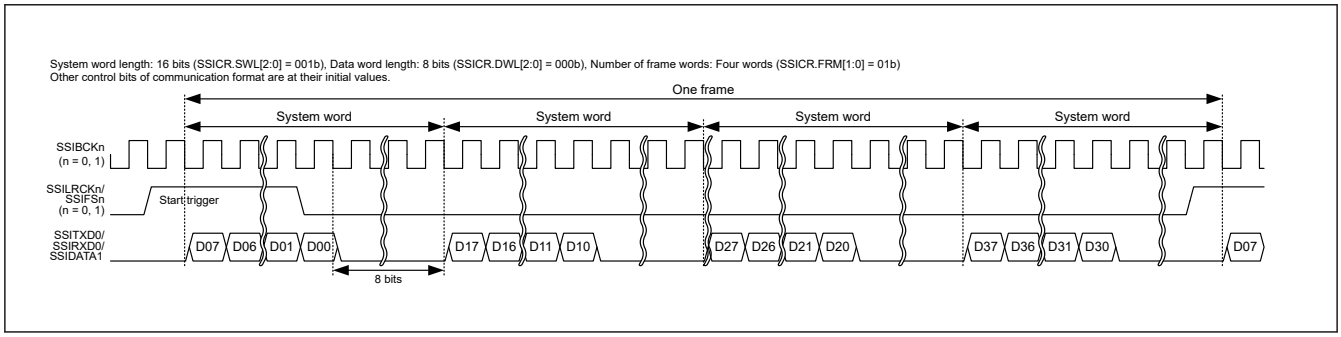


Figure 39.42 TDM format (with padding: system word length > data word length)

For the state of external pins when SSIE is in the idle state, see section 39.5.1. Idle State.

Note: SSIE has the SSILRCKn/SSIFSn (n = 0, 1) pin, which indicates the synchronization of communication. When SSIE is in slave mode (SSICR.MST = 0), the communication format SSIE uses must match that of the other-party device to communicate. SSIE uses the signal input by the SSILRCKn/SSIFSn (n = 0, 1) pin only as a trigger to start communication.

39.4 Communication Modes

SSIE supports the following communication modes. Table 39.15 lists the control bits that are not available with each communication mode. See section 39.4.1. Slave-mode Communication to section 39.4.5. Transmission and Reception for details of these communication modes.

Table 39.14 Communication modes

Communication Mode	SSICR.MST Bit	SSICR.REN Bit	SSICR.TEN Bit
Slave-mode transmission	0	0	1
Slave-mode reception	0	1	0
Slave-mode transmission and reception	0	1	1
Master-mode transmission	1	0	1
Master-mode reception	1	1	0
Master-mode transmission and reception	1	1	1

Table 39.15 Control bits that cannot be used in each communication mode

Control Bit	Slave-mode Reception	Slave-mode Transmission	Slave-mode Transmission and Reception	Master-mode Reception	Master-mode Transmission	Master-mode Transmission and Reception
SSICR.CKS	Invalid	Invalid	Invalid	Available	Available	Available
SSICR.CKDV	Invalid	Invalid	Invalid	Available	Available	Available
SSICR.MUEN	Invalid	Available	Available	Invalid	Available	Available
SSICR.TEN	Invalid	Available	Available	Invalid	Available	Available
SSICR.REN	Available	Invalid	Available	Available	Invalid	Available
SSIFCR.AUCKEN	Invalid	Invalid	Invalid	Available	Available	Available
SSIFCR.TIE	Invalid	Available	Available	Invalid	Available	Available
SSIFCR.RIE	Available	Invalid	Available	Available	Invalid	Available
SSIFCR.TFRST	Invalid	Available	Available	Invalid	Available	Available
SSIFCR.RFRST	Available	Invalid	Available	Available	Invalid	Available
SSIOFR.BCKASTP	Invalid	Invalid	Invalid	Available	Available	Available
SSIOFR.LRCONT	Invalid	Invalid	Invalid	Available	Available	Available
SSIOFR.OMOD	Available	Available	Available	Available	Available	Available
SSISCR.TDES	Invalid	Available	Available	Invalid	Available	Available
SSISCR.RDFS	Available	Invalid	Available	Available	Invalid	Available

“Invalid” means it has no effect on operation. Writing is possible.

39.4.1 Slave-mode Communication

SSIE operates in slave mode with SSICR.MST = 0. The SSIBCKn (n = 0, 1) and SSILRCKn/SSIFSn (n = 0, 1) signals to be used for serial-data communication must be supplied from an external device. If these signals do not match the communication format set for SSIE, operation is not guaranteed.

39.4.2 Master-mode Communication

SSIE operates in master mode with SSICR.MST = 1. The SSIBCKn (n = 0, 1) and SSILRCKn/SSIFSn (n = 0, 1) signals to be used for serial data communication must be internally generated from the audio clock. These signals use the format according to the setting of SSIE. If the communication format the slave device uses does not match the communication format set for SSIE, the operation is unpredictable.

39.4.3 Transmission

SSIE transmits serial data to the other-party device when the SSICR.TEN bit is 1 and the SSICR.REN bit is 0. If the communication format the other-party device uses does not match the communication format set for SSIE, the operation is unpredictable.

39.4.4 Reception

SSIE receives serial data from the other-party device when the SSICR.TEN bit is 0 and the SSICR.REN bit is 1. If the communication format the other-party device uses does not match the communication format set for SSIE, the operation is unpredictable.

39.4.5 Transmission and Reception

SSIE transmits and receives serial data to and from the other-party device when the SSICR.TEN bit is 1 and the SSICR.REN bit is 1. If the communication format the other-party device uses does not match the communication format set for SSIE, the operation is unpredictable.

39.5 Operation

SSIE has the following two main operation states Figure 39.43 shows SSIE state transition.

- Idle state (SSISR.IIRQ = 1)
- Communication state (SSISR.IIRQ = 0).

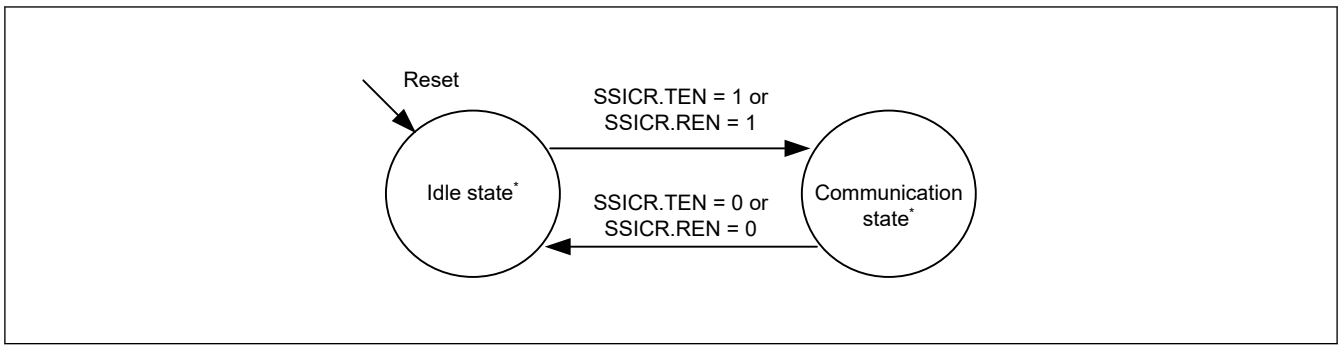


Figure 39.43 SSIE state transition

Note: See section 39.6.1. Start Communication for details of the idle state.
See section 39.6.2. Transmission for details of the communication state.

39.5.1 Idle State

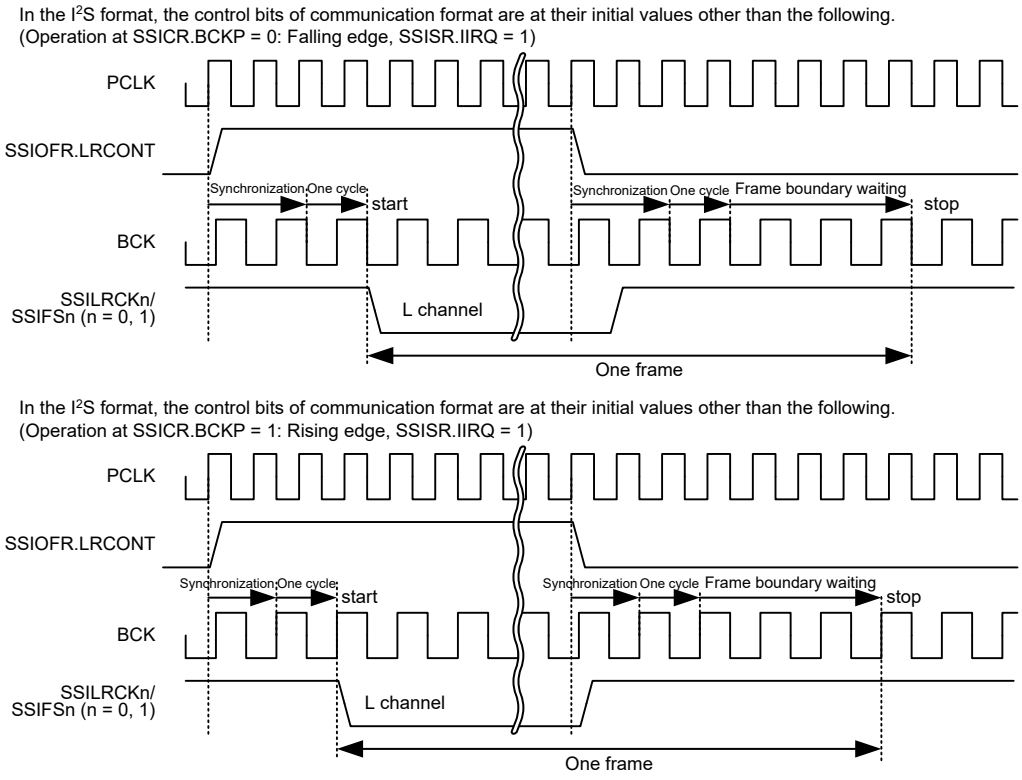
In this state, communication of SSIE is halted. If, however, the SSICR.MST bit is 1, output of the BCK and LR clock/ frame synchronization signals to external pins can be controlled according to the settings of SSIOFR.BCKASTP and SSIOFR.LRCONT bits. This function is common to all formats. For details, see Table 39.16.

Table 39.16 Output from external pins in the idle state (1 of 2)

SSICR.MST	SSIOFR.BCKASTP	SSIOFR.LRCONT	Output from Pins		
			SSIBCKn (n = 0, 1)	SSILRCKn/SSIFSn (n = 0, 1)	SSITXD0/SSIDATA1
0	—	—	Stop	Stop	Stop

Table 39.16 Output from external pins in the idle state (2 of 2)

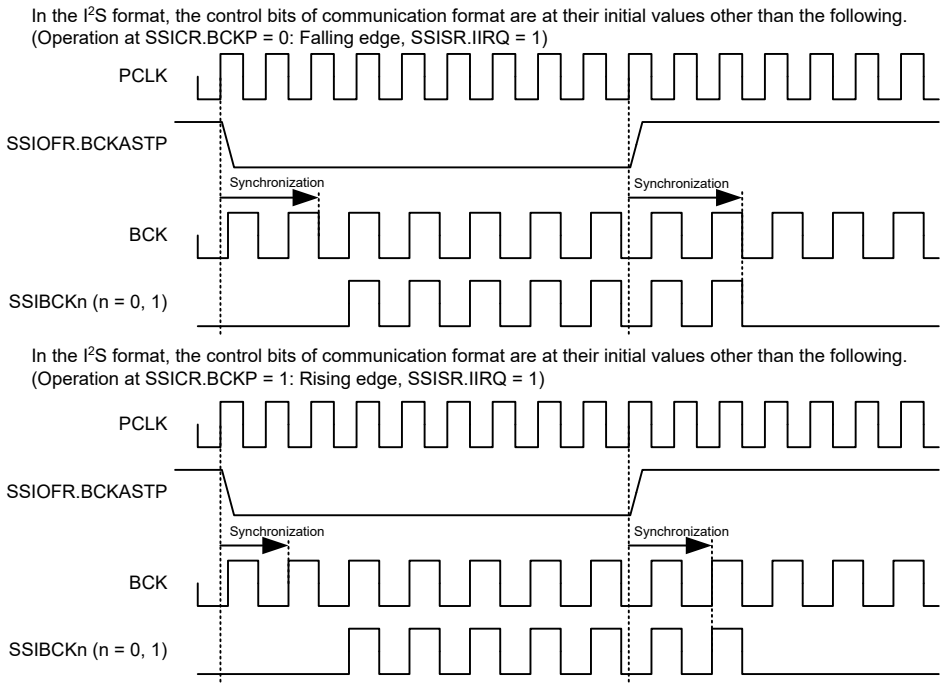
SSICR.MST	SSIOFR.BCKASTP	SSIOFR.LRCONT	Output from Pins		
			SSIBCKn (n = 0, 1)	SSILRCKn/SSIFSn (n = 0, 1)	SSITXD0/SSIDATA1
1	0	0	Supply	Stop	Stop
1	0	1	Supply	Supply	Stop
1	1	0	Stop	Stop	Stop
1	1	1	Stop	Supply	Stop



Note: PCLK = PCLKB

Figure 39.44 Example of disabling LR clock/frame synchronization continuation by SSIOFR.LRCONT

Note: To stop the output to the SSILRCKn/SSIFSn (n = 0, 1) pin with SSIOFR.LRCONT when SSIE is in the idle state in master-mode communication (SSICR.MST = 1), note the following: The output stops when the value of the SSIOFR.LRCONT bit is changed from 1 to 0. Make sure that the other-party device is not affected.



Note: PCLK = PCLKB

Figure 39.45 Example of stopping SSIBCKn (n = 0, 1) with SSIOFR.BCKASTP

Note: To stop the output to the SSIBCKn (n = 0, 1) pin with SSIOFR.BCKASTP in master-mode communication (SSICR.MST = 1) and while SSIE is in the idle state, note the following: The output stops when the value of the SSIOFR.BCKASTP bit is changed from 0 to 1. So, make sure that the other-party device is not affected.

39.5.2 Communication States

In this state, SSIE is during communication. Figure 39.46 shows transitions of communication states and Table 39.17 lists the conditions for transition. If the transition condition is not satisfied, the state does not transit.

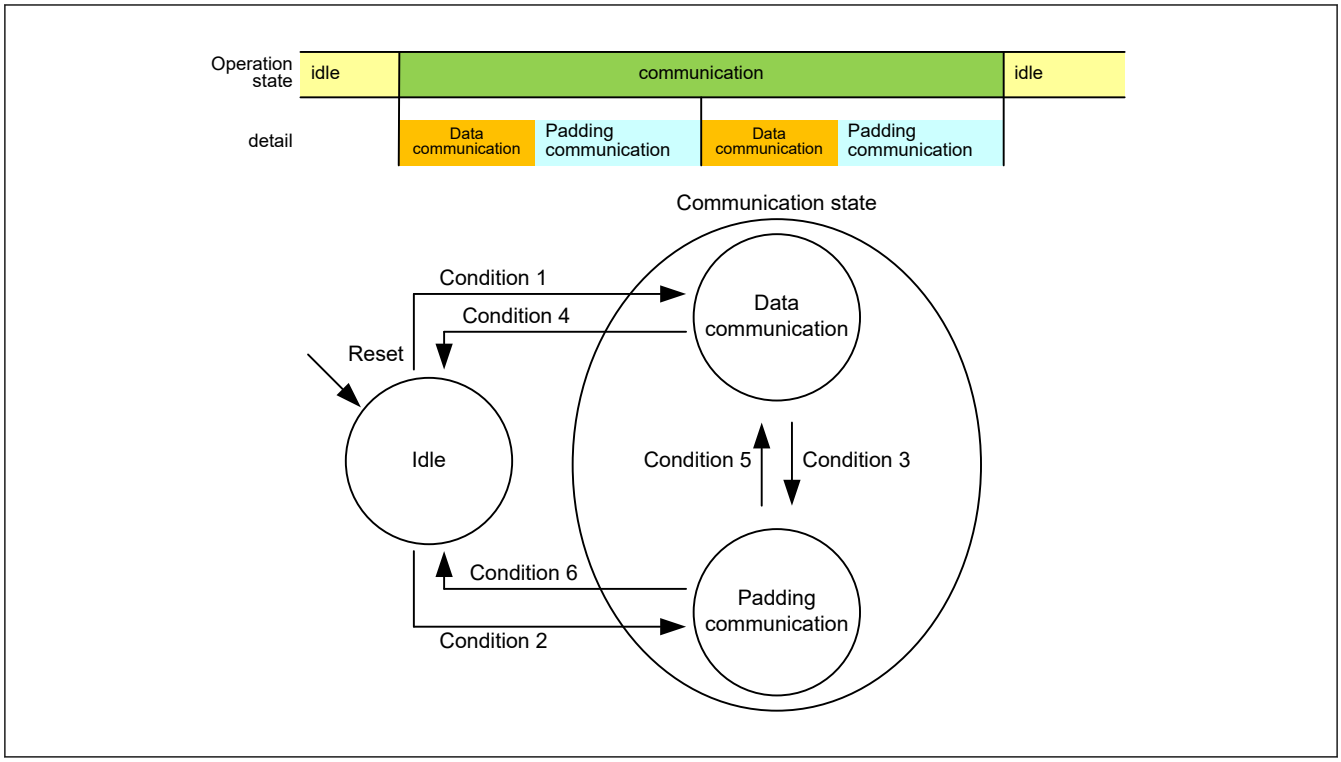


Figure 39.46 Communication state transition

Table 39.17 Condition for communication state transition

Condition Number	Condition for Transition
1	Writing SSICR.TEN = 1 or SSICR.REN = 1 while SSICR.SDTA = 0 or in the setting without padding bits.
2	Writing SSICR.TEN = 1 or SSICR.REN = 1 while SSICR.SDTA = 1 and in the setting with padding bits.
3	The following three conditions are all met: ● SSICR.TEN = 1 or SSICR.REN = 1 ● In the setting with padding bits ● The last bit of the data words has been transferred.
4	Both the following two conditions are met: ● SSICR.SDTA = 1 or without padding bits ● While SSICR.TEN = 0 and SSICR.REN = 0, the last bit of the data words in a frame has been transferred.
5	Transfer of the last padding bit is completed while SSICR.TEN = 1 or SSICR.REN = 1
6	Both the following two conditions are met: ● SSICR.SDTA = 0 and with padding bits ● While SSICR.TEN = 0 and SSICR.REN = 0, the last padding bit has been transferred.

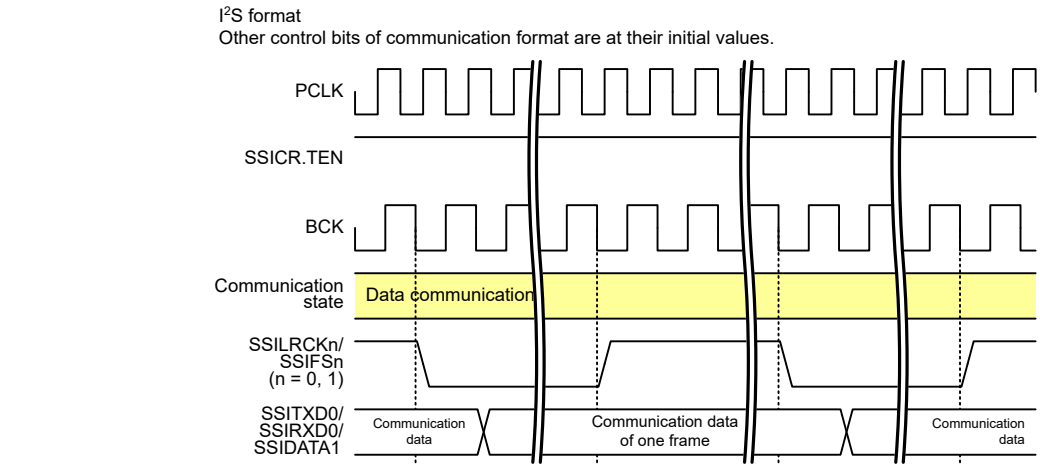
See Table 39.13 for the setting with/without padding bits.

39.5.2.1 Data communication state

In this state, SSIE is during communication. Data of the data word length set with the SSICR.DWL[2:0] bits is transmitted, received, or transmitted and received.

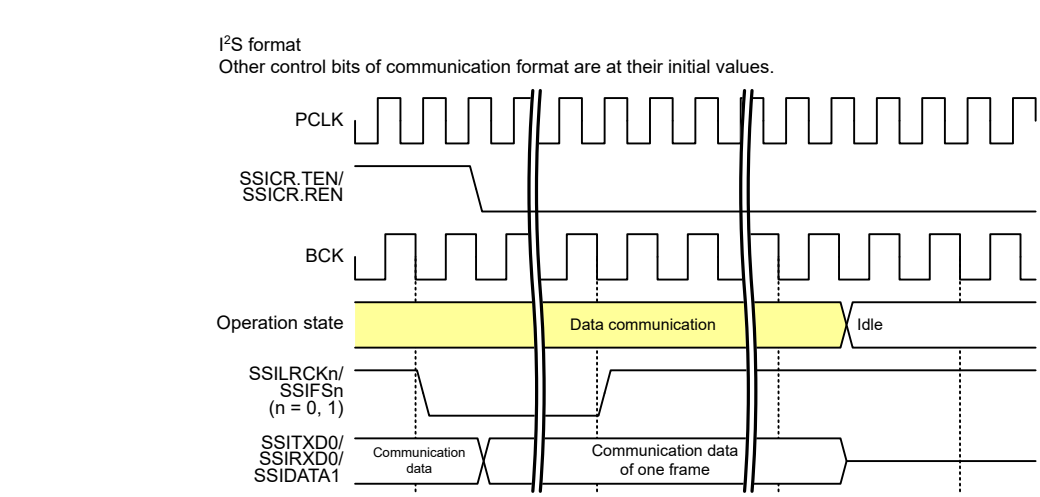
- State Transition in the Setting without Padding Bits

During communication (SSISR.IIRQ = 0), SSIE is during data communication for all the time. By disabling transmission and reception (SSICR.TEN = 0, SSICR.REN = 0), SSIE transits to the idle state. For details, see Figure 39.47 and Figure 39.48.



Note: PCLK = PCLKB

Figure 39.47 Continuation of the data communication



Note: PCLK = PCLKB

Figure 39.48 Halt from the data communication (without padding bits)

- State Transition in the Setting with Padding Bits

When SSIE ends transfer of the last bit of a data word during communication (SSISR.IIRQ = 0), SSIE transitions from the data communication state to the padding communication state in Figure 39.49. Except in the status with SSICR.SDTA = 1 and transmission and reception disabled (SSICR.TEN = 0 and SSICR.REN = 0), SSIE transitions from the data communication state to the idle state when it stops communication in Figure 39.50.

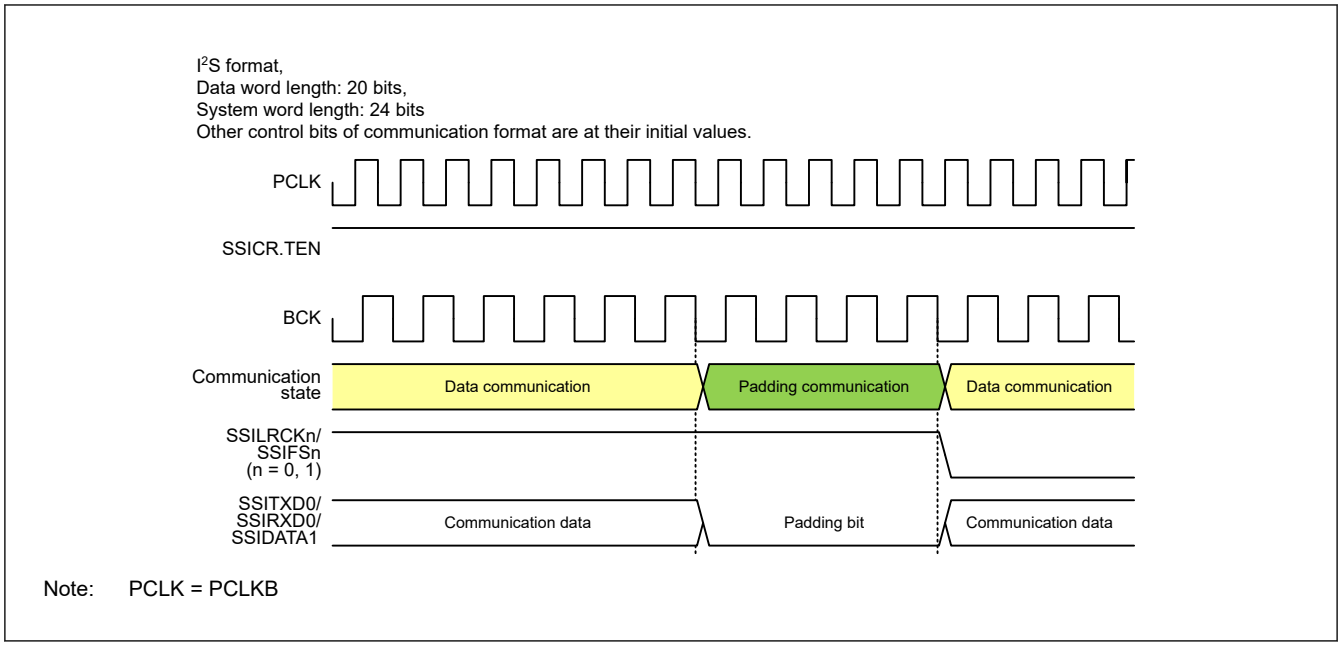


Figure 39.49 Transition from data communication to padding communication

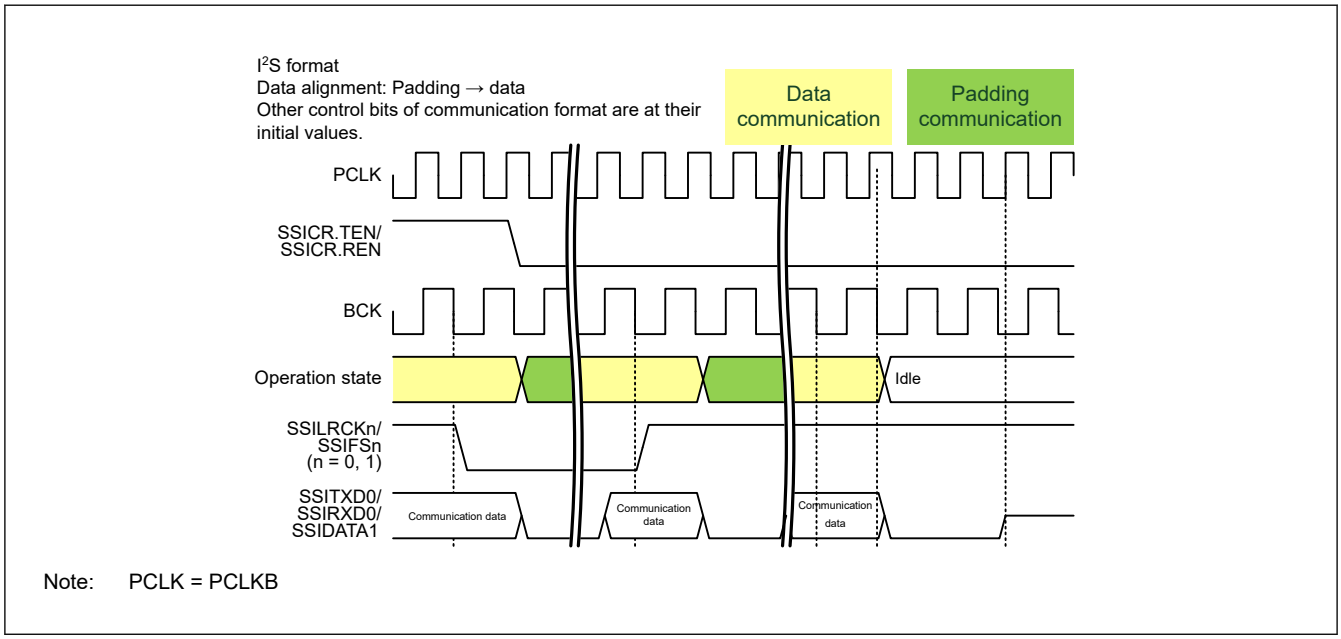


Figure 39.50 Halt from data communication (with padding bits)

39.5.2.2 Padding communication

In this state, SSIE is during communication. The padding bits set with the SSICR.SWL[2:0] bits and SSICR.DWL[2:0] bits are transmitted, received, or transmitted and received.

- State Transition in the Setting with Padding Bits

When SSIE ends transfer of the last padding bit during communication (SSISR.IIRQ = 0), SSIE transitions to the data communication state in Figure 39.49. If SSIE is in the status with SSICR.SDTA = 0 and transmission and reception disabled (SSICR.TEN = 0 and SSICR.REN = 0), SSIE transitions from the padding communication state to the idle state when it stops communication in Figure 39.51.

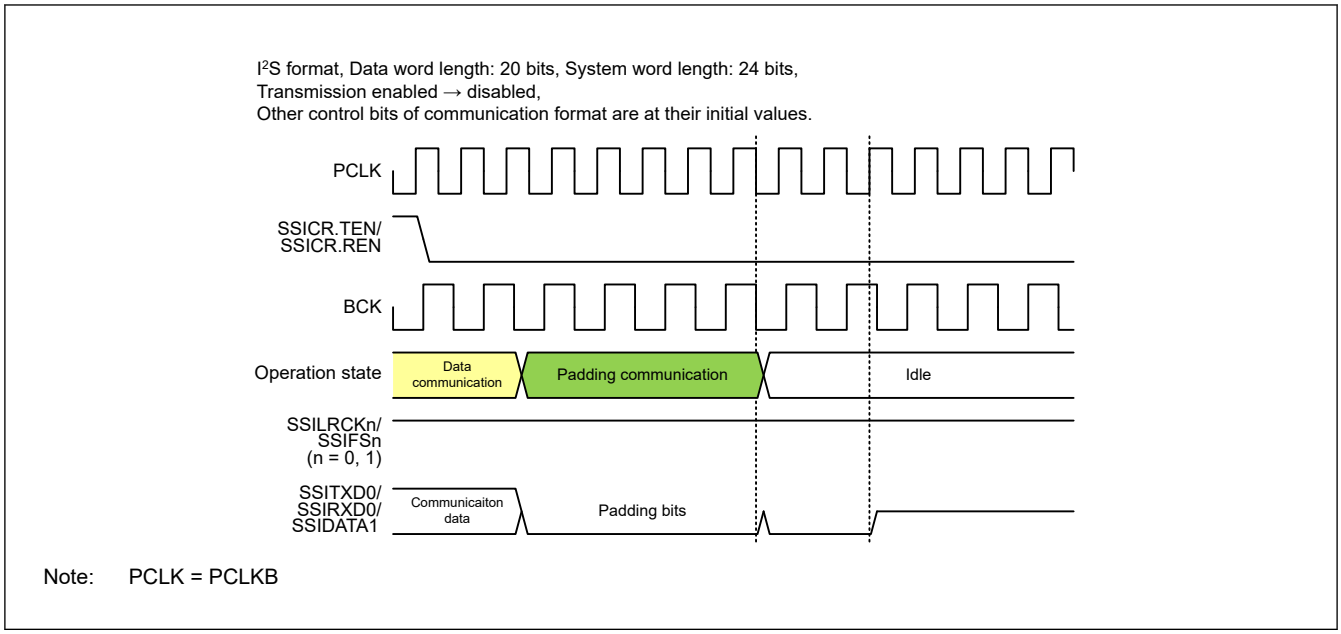


Figure 39.51 Halt from the padding communication

39.6 Communication Operation

Figure 39.52 shows the communication flow of SSIE.

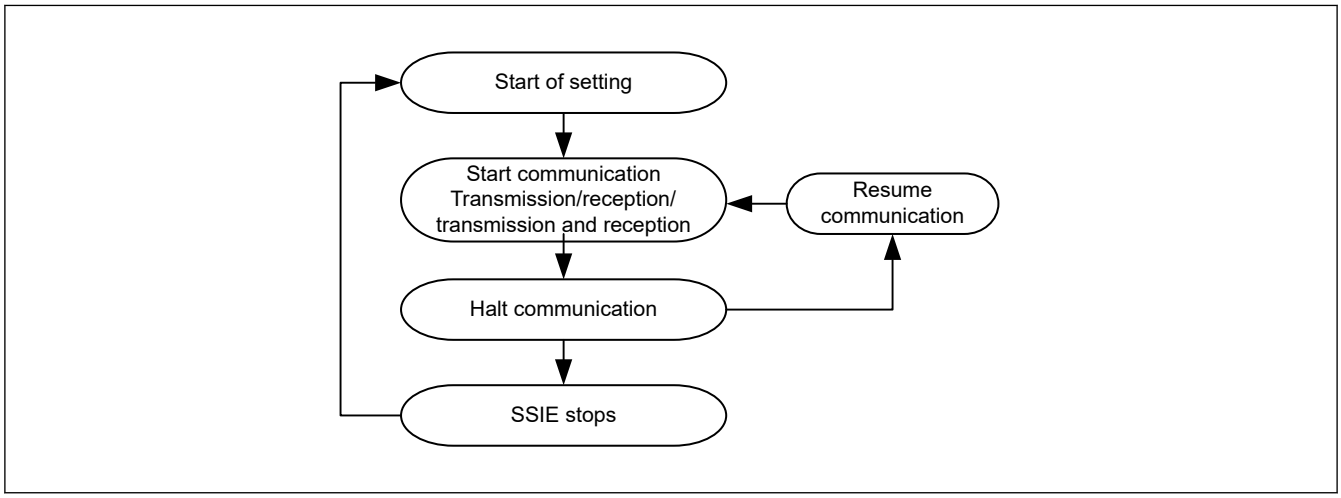
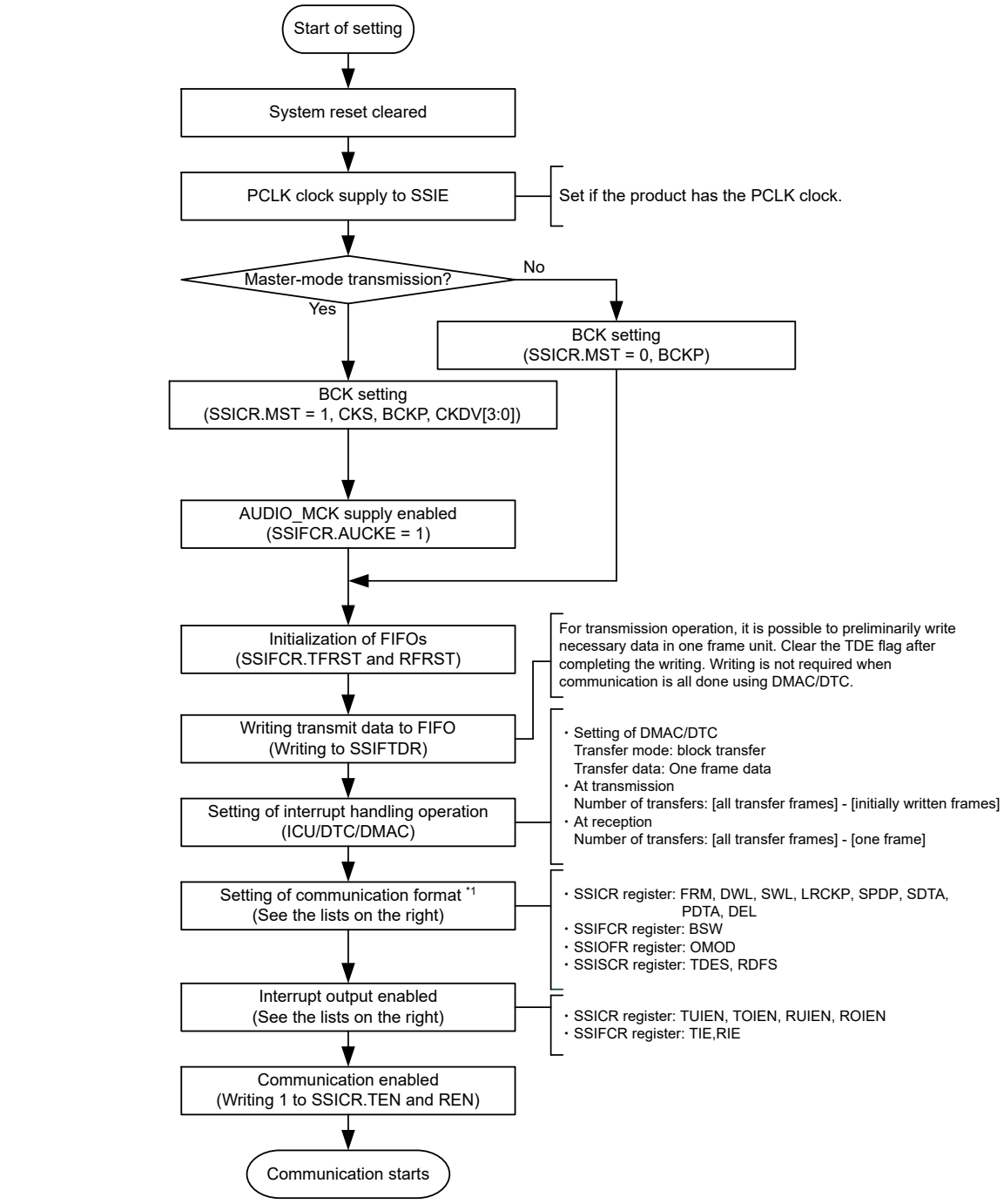


Figure 39.52 SSIE communication operation

The procedure of each operation is described from section 39.6.1. Start Communication to section 39.6.7. Resume Communication.

39.6.1 Start Communication

This section describes how to start communication of SSIE. Figure 39.53 shows the procedure to start communication. Be sure to follow the procedure. See section 39.6.2. Transmission for transmission operation and section 39.6.3. Reception for reception operation.



Note: PCLK = PCLKB
Note 1. Set SSICR.MUEN, SSIOFR.BCKASTP, and SSIOFR.LRCONT after completing the setting of the communication format.

Figure 39.53 Procedure to start communication (CPU operation procedure)

SSIE can perform continuous communication based on interrupts by the DTC/DMAC. For transmission, write 1 to SSIFCR.TIE, SSICR.TUIEN, and SSICR.TOIEN. For reception, write 1 to SSIFCR.RIE, SSICR.RUIEN, and SSICR.ROIEN.

39.6.2 Transmission

The transmission procedure in Figure 39.54 must be followed throughout a transmission operation.

After transmission is enabled (SSICR.TEN = 1 and SSICR.REN = 0), SSIE starts transmission when a start trigger is generated by SSILRCKn/SSIFSn (n = 0, 1) with the serial data for at least a frame contained in the transmit FIFO data register (SSIFTDR). SSIE outputs a transmit data empty interrupt to the DTC/DMAC according to the TDE setting condition (SSISCR.TDES) and the status of transmit data empty interrupt enable (SSIFCR.TIE) bit specified in the communication start procedure. This interrupt requests writing to the transmit FIFO data register (SSIFTDR). In the communication start procedure, specify writing to the transmit FIFO data register (SSIFTDR) as the DTC/DMAC operation in response to the transmit data empty interrupt. With this setting, SSIE can continuously transmit data not through the CPU. The transmit data empty interrupt is generated when the free space size of transmit FIFO data register reaches the value set in SSISCR.TDES. The number of times of writing must be specified in accordance with the free space size of the transmit FIFO data register indicated by the transmit data empty interrupt. If an error occurs, perform the error-handling procedure as instructed in the communication stop procedure.

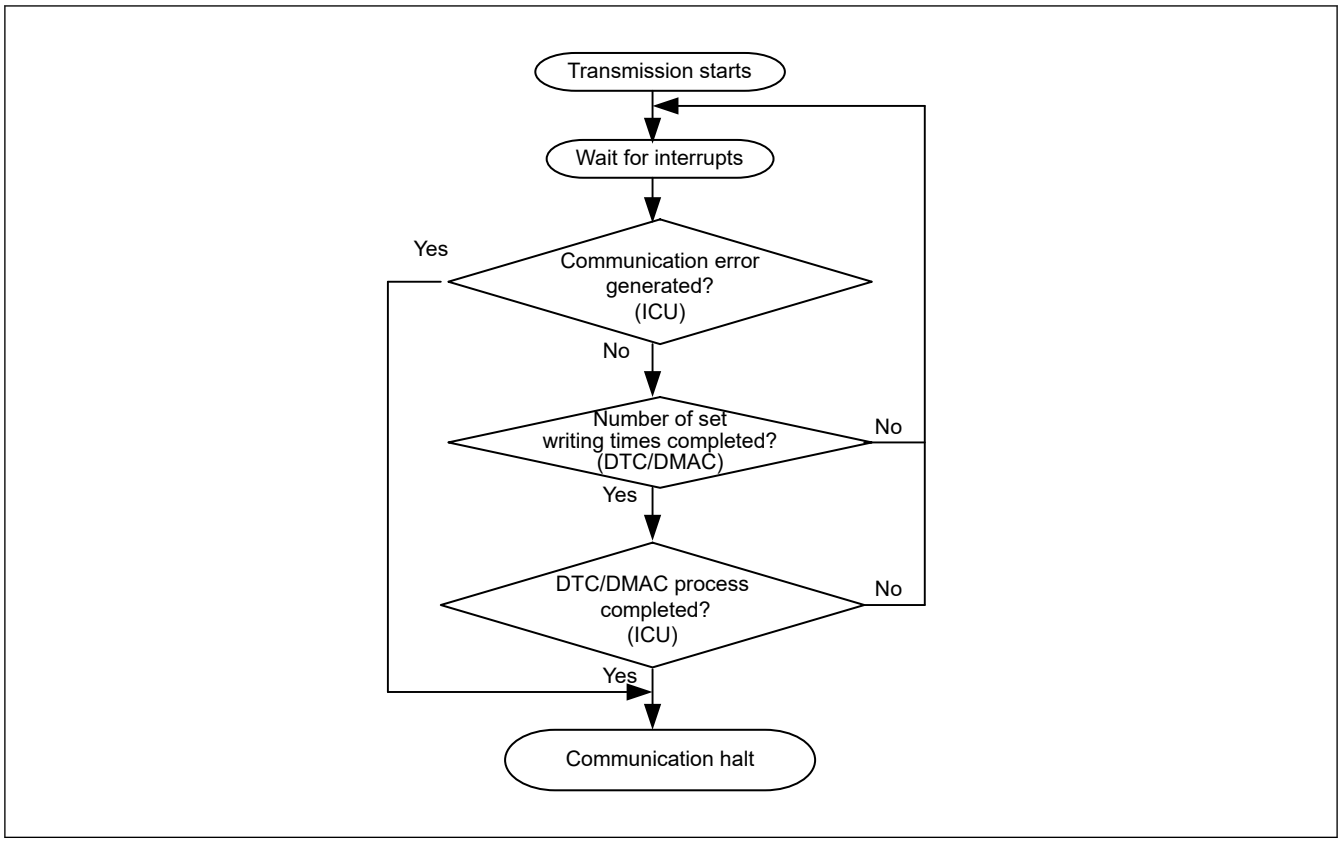


Figure 39.54 Transmission procedure

Note: The communication flow defined in SSIE uses the DTC/DMAC. If you do not use the DTC/DMAC, perform polling of the value 1 of SSIFSR.TDE to write data to SSIFTDR. The number of times of writing data to SSIFTDR by detecting the value 1 of SSIFSR.TDE must be in accordance with the free space size of the transmit FIFO data register specified by SSISCR.TDES. After as much transmit data as the free space size is written to SSIFTDR, the SSIFSR.TDE flag must be cleared. Continuous transmission is enabled by repeating data writing. If the SSIFSR.TDE flag is not cleared, the flag is not cleared automatically.

39.6.3 Reception

The reception procedure in Figure 39.55 must be followed throughout a reception operation.

After reception is enabled (SSICR.TEN = 0 and SSICR.REN = 1), SSIE starts reception when a start trigger is generated by SSILRCKn/SSIFSn (n = 0, 1). SSIE outputs a receive data full interrupt to the DTC/DMAC according to the RDF setting condition (SSISCR.RDFS) and the status of receive data full interrupt enable (SSIFCR.RIE) bit specified in the communication start procedure. This interrupt requests data reading from the receive FIFO data register (SSIFRDR). In the communication start procedure, specify reading from the receive FIFO data register (SSIFRDR) as the DTC/DMAC operation in response to the receive data full interrupt. With this setting, SSIE can continuously read data not through the CPU. The receive data full interrupt is generated when data as much as the capacity of receive FIFO data register has been stored. The number of times of reading must be specified in accordance with the data size of the receive FIFO data register

indicated by the receive data full interrupt. If an error occurs, perform the error-handling procedure as instructed in the communication stop procedure.

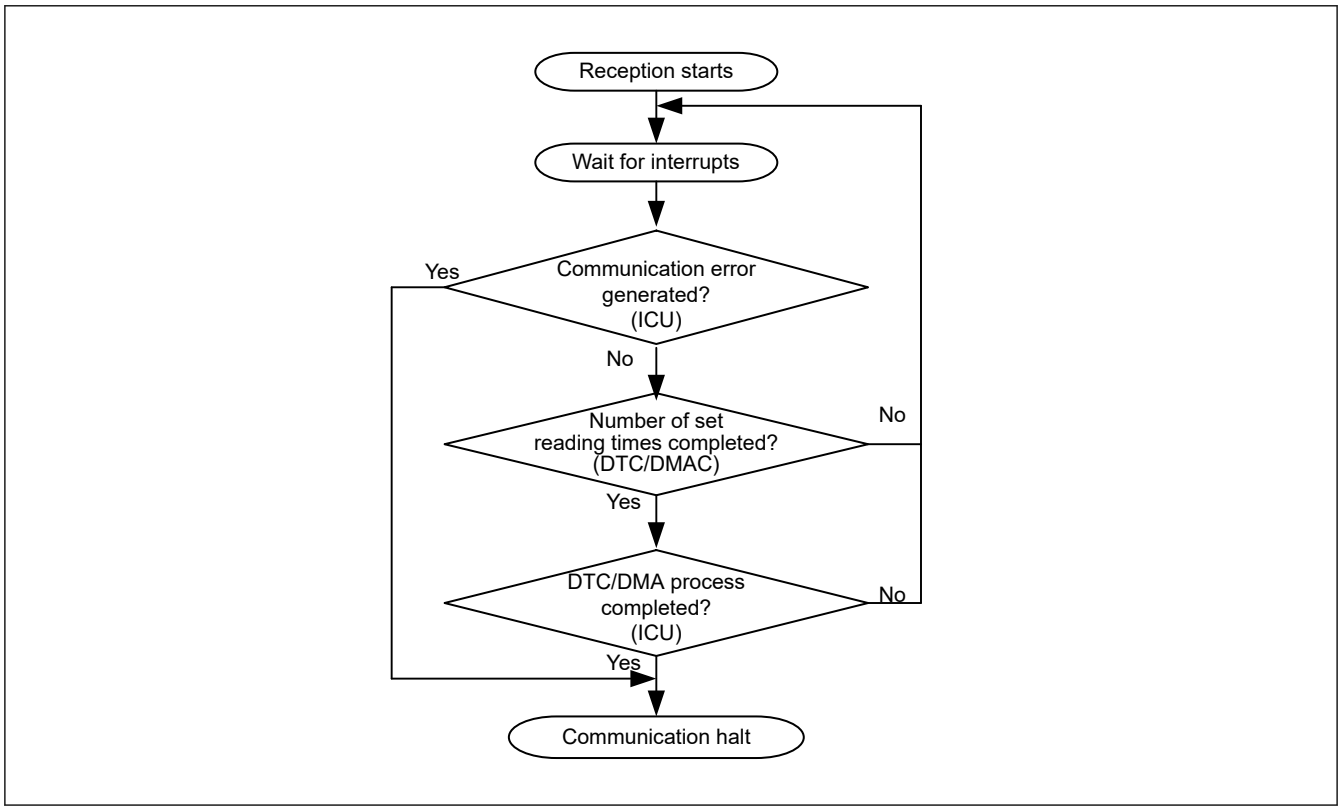


Figure 39.55 Reception procedure

Note: The communication flow defined in SSIE uses the DTC/DMAC. If you do not use the DTC/DMAC, perform polling of the value 1 of SSIFSR.RDF to read data from SSIFRDR. The number of times of reading data from SSIFRDR by detecting the value 1 of SSIFSR.RDF must be in accordance with the receive data storage capacity of the receive FIFO data register specified by SSISCR.RDFS. After received data is read from SSIFRDR, the SSIFSR.RDF flag must be cleared. Continuous reception is enabled by repeating data reading. If the SSIFSR.RDF flag is not cleared, the flag is not cleared automatically.

39.6.4 Transmission and Reception

After transmission and reception are enabled (SSICR.TEN = 1 and SSICR.REN = 1), SSIE starts transmission and reception when a start trigger is generated by SSILRCKn/SSIFSn (n = 0, 1) with the serial data for at least a frame contained in the transmit FIFO data register (SSIFTDR). SSIE can continuously transmit and receive data by performing the procedures described in section 39.6.2. Transmission and section 39.6.3. Reception, respectively. For how to stop transmission and reception, see section 39.6.5. Halt Communication.

39.6.5 Halt Communication

This section describes how to halt communication of SSIE. Figure 39.56 shows the procedure to halt communication. Be sure to follow the procedure.

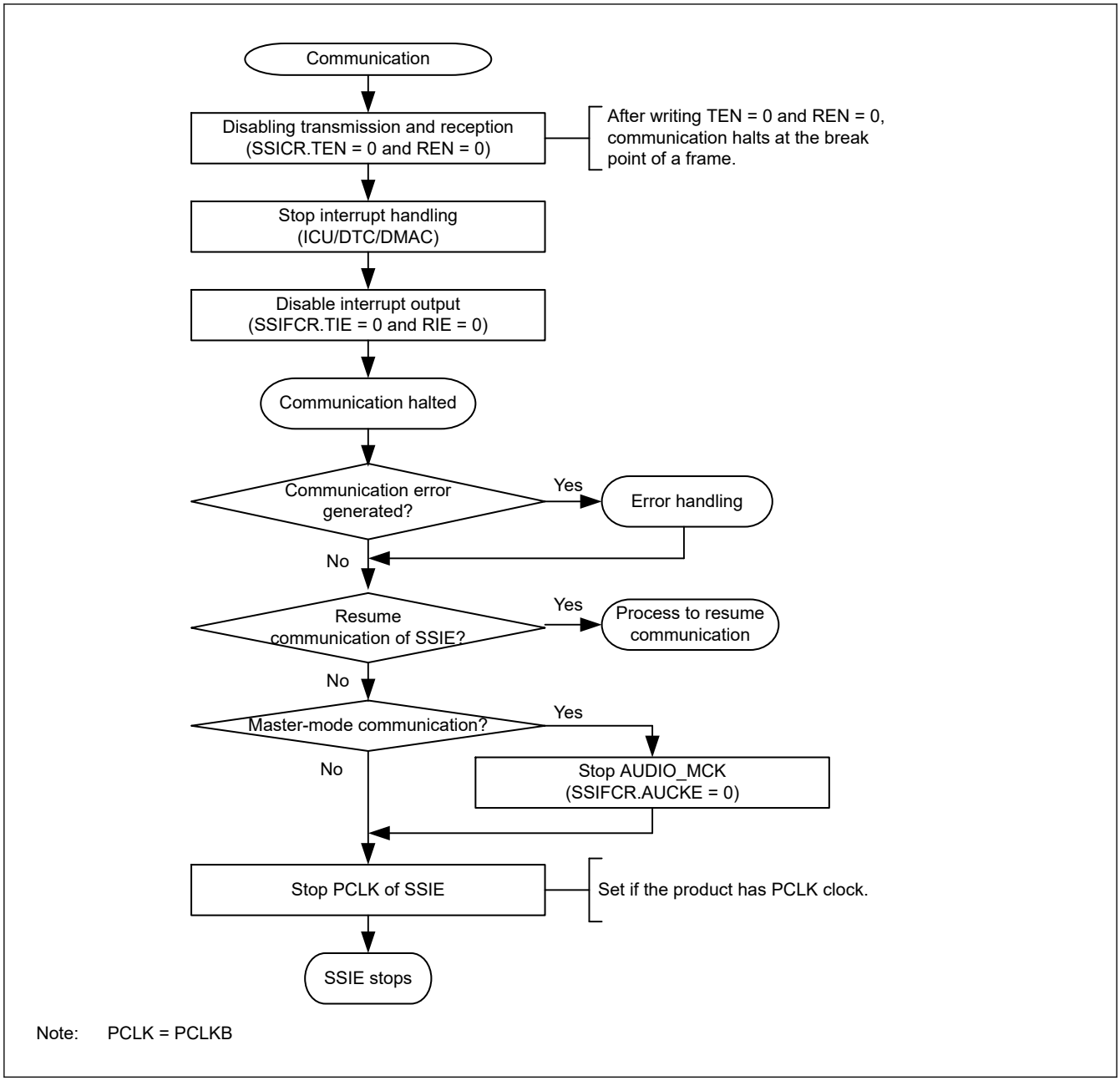


Figure 39.56 Procedure to halt communication (CPU operation procedure)

To halt the communication of SSIE, supply of the following clocks are required until the SSISR.IIRQ bit indicates an idle state.

- Input clock from the SSIBCKn (n = 0, 1) pin when SSICR.MST = 0
- AUDIO_MCK when SSICR.MST = 1
  To resume communication of SSIE in the previous setting, see section 39.6.7. Resume Communication.

Note: When communication of SSIE is halted according to the procedure to halt communication in Figure 39.56, resume communication according to the procedure to resume communication in Figure 39.58.

39.6.6 Error Handling

SSIE has the following four errors.

- Transmit underflow error
- Transmit overflow error
- Receive underflow error

- Receive overflow error.

When an underflow error or overflow error is generated, SSIE need to be restarted. Follow the procedure to halt communication in Figure 39.56 and error-handling procedure in Figure 39.57.

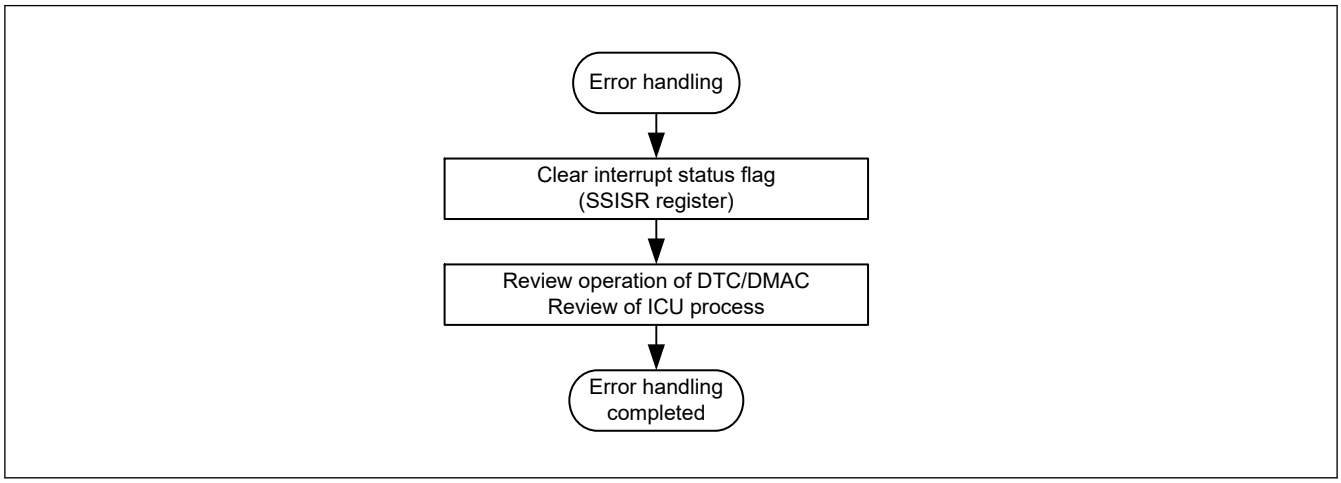


Figure 39.57 Error-handling procedure

Four error operations are described as follows. When the interrupt output enable bit of the SSICR register is enabled and error flags are set, an error interrupt is generated. See section 39.2.2. SSISR : Status Register for the setting conditions of error flags.

(1) Transmit Underflow Error

If a transmit underflow error occurs, review the number of times of writing data to the transmit FIFO data register (SSIFTDR) in response to a transmit data empty interrupt. After a transmit underflow error occurs, SSIE outputs 0s as data. To normally output the serial data written to the transmit FIFO data register (SSIFTDR) to the SSITXD0/SSIDATA1 pin, follow the procedure to halt communication in Figure 39.56 and error-handling procedure in Figure 39.57. After this error occurs, serial data is consumed as usual. If you resume communication, write the serial data from the beginning.

(2) Transmit Overflow Error

If a transmit overflow error occurs, review the number of times of writing data to the transmit FIFO data register (SSIFTDR) in response to transmit data empty interrupts. The serial data written to the transmit FIFO data register (SSIFTDR) that caused the transmit overflow error becomes invalid. This error can occur regardless of whether a transmission operation is being done. To recover from the error, follow the procedure to halt communication in Figure 39.56 and error-handling procedure in Figure 39.57. When you resume communication, deal with the invalid serial data appropriately.

(3) Receive Underflow Error

If a receive underflow error occurs, review the number of times of reading data from the receive FIFO data register (SSIFRDR) in response to receive data full interrupts. The values read from the receive FIFO data register (SSIFRDR) that caused the receive underflow error are undefined. This error can occur regardless of whether a reception operation is being done. To recover from the error, follow the procedure to halt communication in Figure 39.56 and error-handling procedure in Figure 39.57.

(4) Receive Overflow Error

If a receive overflow error occurs, review the number of times of reading data from the receive FIFO data register (SSIFRDR) in response to receive data full interrupts. The receive data that caused the receive overflow error cannot be stored in the receive FIFO data register (SSIFRDR). To recover from the error, follow the procedure to halt communication in Figure 39.56 and error-handling procedure in Figure 39.57.

39.6.7 Resume Communication

When you resume the communication using SSIE, follow the communication resume procedure in Figure 39.58. The communication resume procedure is designed on the assumption that you resume the communication stopped by the

communication stop procedure without changing any settings. If you want to change clock and slave/master settings, use and follow the communication start procedure in Figure 39.53. For details about the transmission operation and reception operation after starting communication, see section 39.6.2. Transmission and section 39.6.3. Reception, respectively.

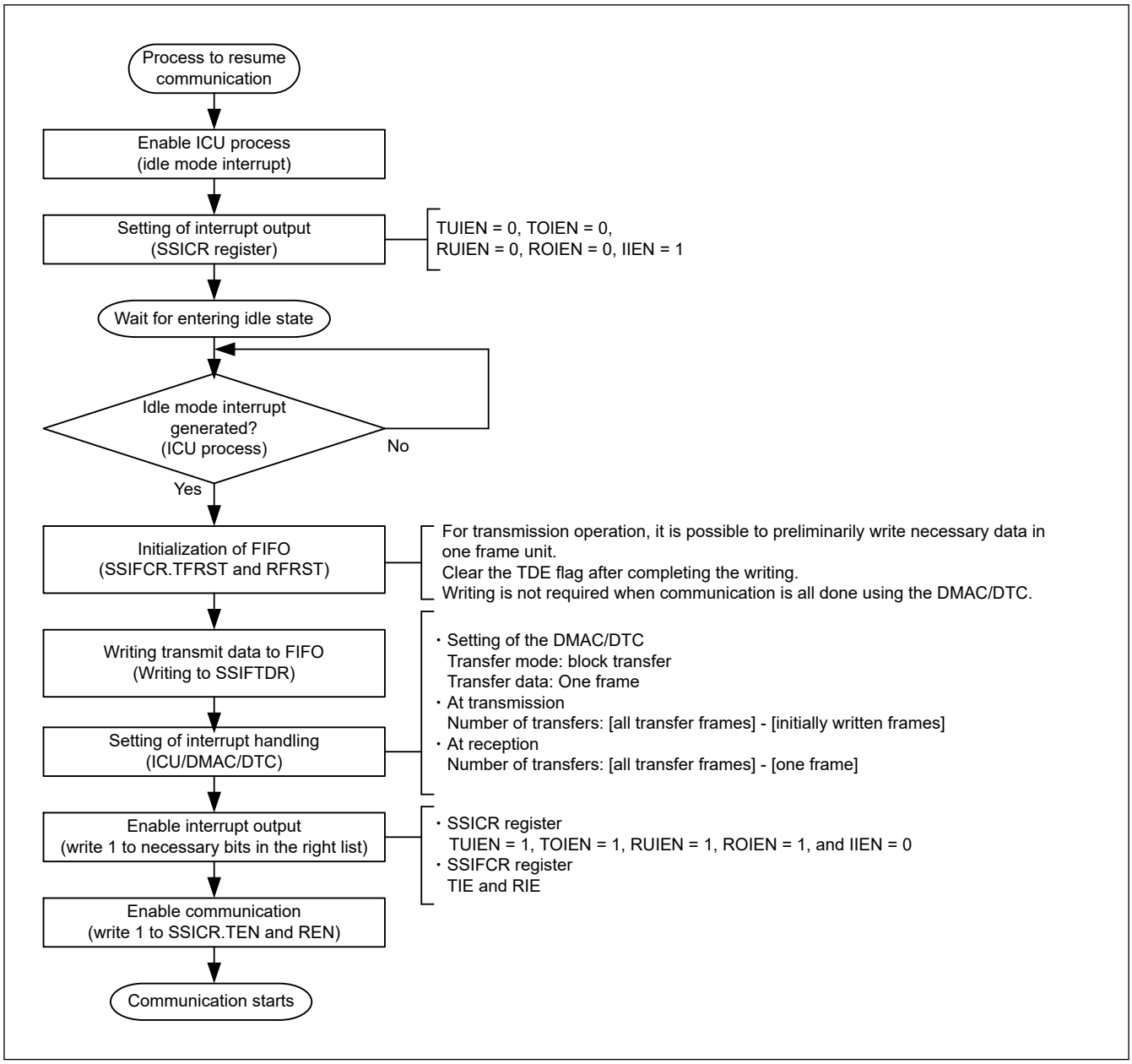


Figure 39.58 Procedure to resume communication (CPU operation procedure)

39.7 Interrupts

Table 39.18 lists the interrupt sources. Set enable/disable of interrupt output of each source with the TUIEN, TOIEN, RUIEN, ROIEN, and IIEN bits in the SSICR register and the TIE and RIE bits in the SSIFCR register.

Table 39.18 SSIE interrupt sources

Channel	Interrupt source	Description	Interrupt flag	DMAC/DTC activation
SSIE0	SSIE0_SSIF	• Transmit underflow interrupt • Transmit overflow interrupt • Receive underflow interrupt • Receive overflow interrupt • Idle interrupt	SSISR.TUIRQ SSISR.TOIRQ SSISR.RUIRQ SSISR.ROIRQ SSISR.IIRQ	Not possible
	SSIE0_SSIRXI	Receive data full interrupt	SSIFSR.RDF	Possible
	SSIE0_SSITXI	Transmit data empty interrupt	SSIFSR.TDE	Possible
SSIE1	SSIE1_SSIF	• Transmit underflow interrupt • Transmit overflow interrupt • Receive underflow interrupt • Receive overflow interrupt • Idle interrupt	SSISR.TUIRQ SSISR.TOIRQ SSISR.RUIRQ SSISR.ROIRQ SSISR.IIRQ	Not possible
	SSIE1_SSIRT	• Receive data full interrupt • Transmit data empty interrupt	SSIFSR.RDF SSIFSR.TDE	Possible

39.7.1 SSIEn_SSIF Interrupt (n = 0, 1)

This interrupt source combines five interrupts. Enable output of necessary interrupts before using SSIE. The five interrupts are operated by using the flags assigned to individual interrupts and interrupt output enable bits. To clear an interrupt, set the interrupt enable to 0 or clear the interrupt flag to 0.

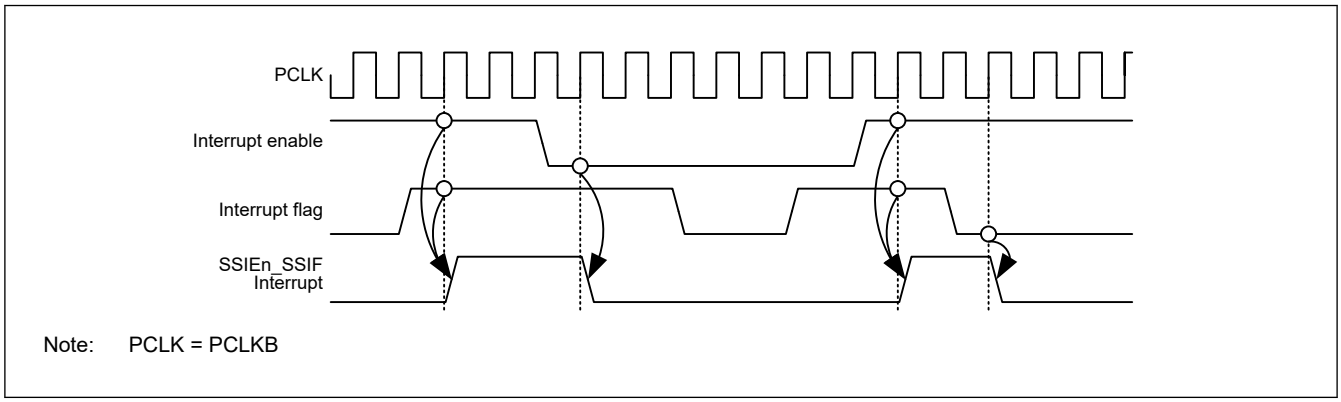


Figure 39.59 Timing Diagram of the common interrupt source, SSIEn_SSIF

- Transmit underflow interrupt

As the transmit underflow interrupt, SSISR.TUIRQ is output while SSICR.TUIEN = 1. When you use SSIE for transmission, enable the output of this interrupt (SSICR.TUIEN = 1). If this interrupt occurs, follow instructions in the procedure to halt communication in Figure 39.56 and error-handling procedure in Figure 39.57.

- Transmit overflow interrupt

As the transmit overflow interrupt, SSISR.TOIRQ is output while SSICR.TOIEN = 1. When you use SSIE for transmission, enable the output of this interrupt (SSICR.TOIEN = 1). If this interrupt occurs, follow instructions in the procedure to halt communication in Figure 39.56 and error-handling procedure in Figure 39.57.

- Receive underflow interrupt

As the receive underflow interrupt, SSISR.RUIRQ is output while SSICR.RUIEN = 1. When you use SSIE for reception, enable the output of this interrupt (SSICR.RUIEN = 1). If this interrupt occurs, follow instructions in the procedure to halt communication in Figure 39.56 and error-handling procedure in Figure 39.57.

- Receive overflow interrupt

As the receive overflow interrupt, SSISR.ROIRQ is output while SSICR.ROIEN = 1. When you use SSIE for reception, enable the output of this interrupt (SSICR.ROIEN = 1). If this interrupt occurs, follow instructions in the procedure to halt communication in Figure 39.56 and error-handling procedure in Figure 39.57.

- Idle mode interrupt

As the idle mode interrupt, SSISR.IIRQ is output while SSICR.IIEN = 1. This interrupt is used to make sure that communication has stopped fully.

39.7.2 SSIE0_SSITXI Interrupt [Full-duplex communication]

The transmit data empty interrupt is a pulse interrupt that is output when the following condition is met:

- SSIFCR.TIE = 1 and SSIFSR.TDE = 1
  SSIE operation: When the value of SSIFSR.TDE changes from 0 to 1 while the value of SSIFCR.TIE is 1
  CPU instruction: When the value of SSIFCR.TIE changes from 0 to 1 while the value of SSIFSR.TDE is 1

This interrupt is subject to the interrupt suppression function. If an interrupt condition for this interrupt occurs when the DTC/DMAC is busy (when the DTC/DMAC cannot accept interrupts), the interrupt suppression function holds the output of this interrupt. The held interrupt will be output after the DTC/DMAC is enabled to accept interrupts. For details, see Figure 39.60.

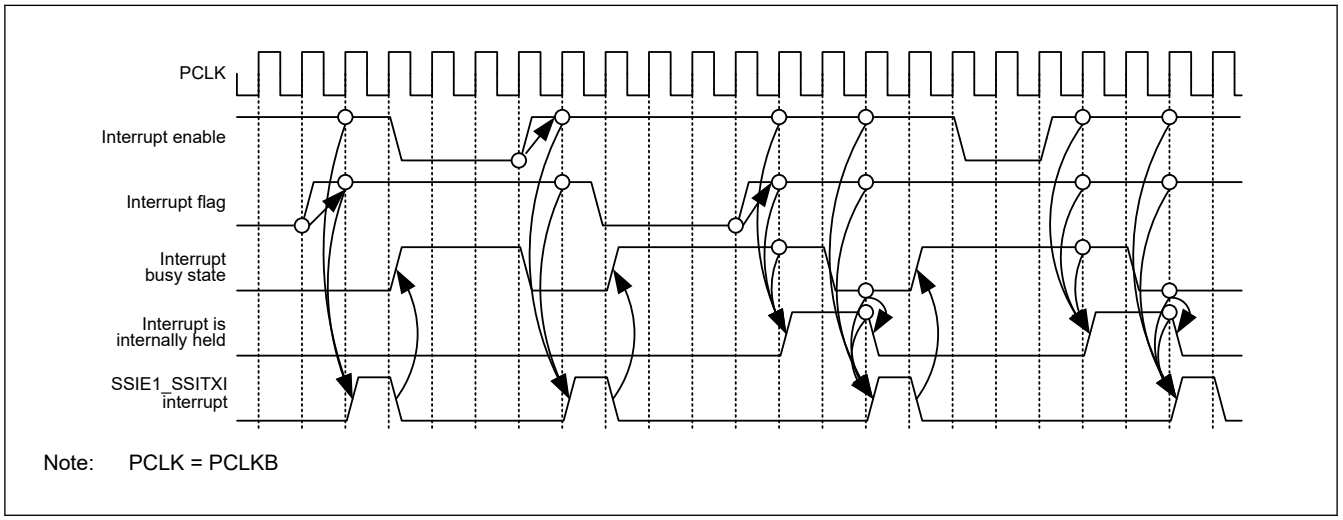


Figure 39.60 SSIE1_SSITXI interrupt timing diagram

39.7.3 SSIE0_SSIRXI Interrupt [Full-duplex communication]

The receive data full interrupt is a pulse interrupt that is output when the following condition is met:

- SSIFCR.RIE = 1 and SSIFSR.RDF = 1.
  SSIE operation: When the value of SSIFSR.RDF changes from 0 to 1 while the value of SSIFCR.RIE is 1
  CPU instruction: When the value of SSIFCR.RIE changes from 0 to 1 while the value of SSIFSR.RDE is 1

This interrupt is subject to the interrupt suppression function. If an interrupt condition for this interrupt occurs when the DTC/DMAC is busy (when the DTC/DMAC cannot accept interrupts), the interrupt suppression function holds the output of this interrupt. The held interrupt will be output after the DTC/DMAC is enabled to accept interrupts. The behavior of this interrupt is the same as the behavior shown in Figure 39.60.

39.7.4 SSIE1_SSIRT Interrupt [Half-duplex communication]

This interrupt is output by two sources, transmit data empty interrupt and receive data full interrupt. When this interrupt is generated, read the interrupt flag and specify the interrupt source.

This interrupt is subject to the interrupt suppression function. If an interrupt condition for this interrupt occurs when the DTC/DMAC is busy (when the DTC/DMAC cannot accept interrupts), the interrupt suppression function holds the output of this interrupt. The held interrupt will be output after the DTC/DMAC is enabled to accept interrupts. For details, see Figure 39.61.

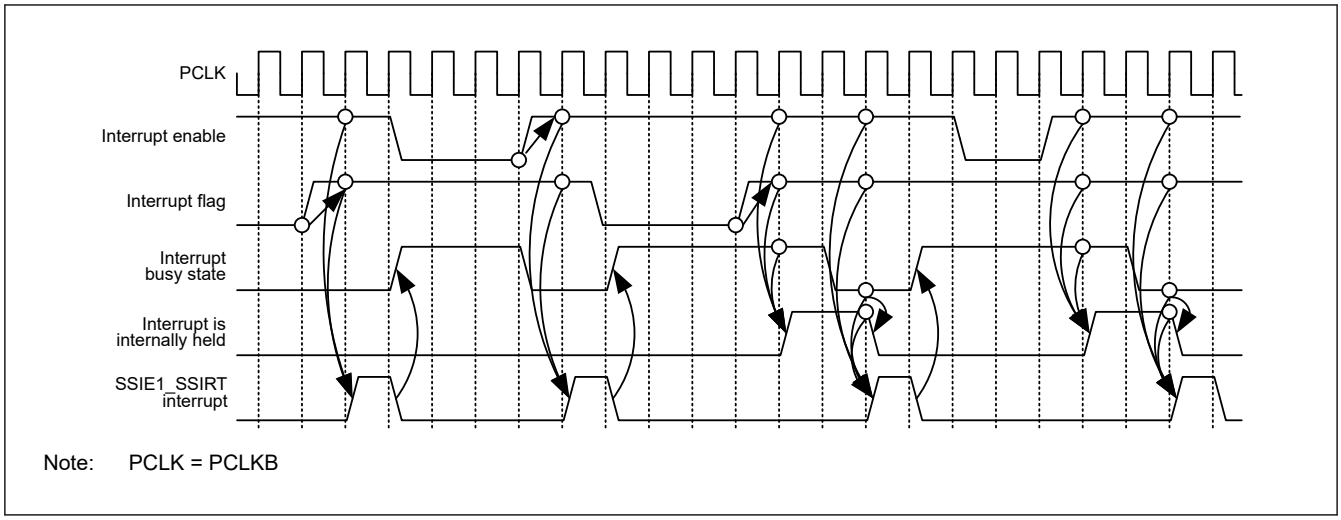


Figure 39.61 SSIE1_SSIRT interrupt timing diagram

39.8 Software Resets

SSIE has three software reset bits to reset its states.

- SSIE software reset (SSIFCR.SSIRST)
- Transmit FIFO data register reset (SSIFCR.TFRST)
- Receive FIFO data register reset (SSIFCR.RFRST).

This section describes the procedures for the three types of software resets.

39.8.1 Software Reset Procedure

(1) SSIE Software Reset

For the SSIE software reset bit (SSIFCR.SSIRST), follow the procedure shown in Figure 39.62. After a reset, the same setting is applied when it is resumed. To change the settings of clocks and slave/master mode, follow the procedure to start communication in Figure 39.53. See section 39.6.2. Transmission and section 39.6.3. Reception respectively for transmission and reception after communication is resumed.

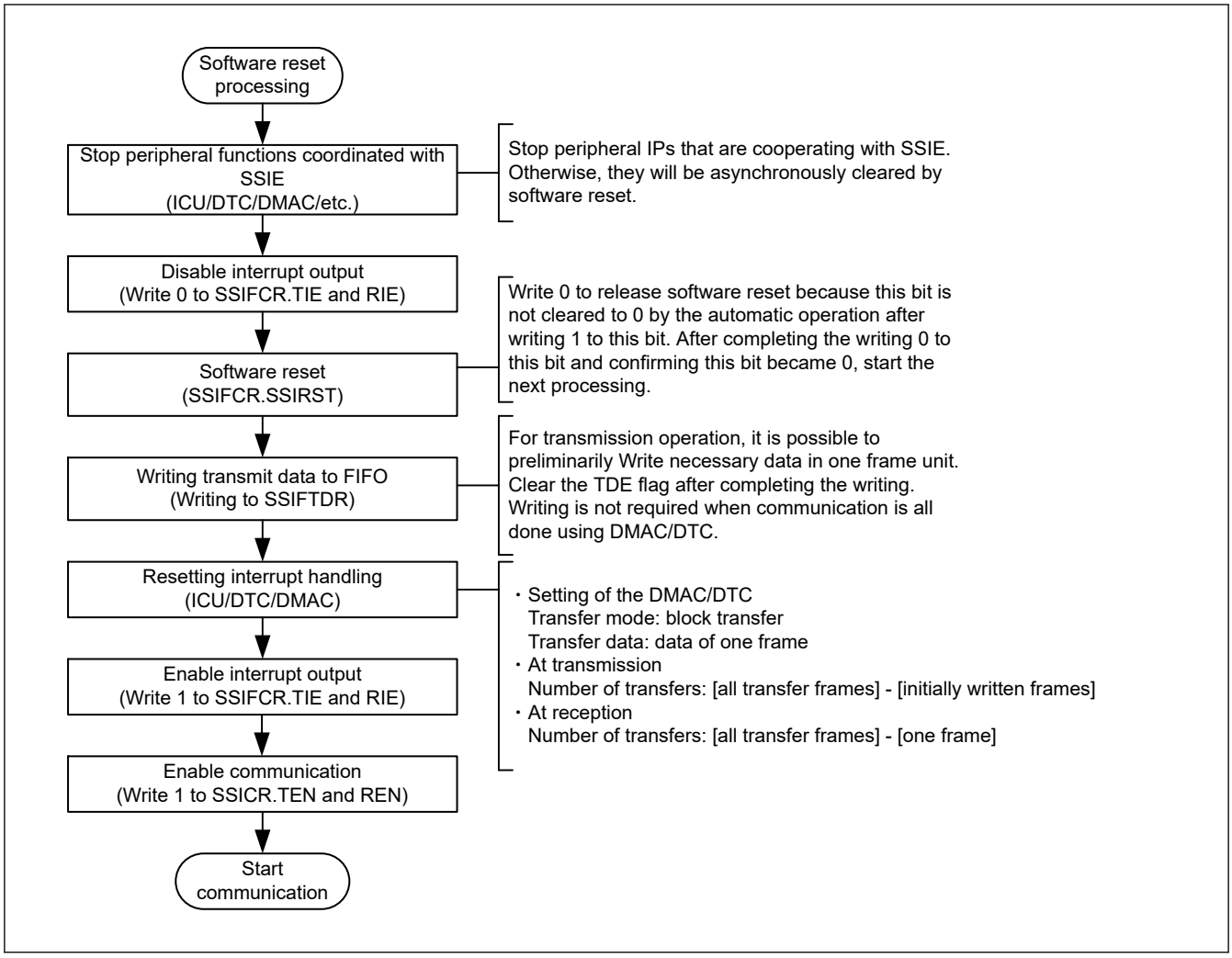


Figure 39.62 Software reset procedure (CPU operation procedure)

(2) Transmit FIFO data register reset

To perform a transmit FIFO data register reset, follow instructions in the procedure to start communication in Figure 39.53 and procedure to resume communication in Figure 39.58.

(3) Receive FIFO data register reset

To perform a receive FIFO data register reset, follow instructions in the procedure to start communication in Figure 39.53 and procedure to resume communication in Figure 39.58.

39.9 Usage Notes

39.9.1 Notes for Slave-mode Communication

39.9.1.1 SSIBCKn (n = 0, 1) control

In slave-mode communication (SSICR.MST = 0), SSIE needs supply of SSIBCKn (n = 0, 1). To stop BCK on the master side, make sure that SSIE is in the idle state (SSISR.IIRQ = 1). If BCK is stopped before SSIE becomes idle, take the procedure to start communication in Figure 39.53 or wait for an idle state by taking the procedure to resume communication in Figure 39.58.

39.9.1.2 SSILRCKn/SSIFSn (n = 0, 1) pin

SSIE has the SSILRCKn/SSIFSn (n = 0, 1) pin, which indicates the synchronization of communication. When SSIE is in slave mode (SSICR.MST = 0), the communication format SSIE uses must match that of the other-party device to communicate. SSIE uses the signal input by the SSILRCKn/SSIFSn (n = 0, 1) pin only as a trigger to start communication.

39.9.2 Notes for Master-mode Communication

39.9.2.1 AUCKE control

In master-mode communication (SSICR.MST = 1), SSIE operates with the audio clock (AUDIO_MCK). To stop SSIE completely, make sure that SSIE is in the idle state (SSISR.IIRQ = 1) and then write 0 to SSIFCR.AUCKE.

39.9.2.2 LRCONT control

To stop the output to the SSILRCKn/SSIFSn (n = 0, 1) pin with SSIOFR.LRCONT when SSIE is in the idle state in master-mode communication (SSICR.MST = 1), note the following: The output stops when the value of the SSIOFR.LRCONT bit is changed from 1 to 0. Make sure that the other-party device is not affected. For details, see Figure 39.44.

39.9.2.3 BCKASTP control

To stop the output to the SSIBCKn (n = 0, 1) pin with SSIOFR.BCKASTP in master-mode communication (SSICR.MST = 1) and while SSIE is in the idle state, note the following: The output stops when the value of the SSIOFR.BCKASTP bit is changed from 0 to 1. So, make sure that the other-party device is not affected. For details, see Figure 39.45.

The BCKASTP bit cannot be used when the other-party device (which is a slave) requires the clock output from the SSIBCKn (n = 0, 1) pin before and during communication.

39.9.3 Notes for Communication Flow

39.9.3.1 When an error interrupt is generated

SSIE has the following four errors.

- Transmit underflow error
- Transmit overflow error
- Receive underflow error
- Receive overflow error

When an underflow error or overflow error is generated, SSIE need to be restarted. Follow the procedure to halt communication in Figure 39.56 and error-handling procedure in Figure 39.57.

(1) Transmit Underflow Error

If a transmit underflow error occurs, review the number of times of writing data to the transmit FIFO data register (SSIFTDR) in response to a transmit data empty interrupt. After a transmit underflow error occurs, SSIE outputs 0s as data. To normally output the serial data written to the transmit FIFO data register (SSIFTDR) to the SSITXD0/SSIDATA1 pin, follow the procedure to halt communication in Figure 39.56 and error-handling procedure in Figure 39.57. After this error occurs, serial data is consumed as usual. If you resume communication, write the serial data from the beginning.

(2) Transmit Overflow Error

If a transmit overflow error occurs, review the number of times of writing data to the Transmit FIFO Data Register (SSIFTDR) in response to transmit data empty interrupts. The serial data written to the Transmit FIFO Data Register (SSIFTDR) that caused the transmit overflow error becomes invalid. This error can occur regardless of whether a transmission operation is being done. To recover from the error, follow the procedure to halt communication in Figure 39.56 and error-handling procedure in Figure 39.57. When you resume communication, deal with the invalid serial data appropriately.

(3) Receive Underflow Error

If a receive underflow error occurs, review the number of times of reading data from the receive FIFO data register (SSIFRDR) in response to receive data full interrupts. The values read from the receive FIFO data register (SSIFRDR) that caused the receive underflow error are undefined. This error can occur regardless of whether a reception operation is being done. To recover from the error, follow the procedure to halt communication in Figure 39.56 and error-handling procedure in Figure 39.57.

(4) Receive Overflow Error

If a receive overflow error occurs, review the number of times of reading data from the receive FIFO data register (SSIFRDR) in response to receive data full interrupts. The receive data that caused the receive overflow error cannot be stored in the receive FIFO data register (SSIFRDR). To recover from the error, follow the procedure to halt communication in Figure 39.56 and error-handling procedure in Figure 39.57.

39.9.3.2 Transmit data empty interrupt

The communication flow defined in SSIE uses the DTC/DMAC. If you do not use the DTC/DMAC, perform polling of the value 1 of SSIFSR.TDE to write data to SSIFTDR. The number of times of writing data to SSIFTDR by detecting the value 1 of SSIFSR.TDE must be in accordance with the free space size of the transmit FIFO data register specified by SSISCR.TDES. After as much transmit data as the free space size is written to SSIFTDR, the SSIFSR.TDE flag must be cleared. Continuous transmission is enabled by repeating data writing. If the SSIFSR.TDE flag is not cleared, the flag is not cleared automatically.

39.9.3.3 Receive data full interrupt

The communication flow defined in SSIE uses the DTC/DMAC. If you do not use the DTC/DMAC, perform polling of the value 1 of SSIFSR.RDF to read data from SSIFRDR. The number of times of reading data from SSIFRDR by detecting the value 1 of SSIFSR.RDF must be in accordance with the receive data storage capacity of the receive FIFO data register specified by SSISCR.RDFS. After received data is read from SSIFRDR, the SSIFSR.RDF flag must be cleared. Continuous reception is enabled by repeating data reading. If the SSIFSR.RDF flag is not cleared, the flag is not cleared automatically.

39.9.3.4 Switching transfer modes

1. For state transition from transmission, reception, and transmission and reception, disable transmission and reception (SSICR.TEN = 0, SSICR.REN = 0).
2. Confirm it is in the idle state (SSISR.IIRQ = 1).
3. In the idle state, set the SSICR.TEN bit or the SSICR.REN bit again and resume transfer.

39.9.3.5 Resume communication after halting SSIE

When communication of SSIE is halted according to the procedure to halt communication in Figure 39.56, resume communication according to the procedure to resume communication in Figure 39.58.

39.9.4 Write Access Restriction

39.9.4.1 SSICR register

If the TEN bit or REN bit is rewritten, make sure that the SSISR.IIRQ bit is in the desired status. If the value of the TEN or REN bit is changed by rewriting, subsequent operation is unpredictable. For example, when transmission or reception is enabled, check that SSISR.IIRQ is 0; when transmission or reception is disabled, check that SSISR.IIRQ is 1.

(1) TEN Bit and REN Bit

These bits enable/disable transmission and reception. When 1 is written to one of these bits, the corresponding communication operation starts in synchronization with a start trigger by the SSILRCKn/SSIFSn (n = 0, 1) signal. For details, see section 39.6.2. Transmission, section 39.6.3. Reception, and section 39.6.4. Transmission and Reception. When 0 is written to this bit, the current communication operation stops at the next frame boundary. To use SSIE for both transmission and reception, always write 1 to these bits together. When stopping the communication using SSIE, always disable both transmission and reception (write 0 to the TEN and REN bits).

39.9.4.2 SSISR register

(1) Clearing TUIRQ and TOIRQ

After communication is enabled (by changing the value of SSICR.TEN bit from 0 to 1), the transmission error flags (TOIRQ and TUIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the transmission error flags might be unable to be read.

(2) Clearing RUIRQ and ROIRQ

After communication is enabled (by changing the value of SSICR.REN bit from 0 to 1), the reception error flags (RUIRQ and ROIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the reception error flags might be unable to be read.

39.9.4.3 Communication state

Writing to the bits with orange-shaded area in Table 39.19 is prohibited. If written, the operation performed immediately after writing is not guaranteed.

Table 39.19 Bits protected from writing during communication

Symbol	Address (BASE+)		+0								+1							
			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSICR	0x00	+0	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	FRM[1:0]		DWL[2:0]			SWL[2:0]		
		+2	—	MST	BCKP	LRCKP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
SSISR	0x04	+0	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SSIFCR	0x10	+0	AUCKE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRST
		+2	—	—	—	—	BSW	—	—	—	—	—	—	—	TIE	RIE	TFRST	RFRST
SSIFSR	0x14	+0	—	—	TDC[5:0]						—	—	—	—	—	—	—	TDE
		+2	—	—	RDC[5:0]						—	—	—	—	—	—	—	RDF
SSIFTDR	0x18	+0	SSIFTDR[31:16]															
		+2	SSIFTDR[15:0]															
SSIFRDR	0x1C	+0	SSIFRDR[31:16]															
		+2	SSIFRDR[15:0]															
SSIOFR	0x20	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	BCKASTP	LRCONT	—	—	—	—	—	—	OMOD[1:0]	
SSISCR	0x24	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]				

39.9.5 Module-stop function

SSIE operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The SSIE module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

40. SD/MMC Host Interface (SDHI)

40.1 Overview

The Secure Digital Host Interface (SDHI) and MultiMediaCard (MMC) Interface provide the functionality required to connect a variety of external memory cards with the MCU. The SDHI supports both 1- and 4-bit buses for connecting different memory cards that support SD, SDHC, and SDXC formats. When developing host devices that are compliant with the SD Specifications, you must comply with the SD Host/Ancillary Product License Agreement (SD HALA).

The MMC interface supports 1-bit, 4-bit, and 8-bit MMC buses that provide eMMC 4.51 (JEDEC Standard JESD 84- B451) device access. This interface also provides backward compatibility and supports for high-speed SDR transfer modes.

Table 40.1 lists the SD/MMC Host Interface specifications and Figure 40.1 shows a block diagram.

Table 40.1 SD/MMC Host Interface specifications

Interface	Parameter	Specifications
SD	SD bus interface	• Compatible with SD memory card and SDIO card • Transfer bus mode selectable from 4-bit wide bus mode or 1-bit default bus mode • Compatible with SD, SDHC, and SDXC formats
	Transfer modes	Default Speed mode, High Speed mode, SDR12 or SDR25
SD and MMC shared	SDHI clock frequency	The SDHI clock is generated by dividing PCLKB by 2^n (n = 0 to 9)
	Error check functions	CRC7 (command/response), CRC16 (transfer data)
	Interrupt sources	Card access interrupt (SDHI_MMCn_ACCS), SDIO access interrupt (SDHI_MMCn_SDIO), Card detection interrupt (SDHI_MMCn_CARD) (n = 0, 1)
	DMA transfer sources	DMAC and DTC triggerable by the DMA transfer requests (SDHI_MMCn_ODMSDBREQ (n = 0, 1)) interrupt SD buffer is read and write accessible using the DMAC
	Other functions	• Card detect function • Write protect support
MMC	MMC bus interface	Transfer bus mode selectable from 1-bit, 4-bit, or 8-bit
	Transfer modes	Backward compatible mode or high-speed SDR mode selectable
	Other functions	e.MMC device access supported
Module-stop function	Module-stop state can be set for each channel to reduce power consumption.	
TrustZone Filter	Security and Privilege attribution can be set for each channel.	

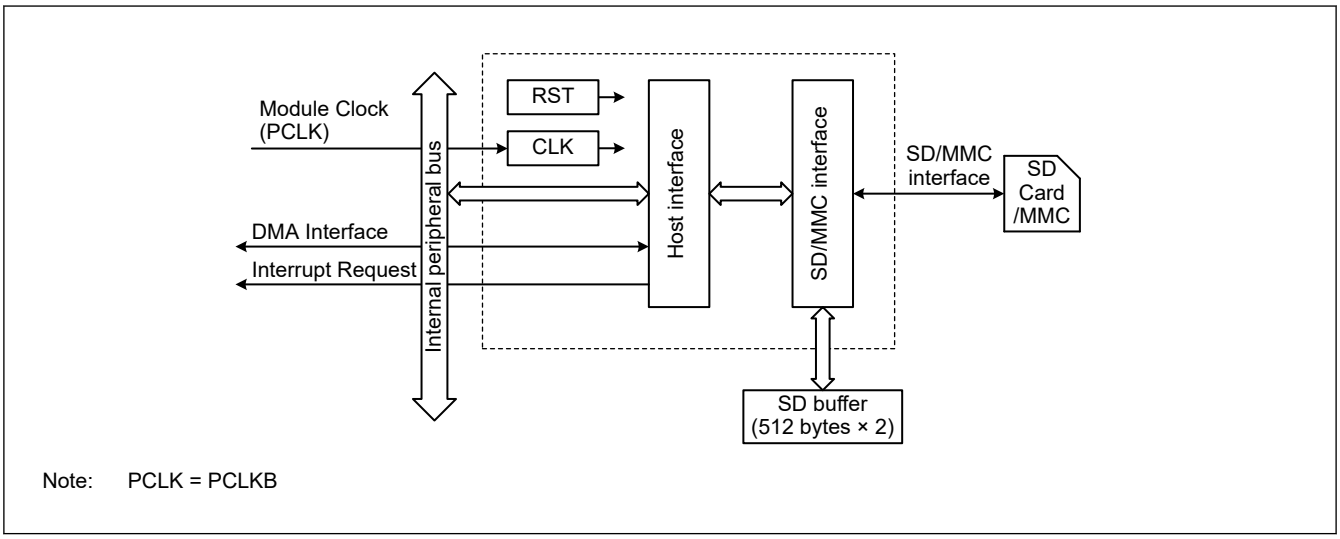


Figure 40.1 SD/MMC Host Interface block diagram

Table 40.2 SDHI I/O pins (n = 0, 1)

Channel	Pin name	I/O	Description
Ch n	SDnCLK	Output	SDHI clock
	SDnCMD	I/O	Command output, response input
	SDnDAT0	I/O	Data 0 (DAT0)
	SDnDAT1	I/O	Data 1 (DAT1), SDIO interrupt
	SDnDAT2	I/O	Data 2 (DAT2), SDIO Read wait
	SDnDAT3	I/O	Data 3 (DAT3), SD Card detect
	SDnDAT4	I/O	MMC Data 4 (DAT4)
	SDnDAT5	I/O	MMC Data 5 (DAT5)
	SDnDAT6	I/O	MMC Data 6 (DAT6)
	SDnDAT7	I/O	MMC Data 7 (DAT7)
	SDnCD	Input	SD card detection
	SDnWP	Input	SD card write protection

40.2 Register Descriptions

40.2.1 SD_CMD : Command Type Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMD12AT[1:0]		TRSTP	CMDRW	CMDTP	RSPTP[2:0]			ACMD[1:0]		CMDIDX[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	CMDIDX[5:0]	Command Index Field Value Select These bits configure the command index field value. The examples shown include the bit values for the ACMD[1:0] bits. 0x06: CMD6 0x12: CMD18 0x4D: ACMD13	R/W
7:6	ACMD[1:0]	Command Type Select 0 0: CMD 0 1: ACMD Others: Setting prohibited	R/W
10:8	RSPTP[2:0]	Response Type Select*1 0 0 0: Normal mode Depending on the command, the response type and transfer method are selected in the ACMD[1:0] and CMDIDX[5:0] bits. At this time, the values for bits 15 to 11 in this register are invalid. 0 1 1: Extended mode and no response 1 0 0: Extended mode and R1, R5, R6, or R7 response 1 0 1: Extended mode and R1b response 1 1 0: Extended mode and R2 response 1 1 1: Extended mode and R3 or R4 response Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
11	CMDTP	Data Transfer Select*2 0: Do not include data transfer (bc, bcr, or ac) in command 1: Include data transfer (adtc) in command	R/W
12	CMDRW	Data Transfer Direction Select*3 0: Write (SD/MMC Host Interface → SD card/MMC) 1: Read (SD/MMC Host Interface ← SD card/MMC)	R/W
13	TRSTP	Block Transfer Select*3 0: Single block transfer 1: Multiple blocks transfer	R/W
15:14	CMD12AT[1:0]	CMD12 Automatic Issue Select*4 0 0: Automatically issue CMD12 during multiblock transfer 0 1: Do not automatically issue CMD12 during multiblock transfer Others: Setting prohibited	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Some commands cannot be used in normal mode.
Note 2. The CMDTP bit is only valid when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b.
Note 3. Bits CMDRW and TRSTP are only valid when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b, and the CMDTP bit is 1.
Note 4. The CMD12AT[1:0] bits are only valid when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b, and the TRSTP bit is 1.

The command type and response type are set in the SD_CMD register. The command type and transfer mode must be set when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b. The sequence starts when a value is written to this register. See Table 40.8 and Table 40.9 for setting examples. Do not write to the SD_CMD register when the SD_INFO2.CBSY flag is 1.

40.2.2 SD_ARG : SD Command Argument Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	—	These bits specify command format[39:8] (argument).	R/W

Note: S-TYPE-3, P-TYPE-3

The SD_ARG register is used for setting the argument field value. Set the SD_ARG register before setting the SD_CMD register. The argument field value of the automatically issued CMD12 is 0x0000_0000 regardless of the SD_ARG register value.

40.2.3 SD_ARG1 : SD Command Argument Register 1

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x00C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits specify command format[39:24] (argument).	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The SD_ARG1 register is used for setting the argument field value. Set the SD_ARG1 register before setting the SD_CMD register. The argument field value of the automatically issued CMD12 is 0x0000_0000 regardless of the SD_ARG1 register value.

40.2.4 SD_STOP : Data Stop Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SEC	—	—	—	—	—	—	—	STP
Value after reset:	0	0	0	0	0	0	0	0*1	0	0	0	0	0	0	0	0*1

Bit	Symbol	Function	R/W
0	STP	Transfer Stop Data transfer stops when this bit is set to 1.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	SEC	Block Count Register Value Select*2 0: Disable SD_SECCNT register value 1: Enable SD_SECCNT register value	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The value is initialized by a reset and also on reset triggered by the SOFT_RST.SDRST flag.
Note 2. Do not rewrite this bit when the SD_INFO2.CBSY flag is 1.

The SD_STOP register stops data transfer. During a multiblock transfer sequence, the SD_SECCNT register value (number of blocks to be transferred) can be set to valid or invalid by setting the SD_STOP register.

STP bit (Transfer Stop)

When the STP bit is set to 1 during multiple block transfer, CMD12 is issued to halt the transfer through the SDHI. However, if a command sequence is halted because of a communications error or timeout, CMD12 is not issued. Although continued buffer access is possible even after STP is set to 1, the buffer access error bit (ILR or ILW) in SD_INFO2 is set accordingly.

When STP is set to 1 during transfer for single block write, the access end flag sets when SD_BUF becomes empty, and CMD12 is not issued. If SD_BUF does contain data, the access end flag sets on completion of reception of the busy state without CMD12 being issued.

When STP is set to 1 during transfer for single block read, the access end flag sets immediately after the STP bit is set, and CMD12 is not issued.

When STP is set to 1 during reception of the busy state after an R1b response, the access end flag sets on completion of reception of the busy state without CMD12 being issued.

When STP is set to 1 after a command sequence is completed, CMD12 is not issued and the access end flag does not set.

Set STP to 1 after the response end flag sets.

Set STP to 0 after the access end flag sets.

SEC bit (Block Count Register Value Select)

When SD_CMD is set in the following section to start the command sequence while the SEC bit is set to 1, CMD12 is automatically issued to stop multiblock transfer with the number of blocks set in SD_SECCNT.

CMD18 or CMD25 in normal mode (SD_CMD[10:8] = 000b)

SD_CMD[15:13] = 001b in extended mode (CMD12 is automatically issued, multiple block transfer)

When the command sequence is halted because of a communications error or timeout, CMD12 is not automatically issued.

40.2.5 SD_SECCNT : Block Count Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	—	These bits set the number of blocks to be transferred.	R/W

Note: S-TYPE-3, P-TYPE-3

When performing a multiblock transfer, SD_SECCNT is a read/write register used to set the number of blocks to be transferred. For example, when the register value is 0x0000_0001, 1 block is transferred. When the register value is 0x0000_FFFF, 65,535 blocks are transferred and when the register value is 0xFFFF_FFFF, 4,294,967,295 blocks are transferred. Do not set this register to 0x0000_0000. Do not rewrite the SD_SECCNT register when the SD_INFO2.CBSY flag is 1.

40.2.6 SD_RSPi : SD Card Response Register i (i = 10, 32, 54)

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x018 (SD_RSP10)
0x020 (SD_RSP32)
0x028 (SD_RSP54)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	—	These bits store the response from the SD card/MMC.	R

Note: S-TYPE-3, P-TYPE-3

40.2.7 SD_RSPj : SD Card Response Register j (j = 1, 3, 5)

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x01C (SD_RSP1)
0x024 (SD_RSP3)
0x02C (SD_RSP5)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits store the response from the SD card/MMC.	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

40.2.8 SD_RSP76 : SD Card Response Register 76

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x030

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	SD_RSP76[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SD_RSP76[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
23:0	SD_RSP76[23:0]	These bits store the response from the SD card/MMC.	R
31:24	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

40.2.9 SD_RSP7 : SD Card Response Register 7

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x034

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SD_RSP7[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SD_RSP7[7:0]	These bits store the response from the SD card/MMC.	R
31:8	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

SD_RSP10, SD_RSP32, SD_RSP54, SD_RSP1, SD_RSP3, SD_RSP5, SD_RSP76, and SD_RSP7 are read-only registers that store the response from the SD card/MMC. Depending on the type of response from the SD card/MMC, the SD/MMC Host Interface divides and stores the response among the four registers.

Table 40.3 lists the correspondence between the response type and its storage destination.

Table 40.3 Correspondence between response type and storage destination (1 of 2)

Response type	SD_RSP10 register	SD_RSP32 register	SD_RSP54 register	SD_RSP1 register	SD_RSP3 register	SD_RSP5 register	SD_RSP76 register	SD_RSP7 register
R1	[39:8]	—	[39:8]*1	—	—	—	—	—
R1b	[39:8]	—	[39:8]*1	—	—	—	—	—
R2	[39:8]	[71:40]	[103:72]	—	—	—	[127:104]	—
R3	[39:8]	—	—	—	—	—	—	—
R4	[39:8]	—	—	—	—	—	—	—

Table 40.3 Correspondence between response type and storage destination (2 of 2)

Response type	SD_RSP10 register	SD_RSP32 register	SD_RSP54 register	SD_RSP1 register	SD_RSP3 register	SD_RSP5 register	SD_RSP76 register	SD_RSP7 register
R5	[39:8]	—	—	—	—	—	—	—
R6	[39:8]	—	—	—	—	—	—	—
R7	[39:8]	—	—	—	—	—	—	—

Note 1. The responses for CMD18 and CMD25 are stored in registers SD_RSP10 and SD_RSP54. Therefore, even if the SD_RSP10 register is overwritten with the response for the automatically issued CMD12, the response for CMD18 or CMD25 can be confirmed by reading the SD_RSP54 register.

40.2.10 SD_INFO1 : SD Card Interrupt Flag Register 1

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x038

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SDD3MON	SDD3IN	SDD3RM	SDWPMON	—	SDCDMON	SDCDIN	SDCDRM	ACEND	—	RSPEND
Value after reset:	0	0	0	0	0	x	0	0	x	0	x	0	0	0*1	0	0*1

Bit	Symbol	Function	R/W
0	RSPEND	Response End Detection Flag 0: Response end not detected 1: Response end detected	R/(W)*2
1	—	This bit is read as 0. The write value should be 0.	R/W
2	ACEND	Access End Detection Flag 0: Access end not detected 1: Access end detected	R/(W)*2
3	SDCDRM	SDnCD Removal Flag 0: SD card/MMC removal not detected by the SDnCD pin 1: SD card/MMC removal detected by the SDnCD pin	R/(W)*2
4	SDCDIN	SDnCD Insertion Flag 0: SD card/MMC insertion not detected by the SDnCD pin 1: SD card/MMC insertion detected by the SDnCD pin	R/(W)*2
5	SDCDMON	SDnCD Pin Monitor Flag 0: SDnCD pin level is high*3 1: SDnCD pin level is low*3	R
6	—	This bit is read as 0. The write value should be 0.	R/W
7	SDWPMON	SDnWP Pin Monitor Flag 0: SDnWP pin level is high 1: SDnWP pin level is low	R
8	SDD3RM	SDnDAT3 Removal Flag 0: SD card/MMC removal not detected by the SDnDAT3 pin 1: SD card/MMC removal detected by the SDnDAT3 pin	R/(W)*2
9	SDD3IN	SDnDAT3 Insertion Flag 0: SD card/MMC insertion not detected by the SDnDAT3 pin 1: SD card/MMC insertion detected by the SDnDAT3 pin	R/(W)*2
10	SDD3MON	SDnDAT3 Pin Monitor Flag 0: SDnDAT3 pin level is low 1: SDnDAT3 pin level is high	R

Bit	Symbol	Function	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The value is initialized by a reset and also on reset triggered by the SOFT_RST.SDRST flag.
Note 2. The flag does not change even if set to 1. Writing 0 changes the flag value to 0.
Note 3. The flag changes when the pin level continues for the period set in the SD_OPTION.CTOP[3:0] bits or longer.

The SD_INFO1 register indicates the detection of a response end or access end for a command sequence. The SD_INFO1 register also indicates the detection SD card/MMC insertion/removal and the write protection status.

During a multiblock transfer sequence, if CMD12 or CMD52 (SDIO abort) is issued, the ACEND flag sets to 1, but the RSPEND flag remains set to 0.

If the command sequence is stopped because of a communication error or timeout, the ACEND flag or RSPEND flag sets to 1.

After a reset is canceled, the SDD3MON bit, SDD3IN flag, and SDD3RM flag values are changed in accordance with the status of the SDnDAT3 (n = 0, 1) pin, and their values are changed when data is being transferred in wide bus mode. These 3 bits are used only for SD card. Set flags to be cleared to 0. Set flags that are not being cleared to 1.

RSPEND flag (Response End Detection Flag)

The RSPEND flag indicates that a response end was detected.

[Setting conditions]

- When reception of the response is completed
- When transmission of a command without response is completed
- When reception of the busy state after R1b response is completed
- When reception of the response to CMD52 that was issued by setting the C52PUB bit to 1 is completed for transfer of multiple block read
- When reception of the response to CMD52 that was issued by setting the C52PUB bit to 1 is completed for transfer of multiple block write
- This bit is set when a command sequence is halted because of a communications error or timeout.

[Clearing conditions]

- When 0 is written to RSPEND
- When a command without data is issued.

Note: When a command is issued in absence of data transfer, the RSPEND flag becomes 1 after the command sequence ends.

ACEND flag (Access End Detection Flag)

The ACEND flag indicates that an access end was detected.

[Setting conditions]

- When read access to the buffer is completed for transfer of single block read
- When read access to the buffer for the last block of data is completed for transfer of multiple block read
- When read access to the buffer and reception of the response to CMD12 are completed for transfer of multiple block read with automatic issuing of CMD12
- When reception of the busy state after reception of the CRC status is completed for transfer of single block write
- When reception of the busy state after reception of the CRC status of the last block of data is completed for transfer of multiple block write
- When reception of the response busy state for CMD12 is completed for transfer of multiple block write with automatic issuing of CMD12
- When reception of the response to CMD12 that was issued by setting the STP bit to1 is completed for transfer of multiple block read

- When reception of the response busy state for CMD12 that was issued by setting the STP bit to1 is completed for transfer of multiple block write
- When reception of the response to CMD52 that was issued by setting the IOABT bit to 1 is completed for transfer of multiple block read
- When reception of the response to CMD52 that was issued by setting the IOABT bit to1 is completed for transfer of multiple block write
- This bit is set when a command sequence is halted because of a communications error or timeout.

[Clearing conditions]

- When 0 is written to ACEND
- When the access end bit is set to 1.

Note: The ACEND flag becomes 1 after the command sequence ends.

SDCDRM flag (SDnCD Removal Flag)

The SDCDRM flag indicates that SDnCD was removed.

[Setting condition]

- After a change in SDnCD from 0 to 1, Mcycle elapsed with SDnCD held at 1.

[Clearing conditions]

- When 0 is written to SDCDRM.

Note: Mcycle is set in bits [3:0] in SD_OPTION.

SDCDIN flag (SDnCD Insertion Flag)

The SDCDIN flag indicates that SDnCD was inserted.

[Setting condition]

- After a change in SDnCD from 1 to 0, Mcycle elapsed with SDnCD held at 0.

[Clearing conditions]

- When 0 is written to SDCDIN.

Note: Mcycle is set in bits [3:0] in SD_OPTION.

SDD3RM flag (SDnDAT3 Removal Flag)

The SDD3RM flag indicates that SDnDAT3 was removed.

[Setting condition]

- After a change in SDnDAT3 from 1 to 0, two cycles of PCLKB elapsed with SDnDAT3 held at 0.

[Clearing condition]

- When 0 is written to SDD3RM.

SDD3IN flag (SDnDAT3 Insertion Flag)

The SDD3IN flag indicates that SDnDAT3 was inserted.

[Setting condition]

- After a change in SDnDAT3 from 0 to 1, two cycles of PCLKB elapsed with SDnDAT3 held at 1.

[Clearing condition]

- When 0 is written to SDD3IN.

40.2.11 SD_INFO2 : SD Card Interrupt Flag Register 2

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x03C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ILA	CBSY	SD_CLK_CTRLEN	—	—	—	BWE	BRE	SDD0MON	RSPTO	ILR	ILW	DTO	ENDE	CRCE	CMDE
Value after reset:	0*1	0*1	1*1	0	x	0	0*1	0*1	x	0*1	0*1	0*1	0*1	0*1	0*1	0*1

Bit	Symbol	Function	R/W
0	CMDE	Command Error Detection Flag 0: Command error not detected 1: Command error detected	R/W*2
1	CRCE	CRC Error Detection Flag 0: CRC error not detected 1: CRC error detected	R/W*2
2	ENDE	End Bit Error Detection Flag 0: End bit error not detected 1: End bit error detected	R/W*2
3	DTO	Data Timeout Detection Flag 0: Data timeout not detected 1: Data timeout detected	R/W*2
4	ILW	SD_BUF0 Illegal Write Access Detection Flag 0: Illegal write access to the SD_BUF0 register not detected 1: Illegal write access to the SD_BUF0 register detected	R/W*2
5	ILR	SD_BUF0 Illegal Read Access Detection Flag 0: Illegal read access to the SD_BUF0 register not detected 1: Illegal read access to the SD_BUF0 register detected	R/W*2
6	RSPTO	Response Timeout Detection Flag 0: Response timeout not detected 1: Response timeout detected	R/W*2
7	SDD0MON	SDnDAT0 Pin Status Flag 0: SDnDAT0 pin is low 1: SDnDAT0 pin is high	R
8	BRE	SD_BUF0 Read Enable Flag 0: Disable read access to the SD_BUF0 register 1: Enable read access to the SD_BUF0 register	R/W*2
9	BWE	SD_BUF0 Write Enable Flag 0: Disable write access to the SD_BUF0 register 1: Enable write access to the SD_BUF0 register	R/W*2
10	—	This bit is read as 0. The write value should be 0.	R/W
11	—	The read value is undefined. The write value should be 1.	R/W
12	—	This bit is read as 0. The write value should be 0.	R/W
13	SD_CLK_CTRLEN	SD_CLK_CTRL Write Enable Flag 0: SD/MMC bus (CMD and DAT lines) is busy, so write access to the SD_CLK_CTRL.CLKEN and CLKSEL[7:0] bits is disabled 1: SD/MMC bus (CMD and DAT lines) is not busy, so write access to the SD_CLK_CTRL.CLKEN and CLKSEL[7:0] bits is enabled	R

Bit	Symbol	Function	R/W
14	CBSY	Command Sequence Status Flag 0: Command sequence complete 1: Command sequence in progress (busy)	R
15	ILA	Illegal Access Error Detection Flag 0: Illegal access error not detected 1: Illegal access error detected	R/W*2
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The value is initialized by a reset and also on reset triggered by the SOFT_RST.SDRST flag.
Note 2. The flag does not change even if set to 1. Writing 0 changes the flag value to 0.

The SD_INFO2 register indicates the status of the SD buffer and the status of the SD card/MMC. Set flags to be cleared to 0. Set flags that are not being cleared to 1.

CMDE flag (Command Error Detection Flag)

The CMDE flag indicates that a command error was detected. The command sequence stops when a command error occurs. When the SDIO_MODE.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence is not completed. Perform the error processing shown in section 40.3.12. IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read) or section 40.3.13. IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write), and complete the command sequence.

[Setting conditions]

- The command index of the transmitted command differs from the command index of the received response.
- The command index of a command issued within a command sequence differs from the command index of the received response.

[Clearing condition]

- When 0 is written to CMDE.

CRCE flag (CRC Error Detection Flag)

The CRCE flag indicates that a CRC error was detected. The command sequence stops when a CRC error occurs. When the SDIO_MODE.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence is not completed. Perform the error processing shown in section 40.3.12. IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read) or section 40.3.13. IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write), and complete the command sequence.

[Setting conditions]

- When an error occurs in the CRC status.
- When a CRC error occurs in the read data.
- When a CRC error occurs in the response.
- A CRC error in the response to a command issued within a command sequence.

[Clearing condition]

- When 0 is written to CRCE.

ENDE flag (End Bit Error Detection Flag)

The ENDE flag indicates that an end bit error was detected. The command sequence is stopped when an end bit error occurs. When the SDIO_MODE.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence is not completed. Perform the error processing shown in section 40.3.12. IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read) or section 40.3.13. IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write), and complete the command sequence.

[Setting conditions]

- When an error occurs in the response length (and the end bit is not detected).
- When an error occurs in the read data length (and the end bit is not detected among the valid bits).

- When an error occurs in the CRC status length (and the end bit is not detected).
- An error in the length of a response to a command issued within a command sequence, for example when the end bit is not detected.

[Clearing condition]

- When 0 is written to ENDE.

DTO flag (Data Timeout Detection Flag)

The DTO flag indicates that a data timeout was detected. The command sequence stops when a data timeout occurs.

[Setting conditions]

- After R1b response, the busy state (SDnDAT0 = 0) continues for longer than Ncycle.
- After CRC status, the busy state (SDnDAT0 = 0) continues for longer than Ncycle.
- After write data, the CRC status is not received though Ncycle has elapsed.
- After read command, read data is not received though a time longer than Ncycle has elapsed.
- After CMD12 is issued within a command sequence, the busy state (SDnDAT0 = 0) for longer than Ncycle continues.
- After the reception of read data, read data for the next block are not received though a time longer than Ncycle has elapsed.
- After release of the read wait state, read data for the next block are not received though a time longer than Ncycle has elapsed.

Note: Ncycle is set in bits [7:4] in SD_OPTION.

[Clearing condition]

- When 0 is written to DTO.

ILW flag (SD_BUF0 Illegal Write Access Detection Flag)

The ILW flag indicates that an SD_BUF0 illegal write access was detected.

[Setting conditions]

- When data is written to SD_BUF0 while it is not in the data read/write command state.
- When data is written to SD_BUF0 while SD_BUF is full.
- When data is written to SD_BUF0 while an error occurs in the CRC status or CRC status length.
- When data is written to SD_BUF0 while a busy state after the CRC status continues for longer than Ncycle.

Note: Ncycle is set in bits [7:4] in SD_OPTION.

[Clearing condition]

- When 0 is written to ILW.

ILR flag (SD_BUF0 Illegal Read Access Detection Flag)

The ILR flag indicates that an SD_BUF0 illegal read access was detected.

[Setting conditions]

- When SD_BUF is empty while SD_BUF0 is read.
- When data with a CRC error or END error is read from SD_BUF0.

[Clearing condition]

- When 0 is written to ILR.

RSPTO flag (Response Timeout Detection Flag)

The RSPTO flag indicates that a response timeout was detected. The command sequence is stopped when a response timeout occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error

or response timeout occurs, the command sequence is not completed. Perform the error processing shown in section 40.3.12. IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read) or section 40.3.13. IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write), and complete the command sequence.

[Setting condition]

- When a response is not received though a time longer than 640 cycles of SD/MMC clock has elapsed (including a response to a command issued within a command sequence).

[Clearing condition]

- When 0 is written to RSPTO.

SDD0MON flag (SDnDAT0 Pin Status Flag)

The SDD0MON flag indicates the status of the SDnDAT0 pin. If the data timeout (DTO) is set but the response timeout (RSPTO) is not set after the Erase command is issued, the end of the Erase sequence (SDD0MON = 1) is confirmed by polling DAT0.

If a communication error or timeout occurs during a write sequence, the DAT0 bit might retain the value 0.

While the SD/MMC clock is stopped, the DAT0 bit retains the value before the clock is stopped.

BRE flag (SD_BUF0 Read Enable Flag)

The BRE flag indicates that SD_BUF0 is enabled for reading.

[Setting conditions]

- When data set in SD_SIZE is stored in SD_BUF0 at single block transfer.
- When data set in SD_SIZE is stored in either bank 1 or bank 2 of SD_BUF0 at multiple block transfer.

[Clearing conditions]

- When 0 is written to BRE
- Reading of a block of data from SD_BUF0 by DMA transfer

When data is read from SD_BUF0 by the CPU, clear BRE then read the amount of data specified in SD_SIZE.

Even if a CRC error or an END error occurs while block data is read, data is stored in SD_BUF0 and BRE is set.

BWE flag (SD_BUF0 Write Enable Flag)

The BWE flag indicates that SD_BUF0 is enabled for writing.

[Setting conditions]

- When SD_BUF0 is empty at single block transfer.
- When either bank 1 or bank 2 of SD_BUF0 is empty at multiple block transfer.

[Clearing conditions]

- When 0 is written to BWE.
- Writing of a block of data to SD_BUF0 by DMA transfer.

When data is written to SD_BUF0 by the CPU, clear BWE and then write the amount of data specified in SD_SIZE.

SD_CLK_CTRLEN flag (SD_CLK_CTRL Write Enable Flag)

When a command sequence is started by writing to SD_CMD, the CBSY bit is set to 1 and, at the same time, the SD_CLK_CTRLEN bit is set to 0. The SD_CLK_CTRLEN bit is set to 1 after 8 cycles of SDCLK have elapsed after the CBSY bit clears to 0 on completion of the command sequence.

ILA flag (Illegal Access Error Detection Flag)

The ILA flag indicates that an illegal access error was detected.

[Setting conditions]

- Writing of data to SD_CMD within a command sequence (CBSY = 1).

- When SD_CMD[11] = 1 (command with data transfer) and SD_CMD[7:0] = 0000 1100b (CMD12) are set in SD_CMD.

[Clearing condition]

- When 0 is written to ILA.

40.2.12 SD_INFO1_MASK : SD INFO1 Interrupt Mask Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x040

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SDD3INM	SDD3RMM	—	—	—	SDCDINM	SDCDRMM	ACENDM	—	RSPENDM
Value after reset:	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	1

Bit	Symbol	Function	R/W
0	RSPENDM	Response End Interrupt Request Mask 0: Do not mask response end interrupt request 1: Mask response end interrupt request	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	ACENDM	Access End Interrupt Request Mask 0: Do not mask access end interrupt request 1: Mask access end interrupt request	R/W
3	SDCDRMM	SDnCD Removal Interrupt Request Mask 0: Do not mask SD card/MMC removal interrupt request by the SDnCD pin 1: Mask SD card/MMC removal interrupt request by the SDnCD pin	R/W
4	SDCDINM	SDnCD Insertion Interrupt Request Mask 0: Do not mask SD card/MMC insertion interrupt request by the SDnCD pin 1: Mask SD card/MMC insertion interrupt request by the SDnCD pin	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	SDD3RMM	SDnDAT3 Removal Interrupt Request Mask 0: Do not mask SD card/MMC removal interrupt request by the SDnDAT3 pin 1: Mask SD card/MMC removal interrupt request by the SDnDAT3 pin	R/W
9	SDD3INM	SDnDAT3 Insertion Interrupt Request Mask 0: Do not mask SD card/MMC insertion interrupt request by the SDnDAT3 pin 1: Mask SD card/MMC insertion interrupt request by the SDnDAT3 pin	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The SD_INFO1_MASK register enables or disables interrupt requests from the status flags in the SD_INFO1 register. See Table 40.5, for details on the relationship between the status flags and the requested interrupt source.

40.2.13 SD_INFO2_MASK : SD INFO2 Interrupt Mask Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x044

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ILAM	—	—	—	—	—	BWEM	BREM	—	RSPTOM	ILRM	ILWM	DTOM	ENDEM	CRCEM	CMDEM
Value after reset:	1	0	0	0	1	0	1	1	0	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	CMDEM	Command Error Interrupt Request Mask 0: Do not mask command error interrupt request 1: Mask command error interrupt request	R/W
1	CRCEM	CRC Error Interrupt Request Mask 0: Do not mask CRC error interrupt request 1: Mask CRC error interrupt request	R/W
2	ENDEM	End Bit Error Interrupt Request Mask 0: Do not mask end bit detection error interrupt request 1: Mask end bit detection error interrupt request	R/W
3	DTOM	Data Timeout Interrupt Request Mask 0: Do not mask data timeout interrupt request 1: Mask data timeout interrupt request	R/W
4	ILWM	SD_BUF0 Register Illegal Write Interrupt Request Mask 0: Do not mask illegal write detection interrupt request for the SD_BUF0 register 1: Mask illegal write detection interrupt request for the SD_BUF0 register	R/W
5	ILRM	SD_BUF0 Register Illegal Read Interrupt Request Mask 0: Do not mask illegal read detection interrupt request for the SD_BUF0 register 1: Mask illegal read detection interrupt request for the SD_BUF0 register	R/W
6	RSPTOM	Response Timeout Interrupt Request Mask 0: Do not mask response timeout interrupt request 1: Mask response timeout interrupt request	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	BREM*1	BRE Interrupt Request Mask 0: Do not mask read enable interrupt request for the SD buffer 1: Mask read enable interrupt request for the SD buffer	R/W
9	BWEM*1	BWE Interrupt Request Mask 0: Do not mask write enable interrupt request for the SD_BUF0 register 1: Mask write enable interrupt request for the SD_BUF0 register	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	—	This bit is read as 1.The write value should be 1.	R/W
14:12	—	These bits are read as 0. The write value should be 0.	R/W
15	ILAM	Illegal Access Error Interrupt Request Mask 0: Do not mask illegal access error interrupt request 1: Mask illegal access error interrupt request	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. When the SD_INFO2_MASK.BWEM bit is 0 or the SD_INFO2_MASK.BREM bit is 0, set the SD_DMAEN.DMAEN bit to 0. When the SD_DMAEN.DMAEN bit is 1, set the SD_INFO2_MASK.BWEM bit to 1 and the SD_INFO2_MASK.BREM bit to 1.

The SD_INFO2_MASK register enables or disables interrupt requests from the status flags in the SD_INFO2 register. See Table 40.5 for details on the relationship between the status flags and the requested interrupt source.

40.2.14 SD_CLK_CTRL : SD Clock Control Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x048

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CLKCTRLEN	CLKEN	CLKSEL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0*1	0	0	1	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CLKSEL[7:0]	SDHI Clock Frequency Select*2 0xFF PCLKB 0x00: PCLKB/2 0x01: PCLKB/4 0x02: PCLKB/8 0x04: PCLKB/16 0x08: PCLKB/32 0x10: PCLKB/64 0x20: PCLKB/128 0x40: PCLKB/256 0x80: PCLKB/512 Others: Setting prohibited	R/W
8	CLKEN	SD/MMC Clock Output Control*2 0: Disable SD/MMC clock output (fix SDnCLK signal low) 1: Enable SD/MMC clock output	R/W
9	CLKCTRLEN	SD/MMC Clock Output Automatic Control Select 0: Disable automatic control of SD/MMC clock output 1: Enable automatic control of SD/MMC clock output	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The value is initialized by a reset and also on reset triggered by the SOFT_RST.SDRST flag.
Note 2. Bits CLKSEL[7:0] and CLKEN cannot be write accessed when the SD_INFO2.SD_CLK_CTRLEN flag is 0.

The SD_CLK_CTRL register controls the SD/MMC clock frequency settings and output. Set the CLKEN bit to 1 before writing to the SD_CMD register to start a command sequence. Do not write to the SD_CLK_CTRL register when the SD_INFO2.SD_CLK_CTRLEN flag is 0.

CLKCTRLEN bit (SD/MMC Clock Output Automatic Control Select)

The CLKCTRLEN bit enables or disables the automatic control function for SD/MMC clock output, which causes the SD/MMC clock to output only within a command sequence.

The timing with which SD/MMC clock output starts and stops is as follows:

- SD/MMC clock output starts after writing to SD_CMD
- SD/MMC clock output stops when 8 cycles of SD/MMC clock have elapsed after the end of the command sequence.

In addition, SD/MMC clock is fixed to 0 while SCLKEN of SD_CLK_CTRL is 0, regardless of the value of this bit.

40.2.15 SD_SIZE : Transfer Data Length Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x04C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	LEN[9:0]									
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	LEN[9:0]	Transfer Data Size Setting These bits specify the transfer data size.*1	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Do not rewrite these bits when the SD_INFO2.CBSY flag is 1.

The SD_SIZE register sets the transfer data size.

LEN[9:0] bits (Transfer Data Size Setting)

When using single block transfer, the transfer data size can be set in the LEN[9:0] bits from 1 byte to 512 bytes. When CMD12 is automatically issued during a multiblock transfer sequence (CMD18 and CMD25), the transfer data size can only be set to 512 bytes. When CMD12 is not automatically issued during a multiblock transfer sequence, the transfer data size can be set to 32, 64, 128, 256, or 512 bytes. However, a 32, 64, 128, or 256 bytes multiblock read transfer can only be performed during an SDIO multiblock transfer (CMD53). Do not set these bits to 0 when using a command that includes data transfer.

40.2.16 SD_OPTION : SD Card Access Control Option Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x050

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WIDTH	—	WIDTH8	—	—	—	—	TOUTMASK	TOP[3:0]				CTOP[3:0]			
Value after reset:	0*1	1	0*1	0	0	0	0	0*1	1*1	1*1	1*1	0*1	1*1	1*1	1*1	0*1

Bit	Symbol	Function	R/W
3:0	CTOP[3:0]	Card Detection Time Counter*2 0x0: PCLKB × 2^{10} 0x1: PCLKB × 2^{11} 0x2: PCLKB × 2^{12} 0x3: PCLKB × 2^{13} 0x4: PCLKB × 2^{14} 0x5: PCLKB × 2^{15} 0x6: PCLKB × 2^{16} 0x7: PCLKB × 2^{17} 0x8: PCLKB × 2^{18} 0x9: PCLKB × 2^{19} 0xA: PCLKB × 2^{20} 0xB: PCLKB × 2^{21} 0xC: PCLKB × 2^{22} 0xD: PCLKB × 2^{23} 0xE: PCLKB × 2^{24} 0xF: Setting prohibited	R/W
7:4	TOP[3:0]	Timeout Counter*2 0x0: SDHI clock × 2^{13} 0x1: SDHI clock × 2^{14} 0x2: SDHI clock × 2^{15} 0x3: SDHI clock × 2^{16} 0x4: SDHI clock × 2^{17} 0x5: SDHI clock × 2^{18} 0x6: SDHI clock × 2^{19} 0x7: SDHI clock × 2^{20} 0x8: SDHI clock × 2^{21} 0x9: SDHI clock × 2^{22} 0xA: SDHI clock × 2^{23} 0xB: SDHI clock × 2^{24} 0xC: SDHI clock × 2^{25} 0xD: SDHI clock × 2^{26} 0xE: SDHI clock × 2^{27} 0xF: Setting prohibited	R/W
8	TOUTMASK	Timeout Mask 0: Activate timeout 1: Inactivate timeout (do not set RSPTO and DTO bits of SD_INFO2 or CRCBSYTO, CRCTO, RDTO, BSYTO1, BSYTO0, RSPTO1 and RSPTO0 bits of SD_ERR_STS2) When timeout occurs because of an inactivated timeout, execute a software reset to terminate the command sequence.	R/W
12:9	—	These bits are read as 0. The write value should be 0.	R/W
13	WIDTH8	Bus Width*2 See bit 15 WIDTH bit.	R/W
14	—	This bit is read as 1. The write value should be 1.	R/W
15	WIDTH	Bus Width*2 **WIDTH / WIDTH8 / Bus Width** 0 / 1 / 8-bit width 0 / 0 / 4-bit width 1 / 0 / 1-bit width 1 / 1 / 1-bit width For 1-byte write transfers, set 4-bit or 1-bit width. Do not set 8-bit width.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The value is initialized by a reset and also on reset triggered by the SOFT_RST.SDRST flag.

Note 2. Do not rewrite these bits when the SD_INFO2.CBSY flag is 1.

The SD bus width and timeout counter are set in the SD_OPTION register.

40.2.17 SD_ERR_STS1 : SD Error Status Register 1

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x058

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CRCTK[2:0]			CRCTKE	RDCRCE	RSPCRCE1	RSPCRCE0	—	—	CRCLENE	RDLENE	RSPLENE1	RSPLENE0	CMDE1	CMDE0
Value after reset:	0	0*1	1*1	0*1	0*1	0*1	0*1	0*1	0	0	0*1	0*1	0*1	0*1	0*1	0*1

Bit	Symbol	Function	R/W
0	CMDE0	Command Error Flag 0 0: No error exists in command index field value of a command*2 response 1: Error exists in command index field value of a command*2 response	R
1	CMDE1	Command Error Flag 1 0: No error exists in command index field value of a command*3 response 1: Error exists in command index field value of a command*3 response (with SD_CMD.CMDIDX[5:0] setting, an error that occurs with CMD12 issue is indicated in the CMDE0 flag)	R
2	RSPLENE0	Response Length Error Flag 0 0: No error exists in command*2 response length 1: Error exists in command*2 response length	R
3	RSPLENE1	Response Length Error Flag 1 0: No error exists in command*3 response length 1: Error exists in command*3 response length (with SD_CMD.CMDIDX[5:0] setting, an error that occurs with CMD12 issue is indicated in the RSPLENE0 flag)	R
4	RDLENE	Read Data Length Error Flag 0: No read data length error occurred 1: Read data length error occurred	R
5	CRCLENE	CRC Status Token Length Error Flag 0: No CRC status token length error occurred 1: CRC status token length error occurred	R
7:6	—	These bits are read as 0.	R
8	RSPCRCE0	Response CRC Error Flag 0 0: No CRC error detected in command*2 response 1: CRC error detected in command*2 response	R
9	RSPCRCE1	Response CRC Error Flag 1 0: No CRC error detected in command*3 response (with SD_CMD.CMDIDX[5:0] setting, an error that occurs with CMD12 issue is indicated in the RSPCRCE0 flag) 1: CRC error detected in command*3 response	R
10	RDCRCE	Read Data CRC Error Flag 0: No CRC error detected in read data 1: CRC error detected in read data	R
11	CRCTKE	CRC Status Token Error Flag 0: No error detected in CRC status token 1: Error detected in CRC status token	R

Bit	Symbol	Function	R/W
14:12	CRCTK[2:0]	CRC Status Token These bits store the CRC status token value (normal value is 010b).	R
31:15	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3
Note 1. The value is initialized by a reset and also on reset triggered by the SOFT_RST.SDRST flag.
Note 2. CMD other than CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.
Note 3. CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

The SD_ERR_STS1 register indicates the CRC status token, CRC error, end bit error, and command error.

40.2.18 SD_ERR_STS2 : SD Error Status Register 2

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x05C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	CRCBSYTO	CRCTO	RDTO	BSYTO1	BSYTO0	RSPTO1	RSPTO0
Value after reset:	0	0	0	0	0	0	0	0	0	0*1	0*1	0*1	0*1	0*1	0*1	0*1

Bit	Symbol	Function	R/W
0	RSPTO0	Response Timeout Flag 0 0: After command*2 was issued, response was received in less than 640 cycles of the SD/MMC clock 1: After command*2 was issued, response was not received in 640 or more cycles of the SD/MMC clock	R
1	RSPTO1	Response Timeout Flag 1 0: After command*3 was issued, response was received in less than 640 cycles of the SD/MMC clock 1: After command*3 was issued, response was not received after 640 or more cycles of the SD/MMC clock (with SD_CMD.CMDIDX[5:0] setting, an error that occurs with CMD12 issue is indicated in the RSPTO0 flag)	R
2	BSYTO0	Busy Timeout Flag 0 0: After R1b response was received, SD/MMC was released from the busy state during the specified period*4 1: After R1b response was received, SD/MMC was in the busy state after the specified period*4 elapsed	R
3	BSYTO1	Busy Timeout Flag 1 0: After CMD12 was automatically issued, SD/MMC was released from the busy state during the specified period*4 1: After CMD12 was automatically issued, SD/MMC was in the busy state after the specified period*4 elapsed (with SD_CMD.CMDIDX[5:0] setting, an error that occurs with CMD12 issue is indicated in the BSYTO0 flag)	R
4	RDTO	Read Data Timeout Flag When a read command is issued, this flag sets to 1 when read data is not received after the specified period*4 elapses. When read data is received, this flag sets to 1 when the next block of read data is not received after the specified period*4 elapses. When the SD/MMC exits the read wait state, this flag sets to 1 when the next block of read data is not received after the specified period*4 elapses.	R

Bit	Symbol	Function	R/W
5	CRCTO	CRC Status Token Timeout Flag 0: After CRC data was written to the SD card/MMC, a CRC status token was received during the specified period*4 1: After CRC data was written to the SD card/MMC, a CRC status token was not received after the specified period*4 elapsed	R
6	CRCBSYTO	CRC Status Token Busy Timeout Flag 0: After a CRC status token was received, the SD/MMC was released from the busy state during the specified period*4 1: After a CRC status token was received, the SD/MMC was in the busy state after the specified period*4 elapsed	R
31:7	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3
Note 1. The value is initialized by a reset and also on reset triggered by the SOFT_RST.SDRST flag.
Note 2. CMD other than CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.
Note 3. CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.
Note 4. Set the SD_OPTION.TOP[3:0] bits to select the number of *n* cycles.

The SD_ERR_STS2 register indicates the timeout status.

40.2.19 SD_BUF0 : SD Buffer Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x060

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:																
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	n/a	When writing to the SD card, the write data is written to this register. When reading from the SD card, the read data is read from this register.	R/W

This register is internally connected to two 512-byte buffers.

If both buffers are not empty when executing multiple block read, the SD card/MMC clock is stopped to suspend receiving data. When one of the buffers is empty, the SD card/MMC clock is supplied to resume receiving data.

40.2.20 SDIO_MODE : SDIO Mode Control Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x068

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	C52PUB	IOABT	—	—	—	—	—	RWREQ	—	INTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	INTEN	SDIO Interrupt Acceptance Enable*1 0: Disable SDIO interrupt acceptance 1: Enable SDIO interrupt acceptance	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	RWREQ	Read Wait Request 0: Allow SD/MMC to exit read wait state 1: Request for SD/MMC to enter read wait state	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	IOABT	SDIO Abort If this bit is set to 1 during multiblock transfer triggered by CMD53, CMD52 is immediately issued, and the command sequence is aborted.	R/W
9	C52PUB	SDIO None Abort If this bit is set to 1 during multiblock transfer triggered by CMD53, CMD52 is issued after the transfer process is complete, and the command sequence is completed.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Do not rewrite this bit when the SD_INFO2.CBSY flag is 1.

The SDIO_MODE register controls reception of the SDIO interrupt, CMD52 issuance during multiblock transfer, and read wait request. Do not set bits C52PUB and IOABT to 1 at the same time.

RWREQ bit (Read Wait Request)

When RWREQ is set to 1 in the CMD53 (multiple block) read sequence, the block transfer enters the read wait state between blocks.

[Read wait state releasing]

- The read wait state is released, when RWREQ is cleared to 0 in the read wait state.
- When IOABT is set to 1 in the read wait state, RWREQ is automatically cleared to 0 after CMD52 is issued, and then the read wait state is released.
- When C52PUB and RWREQ are set to 1 simultaneously in the CMD53 (multiple block) read sequence, the read wait state is not automatically released. Therefore, after the CMD52 response is received, clear RWREQ. You must set RWREQ and C52PUB simultaneously.

When RWREQ is set to 1 while the last block in the CMD53 (multiple block) read sequence is transferred, the read wait state is not entered and RWREQ is automatically cleared to 0 by setting access end. Set RWREQ to 1 after the response end flag sets.

IOABT bit (SDIO Abort)

When the IOABT bit is set to 1 in a CMD53 (multiple block) sequence, the CMD53 sequence is halted and CMD52 is issued. However, if a command sequence is halted because of a communication error or timeout, CMD52 is not issued.

Although continued buffer access is possible even after IOABT is set to 1, the buffer access error bit (ILR or ILW) in SD_INFO2 is set accordingly. Set SD_ARG before setting IOABT to 1.

When IOABT is set to 1 during transfer for a single block write, the access end flag sets when SD_BUF0 becomes empty, and CMD52 is not issued. If SD_BUF0 contains data, the access end flag sets on completion of reception of the busy state without CMD52 being issued.

When IOABT is set to 1 during transfer for single block read, the access end flag sets immediately after IOABT is set, and CMD52 is not issued.

When IOABT is set to 1 during reception of the busy state after an R1b response, the access end flag sets on completion of reception of the busy state without CMD52 being issued.

When IOABT is set to 1 after a command sequence is completed, CMD52 is not issued and the access end flag does not set.

Set IOABT to 1 after the response end flag sets.

Set IOABT to 0 after the access end flag sets.

C52PUB bit (SDIO None Abort)

When the C52PUB bit is set to 1 in the CMD53 (multiple block) write sequence, CMD52 is automatically issued between blocks if SD_BUF0 becomes empty. C52PUB is automatically cleared to 0 after reception of the response to CMD52 is completed. Additionally, if C52PUB is set to 1 while the last block is being transferred, CMD52 is not issued. In this case, C52PUB is automatically cleared to 0 after the access end flag sets to 1.

When C52PUB and RWREQ are set to 1 in the CMD53 (multiple block) read sequence, the block transfer enters the read wait state between blocks and CMD52 is automatically issued. C52PUB is automatically cleared to 0 after reception of the response to CMD52 is completed. Additionally, if C52PUB is set to 1 while the last block is being transferred, CMD52 is not issued. In this case, C52PUB is automatically cleared to 0 after the access end flag sets to 1.

If C52PUB is set to 1 in the CMD53 (multiple block) read sequence, you must set RWREQ to 1 in addition to C52PUB.

Set SD_ARG before setting C52PUB to 1.

Set C52PUB to 1 after the response end flag sets.

40.2.21 SDIO_INFO1 : SDIO Interrupt Flag Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x06C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EXWT	EXPUB52	—	—	—	—	—	—	—	—	—	—	—	—	—	IOIRQ
Value after reset:	0*1	0*1	0	0	0	0	0	0	0	0	0	0	0	x	x	0*1

Bit	Symbol	Function	R/W
0	IOIRQ	SDIO Interrupt Status Flag 0: No SDIO interrupt detected 1: SDIO interrupt detected	R/(W)*2
2:1	—	The read values are undefined. The write value should be 1.	R/W
13:3	—	These bits are read as 0. The write value should be 0.	R/W
14	EXPUB52	EXPUB52 Status Flag Indicates the status of the EXPUB52.	R/(W)*2
15	EXWT	EXWT Status Flag Indicates the status of the EXWT.	R/(W)*2

Bit	Symbol	Function	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The value is initialized by a reset and also on reset triggered by the SOFT_RST.SDRST flag.
Note 2. Only 0 can be written to clear the bit.

The SDIO_INFO1 register indicates the status of the SDIO card access. Set flags to be cleared to 0. Set flags that are not being cleared to 1.

IOIRQ flag (SDIO Interrupt Status Flag)

The IOIRQ flag indicates that an SDIO interrupt occurred.

[Setting condition]

- When SDIO interrupt from an SDIO card is received while INTEN in SDIO_MODE is set to 1.

[Clearing condition]

- When 0 is written to IOIRQ.*1

Note 1. Before clearing this bit, access the SDIO card to negate the SDIO interrupt signal from the SDIO card. If the interrupt signal is not negated, this bit can be set again.

EXPUB52 flag (EXPUB52 Status Flag)

The EXPUB52 flag indicates the EXPUB52 status.

[Setting conditions]

- While the last block in the CMD53 (multiple block) sequence is transferred, C52PUB in SDIO_MODE is set to 1.
- While C52PUB is set to 1 in the CMD53 (multiple block) write sequence, the last block is transferred.

[Clearing condition]

- When 0 is written to EXPUB52.

EXWT flag (EXWT Status Flag)

The EXWT flag indicates the EXWT status.

[Setting condition]

- While the last block in the CMD53 (multiple block) read sequence is transferred, RWREQ in SDIO_MODE is set to 1.

[Clearing condition]

- When 0 is written to EXWT.

40.2.22 SDIO_INFO1_MASK : SDIO INFO1 Interrupt Mask Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x070

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EXWTM	EXPUB52M	—	—	—	—	—	—	—	—	—	—	—	—	—	IOIRQM
Value after reset:	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Function	R/W
0	IOIRQM	IOIRQ Interrupt Mask Control 0: Do not mask IOIRQ interrupts 1: Mask IOIRQ interrupts	R/W
2:1	—	These bits are read as 1.The write value should be 1.	R/W
13:3	—	These bits are read as 0. The write value should be 0.	R/W
14	EXPUB52M	EXPUB52 Interrupt Request Mask Control 0: Do not mask EXPUB52 interrupt requests 1: Mask EXPUB52 interrupt requests	R/W
15	EXWTM	EXWT Interrupt Request Mask Control 0: Do not mask EXWT interrupt requests 1: Mask EXWT interrupt requests	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The SDIO_INFO1_MASK register enables or disables interrupt requests from the status flags in the SDIO_INFO1 register. See Table 40.5 for details on the relationship between the status flags and the requested interrupt source.

40.2.23 SD_DMAEN : DMA Mode Enable Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x1B0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMAEN	—
Value after reset:	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	DMAEN	DMA Transfer Enable*1 *2 0: Disable use of DMA transfer to access SD_BUF0 register 1: Enable use of DMA transfer to access SD_BUF0 register	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	—	This bit is read as 1. The write value should be 1.	R/W
11:5	—	These bits are read as 0. The write value should be 0.	R/W
12	—	This bit is read as 1. The write value should be 1.	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Do not rewrite this bit when the SD_INFO2.CBSY bit is 1.
Note 2. When the SD_INFO2_MASK.BWEM bit is 0 or the SD_INFO2_MASK.BREM bit is 0, set the SD_DMAEN.DMAEN bit to 0. When the SD_DMAEN.DMAEN bit is 1, set the SD_INFO2_MASK.BWEM bit to 1 and the SD_INFO2_MASK.BREM bit to 1.

The SD_DMAEN register enables or disables DMA transfers.

DMAEN bit (DMA Transfer Enable)

When using DMA transfer to access the SD buffer, set the DMAEN bit to 1 before setting the SD_CMD register.

40.2.24 SOFT_RST : Software Reset Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x1C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Function	R/W
0	SDRST	Software Reset Control 0: Reset SD/MMC Host Interface software 1: Cancel reset of SD/MMC Host Interface software	R/W
2:1	—	These bits are read as 1. The write value should be 1.	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Table 40.4 lists the bits and flags initialized by SD/MMC Host Interface software reset.

Table 40.4 Bits and flags initialized by SD/MMC Host Interface software reset

Register	Bit/flag
SD_STOP	SEC, STP
SD_INFO1	RSPEND, ACEND
SD_INFO2	CMDE, CRCE, ENDE, DTO, ILW, ILR, RSPTO, BRE, BWE, SD_CLK_CTRLEN, CBSY, ILA
SD_CLK_CTRL	CLKEN
SD_OPTION	CTOP[3:0], TOP[3:0], TOUTMASK, WIDTH8, WIDTH
SD_ERR_STS1	CMDE0, CMDE1, RSPLENE0, RSPLENE1, RDLENE, CRCLENE, RSPCRCE0, RSPCRCE1, RDCRCE, CRCTKE, CRCTK[2:0]
SD_ERR_STS2	RSPTO0, RSPTO1, BSYTO0, BSYTO1, RDTO, CRCTO, CRCBSYTO
SDIO_INFO1	IOIRQ, EXPUB52, EXWT
SDIF_MODE	NOCHKCR

40.2.25 SDIF_MODE : SD Interface Mode Setting Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x1CC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	NOCHKCR	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0*1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	NOCHKCR	CRC Check Mask CRC check mask bit for MMC test commands. Set when CRC16 or CRC status value check is not executed. 0: Enable CRC check 1: Disable CRC Check (ignore CRC16 valued when reading and ignore CRC status value when writing)	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The value is initialized by a reset and also on reset triggered by the SOFT_RST.SDRST flag.

NOCHKCR bit (CRC Check Mask)

The NOCHKCR bit is used for MMC test commands. This bit is set when CRC16 or CRC status value check is not executed.

40.2.26 EXT_SWAP : Swap Control Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x1E0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	BRSWP	BWSWP	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	BWSWP	SD_BUF0 Swap Write*1 0: Normal write operation 1: Swap the byte endian order before writing to SD_BUF0 register	R/W
7	BRSWP	SD_BUF0 Swap Read*1 0: Normal read operation 1: Swap the byte endian order before reading SD_BUF0 register	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Do not rewrite this bit when the SD_INFO2.CBSY flag is 1.

The EXT_SWAP register selects whether or not the byte endian order is swapped when accessing the SD_BUF0 register. See section 40.3.1. SD/MMC Interface for details on the differences in accessing the SD_BUF0 register based on the EXT_SWAP register value.

40.3 Operation

40.3.1 SD/MMC Interface

When data is read from the SD card/MMC, the process is as follows:

1. The SD/MMC Host Interface receives data from the SD card/MMC through the SDnDAT signal (see Figure 40.2 and Figure 40.3).
2. The received data is stored in SD_BUF of the MMC Host Interface (see Figure 40.4).

3. The data stored in SD_BUF is read from SD_BUF0 (see Figure 40.5).

When data is written to the SD card/MMC, the specified procedure is reversed.

When accessing SD_BUF0, pay attention to the transfer order in SDnDAT and the store order in SD_BUF. If required, you can change the byte endian of the data read from or written to SD_BUF0 using the SDSWAP register. See Figure 40.6.

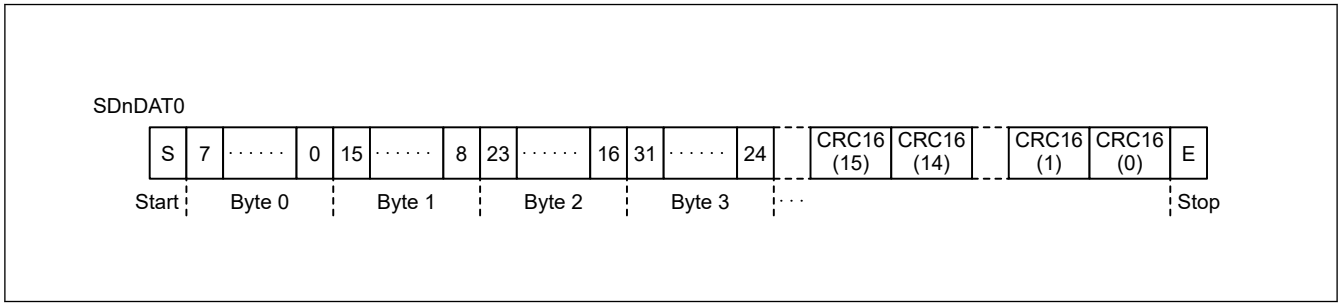


Figure 40.2 SDnDAT in 1-bit width mode

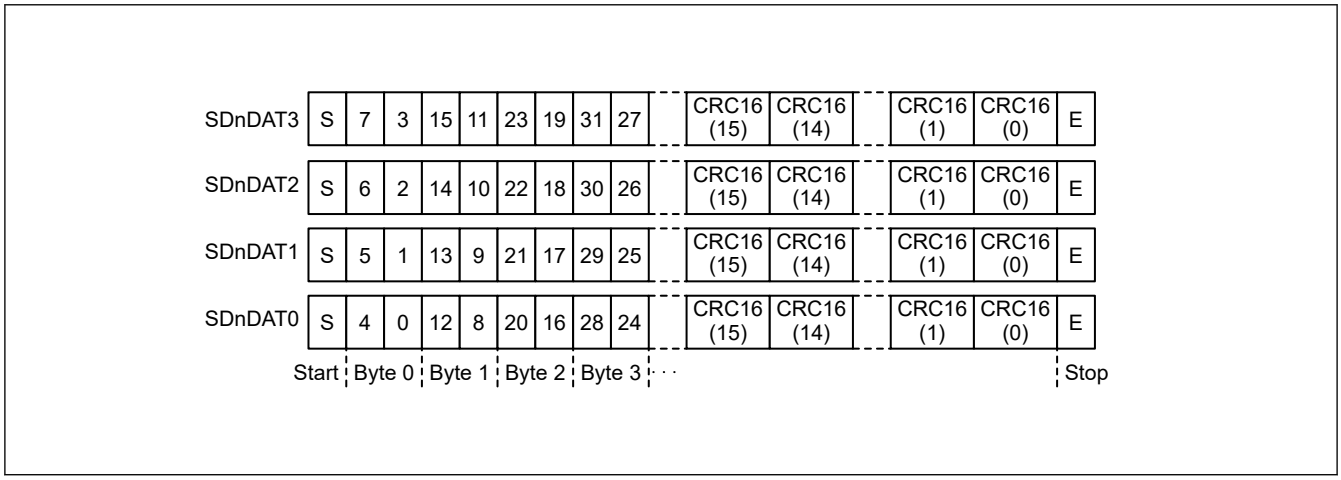


Figure 40.3 SDnDAT in 4-bit width mode

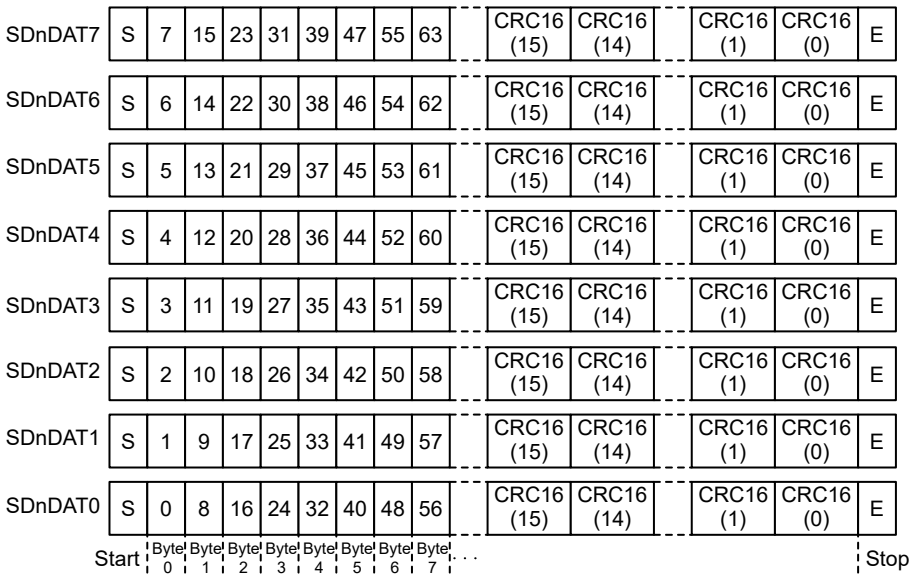


Figure 40.4 SDnDAT in 8-bit width mode

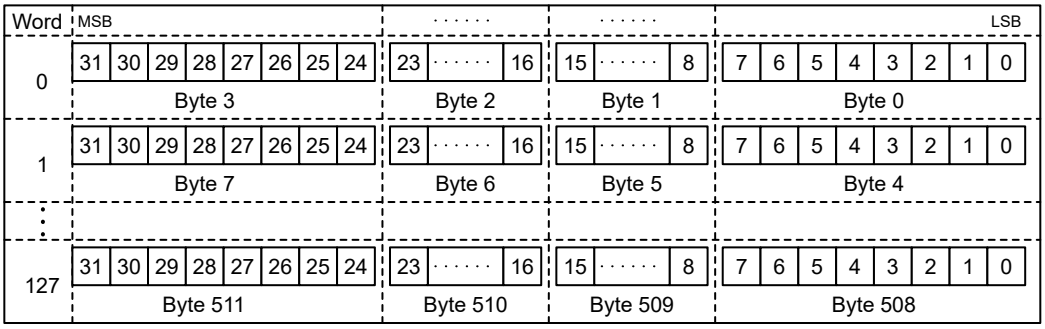


Figure 40.5 SD_BUF store data

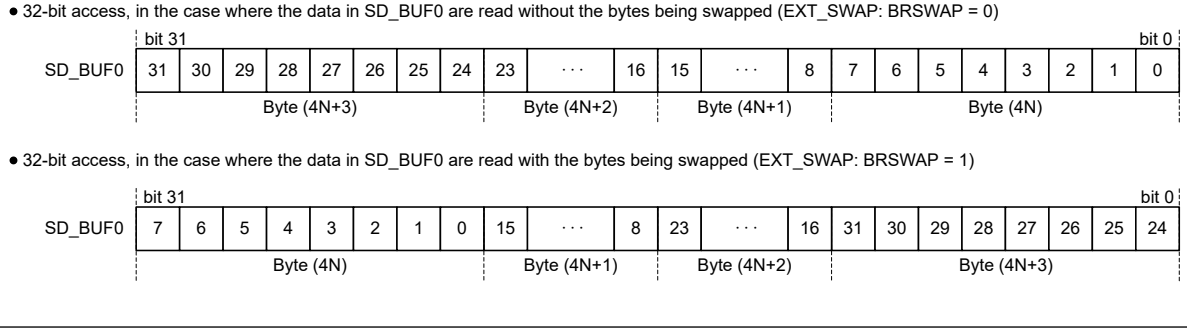


Figure 40.6 Read from SD_BUF0

40.3.2 Card Detect/Write Protect

40.3.2.1 Card detect

The SD/MMC Host Interface has two types of card detect functions.

(1) Card detect with SDnCD (n = 0, 1)

Figure 40.7 shows the timing for card detect using SDnCD. SDnCD is connected to the card socket and pulled up on the host device. The resistance of the pull-up resistor is determined by the specification of the SD/MMC host device.

(2) Card insertion

SDnCD is pulled down when a card is inserted. At this point, if SDnCD is pulled down for the Mcycle period (set in SD_OPTION), SDCDIN in SD_INFO1 is set to 1. It is cleared by writing 0.

(3) Card removal

SDnCD is pulled up when a card is removed. At this point, if SDnCD is pulled up for the Mcycle period (set in SD_OPTION), SDCDRM in SD_INFO1 is set to 1. It is cleared by writing 0.

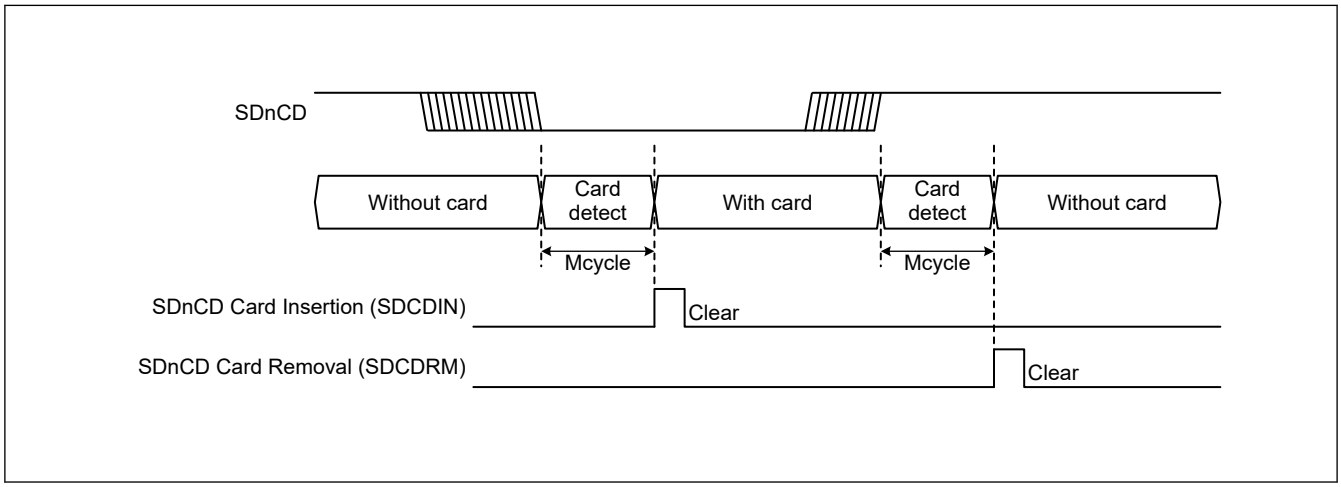


Figure 40.7 Example of card detect with SDnCD

(4) SD card detect with SDnDAT3 (n = 0, 1)

Figure 40.8 shows the timing when the SD card is detected with SDnDAT3. In addition, SDnDAT3 is pulled down by the host device, and the resistance value for pulling down is determined by the specification of the SD host device.

(5) Card insertion

When an SD card is inserted, SDnDAT3 is pulled up and SDD3IN in SD_INFO1 is set to 1. It is cleared by writing 0.

(6) Card removal

When an SD card is removed, SDnDAT3 is pulled down and SDD3RM in SD_INFO1 is set to 1. It is cleared by writing 0.

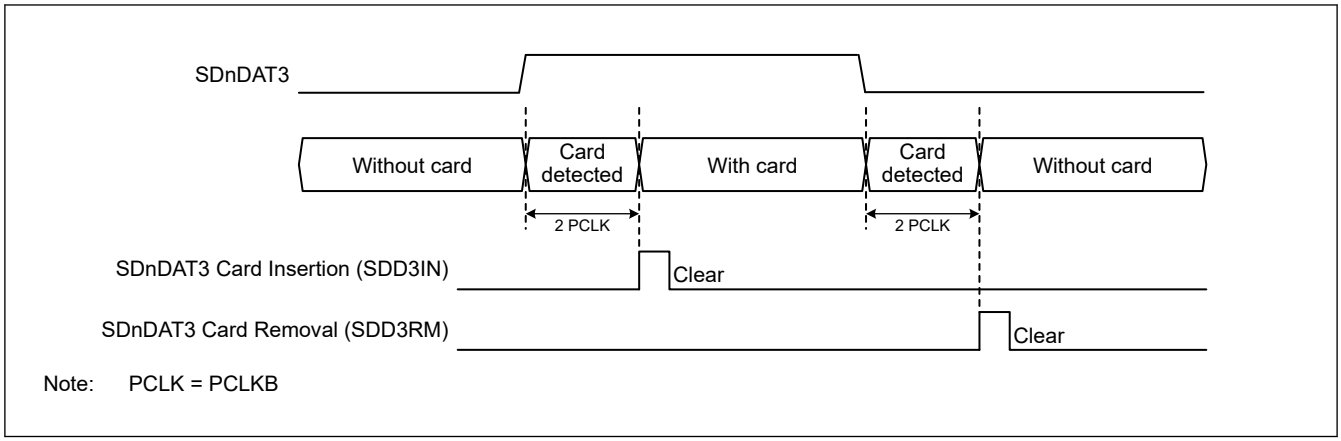


Figure 40.8 SD card detect with SDnDAT3

40.3.2.2 Write protect

The SD/MMC Host Interface has two types of write protect functions.

(1) Write protect with SDnWP (n = 0, 1)

SDnWP is connected to the card socket and pulled up or pulled down by the card insertion. The selection of pulling up or pulling down and the resistance value is determined by the specification of the SD host device. When the SDnWP state is reflected to SDWPMON in SD_INFO1, the write protect state is set after the SD card is inserted.

(2) Write protect with command

The internal write protection of the card and the lock/unlock operation of the card are realized by the command.

40.3.3 Interrupt Request and DMA Transfer Request

40.3.3.1 Interrupts

Table 40.5 lists the SDHI interrupt sources. The SDHI requests an interrupt when:

- The status flags in registers SD_INFO1, SD_INFO2, and SDIO_INFO1 set to 1.
- The associated bits in the SD_INFO1_MASK, SD_INFO2_MASK, and SDIO_INFO1_MASK registers are 0.

When clearing the status flags in registers SD_INFO1, SD_INFO2, and SDIO_INFO1, write 0 to the status flags to be cleared and write 1 to the status flags that are not being cleared.

Table 40.5 Interrupt sources

Interrupt sources	Status flag register		Interrupt mask register		Interrupt name
	Register symbol	Bit symbol	Register symbol	Bit symbol	Channel n (n = 0, 1)
Card Access Interrupt (CACI)	SD_INFO1	ACEND	SD_INFO1_MASK	ACENDM	SDHI_MMCn_ACCS
		RSPEND		RSPENDM	
	SD_INFO2	ILA	SD_INFO2_MASK	ILAM	
		BWE		BWEM	
		BRE		BREM	
		RSPTO		RSPTOM	
		ILR		ILRM	
		ILW		ILWM	
		DTO		DTOM	
		ENDE		ENDEM	
		CRCE		CRCEM	
		CMDE		CMDEM	
SDIO Access Interrupt (SDACI)	SDIO_INFO1	EXWT	SDIO_INFO1_MASK	EXWTM	SDHI_MMCn_SDIO
		EXPUB52		EXPUB52M	
		IOIRQ		IOIRQM	
Card Detect Interrupt (CDETI)	SD_INFO1	SDD3IN	SD_INFO1_MASK	SDD3INM	SDHI_MMCn_CARD
		SDD3RM		SDD3RMM	
		SDCDIN		SDCDINM	
		SDCDRM		SDCDRMM	

40.3.3.2 DMA transfer requests (SDHI_MMCn_ODMSDBREQ, n = 0, 1)

The SD/MMC Host Interface has two types of DMA transfer requests.

(1) SD_BUF write DMA transfer request

- When the BWE bit in SD_INFO2 is set to 1 while the DMAEN bit in SD_DMAEN is set to 1, the SD_BUF write DMA transfer request is asserted.
- The SD_BUF write DMA transfer request is negated when the last data in one block (based on the transfer data size set in SD_SIZE) is transferred. The SD_BUF write DMA transfer request is also negated by clearing the SDRST bit in SOFT_RST to 0 or setting the STP bit in SD_STOP to 1. However, if a communications error or timeout occurs at the DMA transfer, the SD_BUF write DMA transfer request is not negated.
- The BWE bit in SD_INFO2 is cleared after transfer of the last data in one block following a request for writing to SD_BUF by DMA transfer.
- The number of DMA transfers must be n × one block. (n = integer, one block = the transfer data size set in SD_SIZE)
- When the IOABT bit in SDIO_MODE is set to 1, the SD_BUF write DMA transfer request is negated.
- The DMA transfer request is also negated by clearing the DMAEN bit to 0. However, the DMA transfer request is asserted again when the DMAEN bit is set to 1 before writing to SD_CMD.
- Because the BWE bit in SD_INFO2 is not cleared in response to setting the STP/IOABT bit, or to a communications error or timeout, clear the bit to 0 before issuing the next command. The next request to write to SD_BUF by DMA transfer is not issued while the BWE bit is set.

(2) SD_BUF read DMA transfer request

- When the BRE bit in SD_INFO2 is set to 1 while the DMAEN bit in the SD_DMAEN register is set to 1, the SD_BUF read DMA transfer request is asserted.

- The SD_BUF read DMA transfer request is negated when the last data in one block (based on the transfer data size set in SD_SIZE) is transferred. The SD_BUF read DMA transfer request is also negated by clearing the SDRST bit in SOFT_RST to 0 or setting the STP bit in SD_STOP to 1. However, if a communications error or timeout occurs at the DMA transfer, the SD_BUF read DMA transfer request is not negated.
- The BRE bit in SD_INFO2 is cleared after transfer of the last data in one block following a request to write to SD_BUF by DMA transfer.
- The number of DMA transfers must be n × one block. (n = integer, one block = the transfer data size set in SD_SIZE)
- When the IOABT bit in SDIO_MODE is set to 1, the SD_BUF read DMA transfer request is negated.
- The DMA transfer request is also negated by clearing the DMAEN bit to 0. However, the DMA transfer request is asserted again when the DMAEN bit is set to 1 before writing to SD_CMD.
- Because the BRE bit in SD_INFO2 is not cleared in response to setting the STP/IOABT bit or in response to a communications error or timeout, clear the bit to 0 before issuing the next command. The next request to write to SD_BUF by DMA transfer is not issued while the BRE bit is set.

40.3.4 Communication Errors and Timeouts

When a communication error or timeout error occurs, depending on the type of error, the associated status flag in the SD_INFO2 register sets to 1. Also, depending on the source of the error, the associated flag in the SD_ERR_STS1 or SD_ERR_STS2 register sets to 1.

The status flags in registers SD_ERR_STS1 and SD_ERR_STS2 clear to 0 by writing to the SD_CMD register, or by setting the SOFT_RST.SDRST bit to 0.

Table 40.6 Communication errors

Communication error	Interrupt flag register		Error status register		This occurs when…
	Register symbol	Bit symbol	Register symbol	Bit symbol	
End bit error	SD_INFO2	ENDE	SD_ERR_STS1	CRCLENE	The CRC status token length is in error
				RDLENE	The read data length is in error
				RSPLENE1	The response length is in error*1
				RSPLENE0	The response length is in error*2
CRC error		CRCE		CRCTKE	The CRC status token is in error
				RDCRCE	There is a CRC error in the read data
				RSPCRCE1	There is a CRC error in the response*1
				RSPCRCE0	There is a CRC error in the response*2
Command error		CMDE		CMDE1	The command index field value for the transmitted command and received response do not match*1
				CMDE0	The command index field value for the transmitted command and received response do not match*2

Note 1. CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.
Note 2. CMD other than CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

Table 40.7 Timeouts

Timeout	Interrupt flag register		Error status register		This occurs when…
	Register symbol	Bit symbol	Register symbol	Bit symbol	
Response timeout	SD_INFO2	RSPTO	SD_ERR_STS2	RSPTO1	A response is not received even after a minimum of 640 SDHI clock cycles elapse*1
				RSPTO0	A response is not received even after a minimum of 640 SDHI clock cycles elapse*2
Data timeout (excluding response timeout)		DTO		CRCBSYTO	After the CRC status token is received, the SDHI is busy for at least the period set*3
				CRCTO	After the write data is transmitted, the CRC status token is not received even after at least the period set*3 elapses
				RDTO	After the read command is issued, the read data is not received even after at least the period set*3 elapses
					After the read data is received, the next block read data is not received even after at least the period set*3 elapses
					After the SDHI exits the read wait state, the next block read data is not received even after at least the period set*3 elapses
				BSYTO1	After CMD12 is issued during the command sequence, the SDHI is busy for at least the period set*3
				BSYTO0	After the R1b response is received, the SDHI is busy for at least the period set*3 (a command other than CMD12 is issued during the command sequence)

Note 1. CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.
Note 2. CMD other than CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP bit is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.
Note 3. The period is set in the SD_OPTION.TOP[3:0] bits.

40.3.5 Command without Data Transfer (SD/MMC)

The following legend is used for description of register read/write.

W (register name, value): Write to register

R (register name): Read from register

Figure 40.9 and Figure 40.10 show example flows.

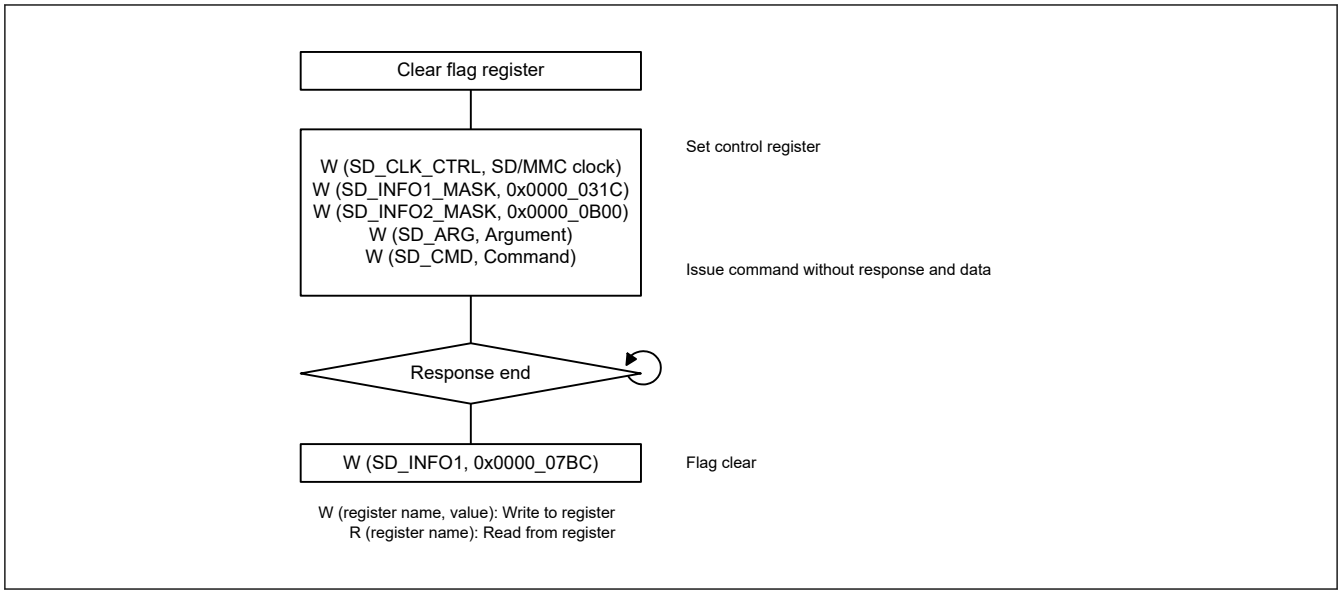


Figure 40.9 Example flow of command without response or data

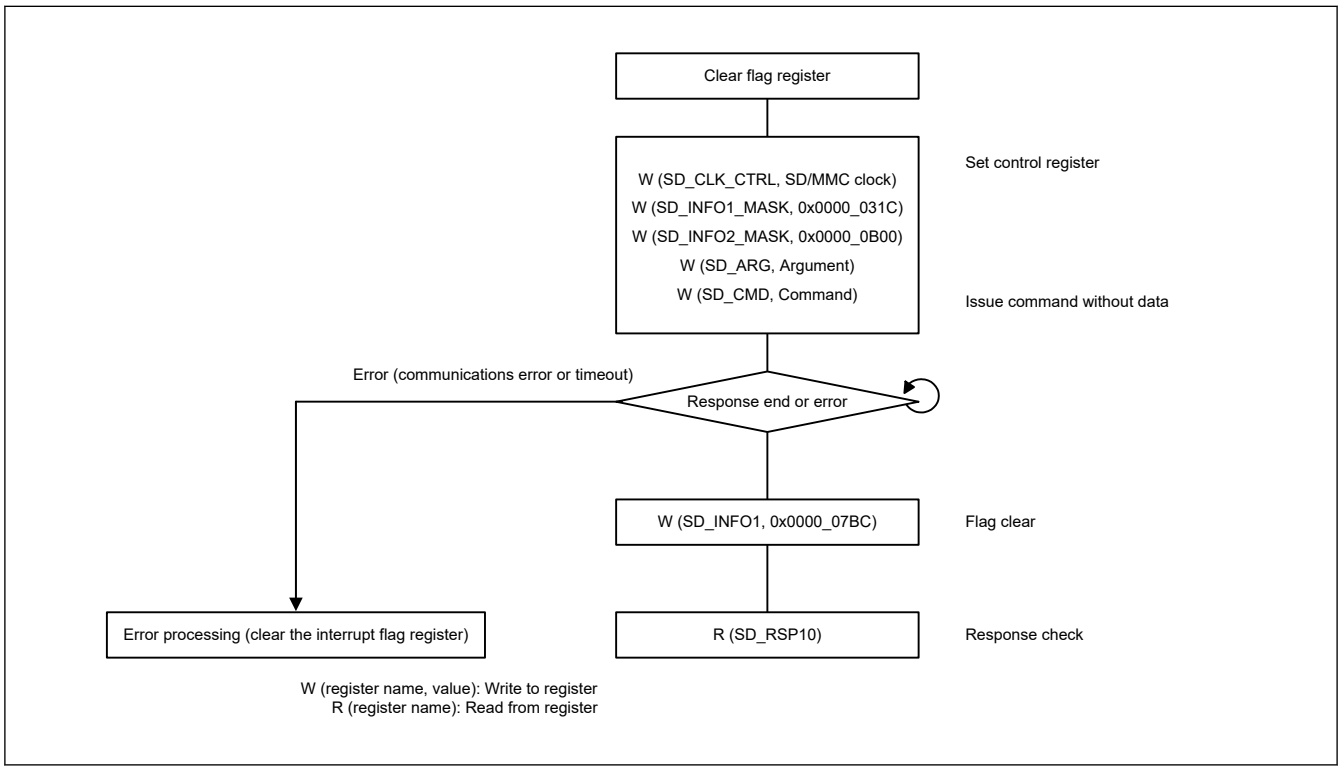


Figure 40.10 Example flow of command without data

40.3.5.1 Operation for command without data transfer

The operation is described in the following section.

(1) Command without response and data

a. Flag register clear
First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)

b. Control register set
Set the SD/MMC clock and interrupt masking. (SD_CLK_CTRL, SD_INFO1_MASK, and SD_INFO2_MASK)

c. Command issue
Set CMD argument in SD_ARG and write to SD_CMD.

Accordingly, CMD is issued, and the operation is started.

d. Flag clear
When transmission of a command is completed, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear RSPEND to 0.

(2) Command without data

a. Flag register clear
First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)

b. Control register set
Set the SD/MMC clock and interrupt masking. (SD_CLK_CTRL, SD_INFO1_MASK, and SD_INFO2_MASK)

c. Command issue
Set CMD argument in SD_ARG and write to the SD_CMD.
Accordingly, CMD is issued, and the operation is started.

d. Flag clear
When a response is received, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear RSPEND to 0.

e. Read a response from SD_RSP10. Additionally, perform error processing (clear the interrupt flag register) if a communication error or timeout occurs.

40.3.6 Single Block Read (SD/MMC)

Figure 40.11 shows an example flow of a single block read operation.

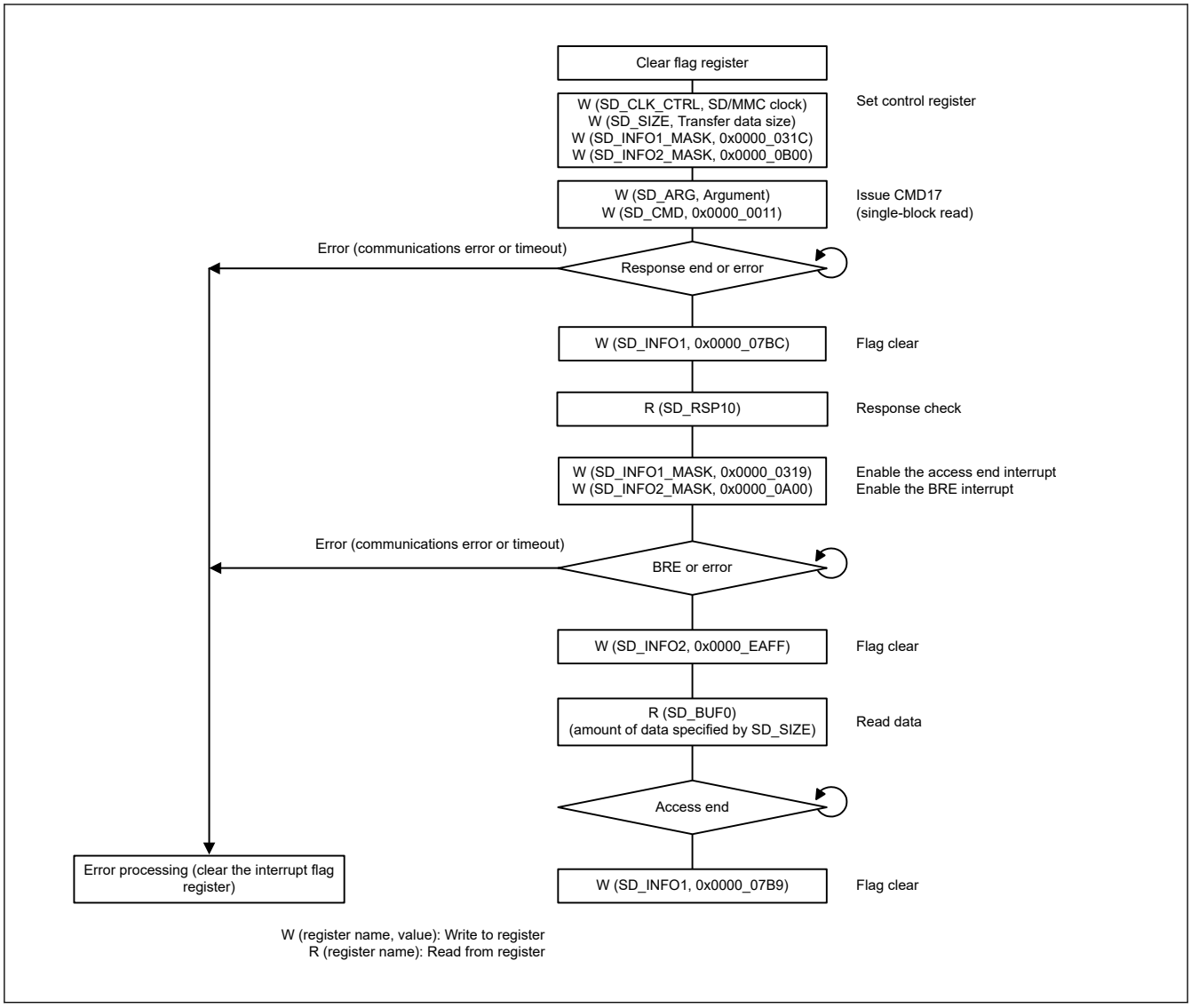


Figure 40.11 Example flow of single block read operation

40.3.6.1 Single block read operation

The operation of the single block read is described as follows:

a. Flag register clear
First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set
Set the SD/MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).

c. Command issue (CMD17)
Set CMD17 argument in SD_ARG and write 0x0000_0011 to SD_CMD. CMD17 is issued and the single block read operation is started.

d. Response check
On receiving the response, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear RSPEND to 0 and read the response from SD_RSP10. If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP or the IOABT bit in SDIO_MODE to 1. In addition, this causes CMD12 and CMD52 to not be issued. If the ACEND bit (access end) in SD_INFO1 is set, halting the command sequence also leads to the generation of an interrupt.

e. Data receive from SD card/MMC and data read

Write 0x0000_0319 to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000_0A00 to SD_INFO2_MASK to enable the BRE interrupt. When the data received from the SD card/MMC is completed, the BRE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BRE bit to 0 and read the amount of data specified in SD_SIZE from SD_BUF0.
A communication error or timeout might be generated if data is being received while reading of SD_BUF0 is in progress.

f. Operation complete
When the data read from SD_BUF0 is completed, ACEND (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear ACEND to 0 to end the single block read operation.
Additionally, perform error processing (clear the interrupt flag register) if a communication error or timeout occurs.

40.3.7 Single Block Write (SD/MMC)

Figure 40.12 shows an example flow of a single block write operation.

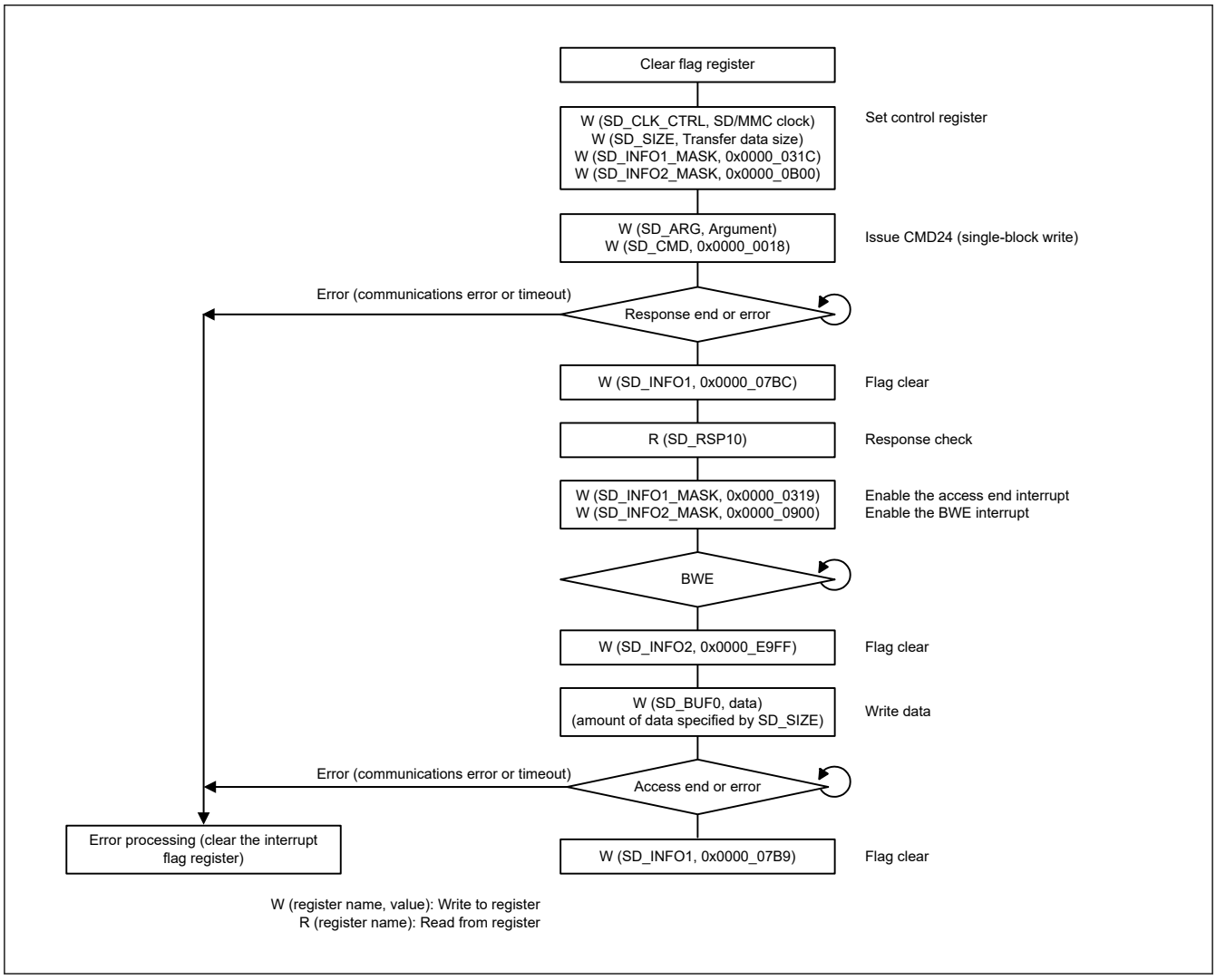


Figure 40.12 Example of single block write operation

40.3.7.1 Single block write operation

The operation of the single block write is described as follows:

a. Flag register clear
First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set

Set the SD/MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).

c. Command issue (CMD24)
Set CMD24 argument in SD_ARG and write 0x0000_0018 to SD_CMD. CMD24 is issued and the single block write operation is started.

d. Response check
On receiving the response, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear RSPEND to 0 and read the response from SD_RSP10. If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP or the IOABT bit in SDIO_MODE to 1. In addition, this causes CMD12 and CMD52 to not be issued. If the ACEND bit (access end) in SD_INFO is set, halting the command sequence also leads to the generation of an interrupt.

e. Data write and data transmit to SD card/MMC
Write 0x0000_0319 to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000_0900 to SD_INFO2_MASK to enable the BWE interrupt. When SD_BUF0 is ready for the data to be written, the BWE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified in SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card. Then, the CRC status and busy state are received from the SD card/MMC.
However, a communications error or timeout might be generated if data is being transmitted after writing to SD_BUF0.

f. Operation complete
When the CRC status and busy state are received from the SD card/MMC, ACEND (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the ACEND bit to 0 to end the single block write operation.
In addition, perform error processing (clear the interrupt flag register) if a communication error or timeout occurs.

40.3.8 Multiple Block Read (SD/MMC)

Figure 40.13 shows an example flow of a multiple block read operation.

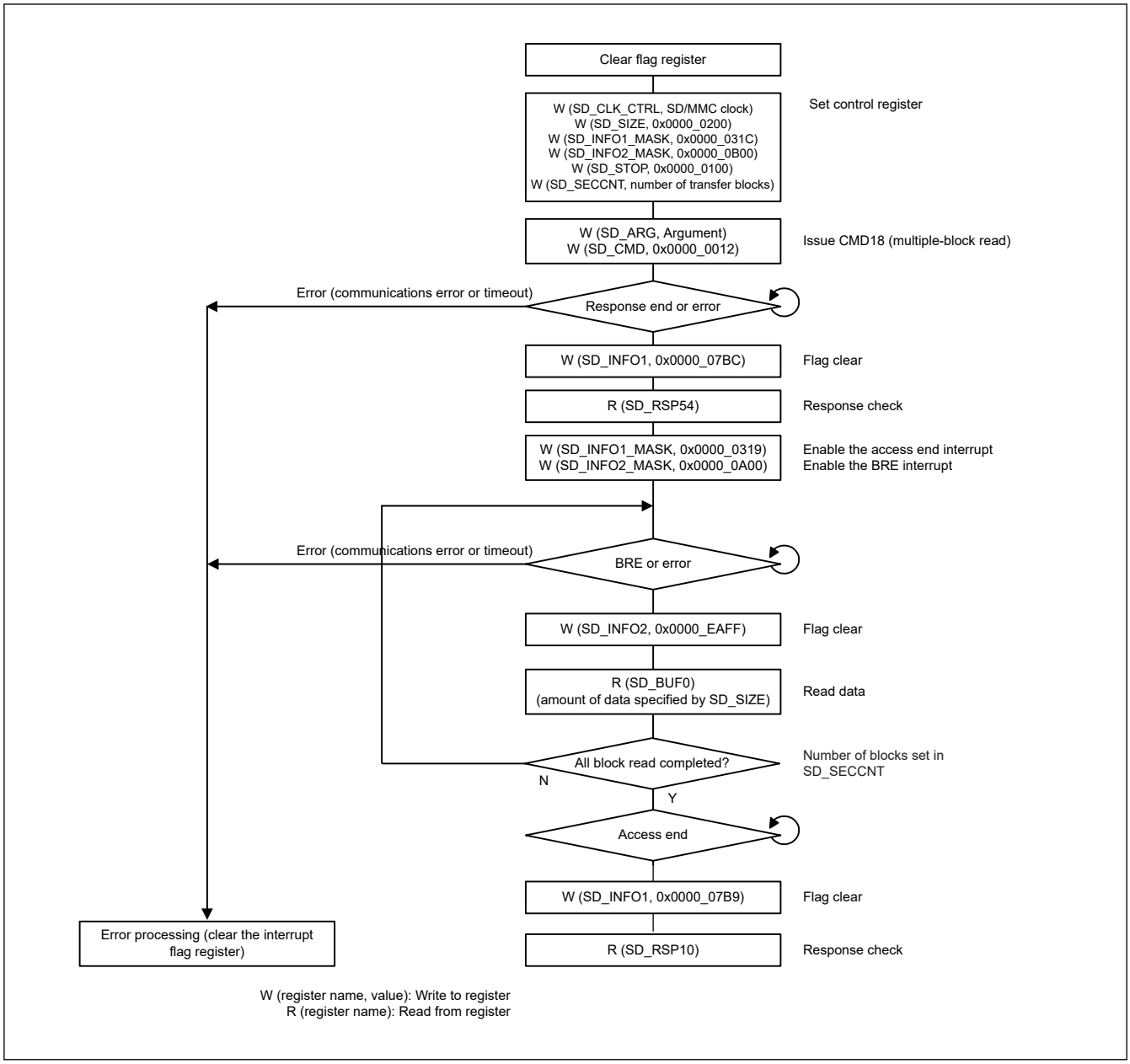


Figure 40.13 Example of multiple block read operation

40.3.8.1 Multiple block read operation

The operation of the multiple block read is described as follows:

a. Flag register clear
First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set
Set the SD/MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).
Set SEC in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.

c. Command issue (CMD18)
Set CMD18 argument in SD_ARG and write 0x0000_0012 to SD_CMD. CMD18 is issued and the multiple block read operation is started.

d. Response check
On receiving the response, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear RSPEND to 0 and read the response from SD_RSP54. If the result of response decoding is an error, the command sequence can be

halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt is enabled, an interrupt is generated when the ACEND bit (access end) bit in SD_INFO1 sets to 1 on completion of response reception. Clear the ACEND bit to 0 and read the response.

e. Data receive from SD card/MMC and data read
Write 0x0000_0319 to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000_0A00 to SD_INFO2_MASK to enable the BRE interrupt. When one-block data received from the SD card/MMC is completed, the BRE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BRE bit to 0 and read the amount of data specified in SD_SIZE from SD_BUF0. Doing this repeats transfer of the number of blocks set in SD_SECCNT. However, a communication error or timeout might be generated if data is being received while reading of SD_BUF0 is in progress. CMD12 is automatically issued to stop multiblock transfer with the number of blocks that is set to SD_SECCNT and the response is received. At this point, CMD12 argument is automatically set to 0x0000_0000.

f. Operation complete
When all-block data read and the CMD12 response received are completed, ACEND (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear ACEND to 0 to read the response. This is the end of multiple block read operation. In addition, perform error processing (clear the interrupt flag register) if a communication error or timeout occurs.

40.3.9 Multiple Block Write (SD/MMC Using Internal Timer)

Figure 40.14 shows an example flow of a multiple block write using internal timer.

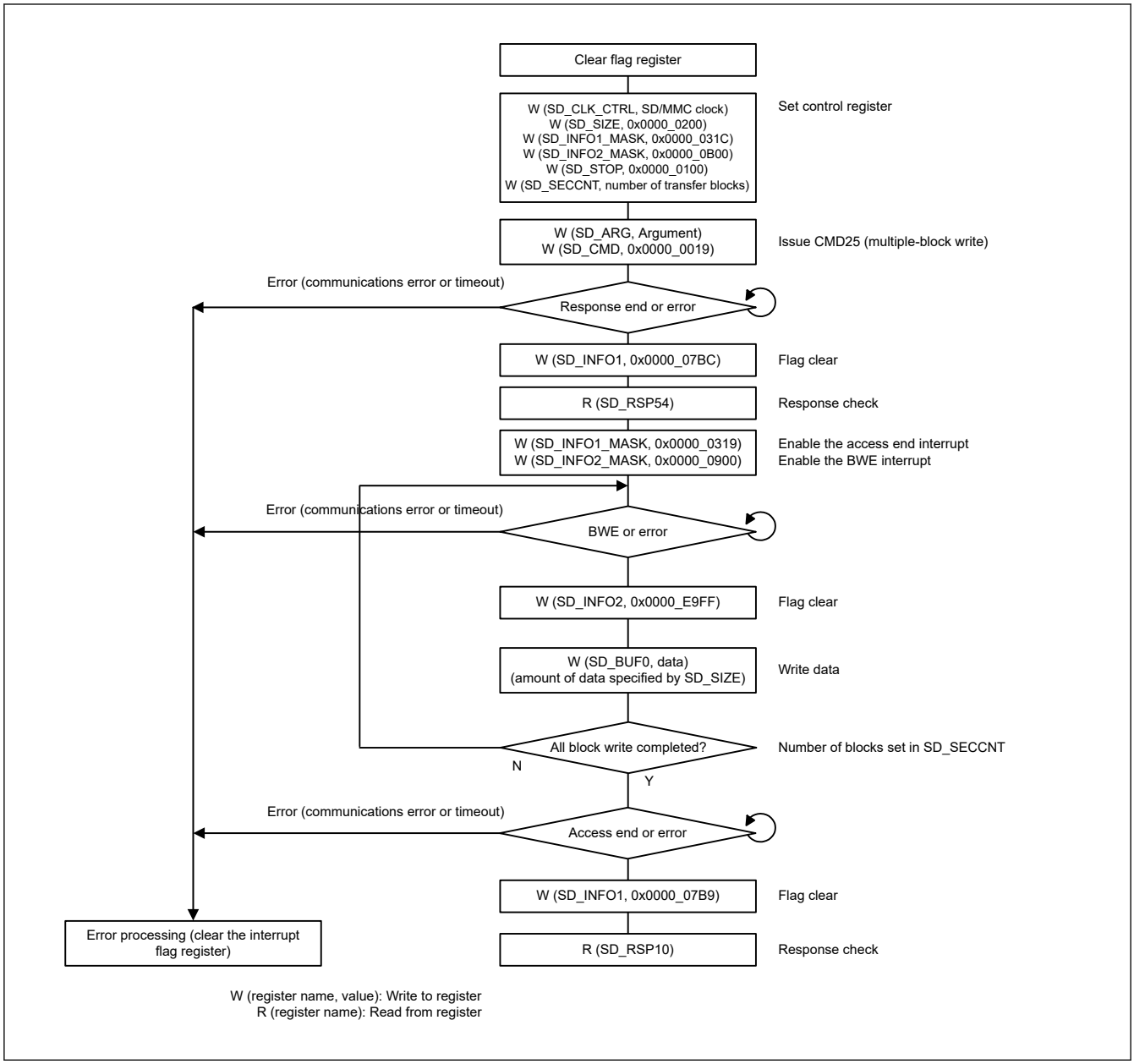


Figure 40.14 Example of multiple block write operation using internal timer

40.3.9.1 Multiple block write operation using internal timer

The operation of the multiple block write is described as follows:

a. Flag register clear
First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set
Set the SD/MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).
Set the SEC bit in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.

c. Command issue (CMD25)
Set CMD25 argument in SD_ARG and write 0x0000_0019 to SD_CMD. CMD25 is issued and the multiple block write operation is started.

d. Response check
On receiving the response, the RSPEND bit (response end) in SD_INFO1 is set to 1 to generate an interrupt.

Clear the RSPEND bit to 0 and read the response from SD_RSP54. If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt is enabled, an interrupt is generated by when the ACEND bit (access end) bit in SD_INFO1 sets to 1 on completion of response reception. Clear the ACEND bit to 0 and read the response.

e. Data write and data transmit to SD card/MMC
Write 0x0000_0319 to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000_0900 to SD_INFO2_MASK to enable the BWE interrupt. When SD_BUF0 is ready for the data to be written, the BWE bit in the SD_INFO2 resister is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified in SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card/MMC. The CRC status and busy state are received from the SD card/MMC. This repeats transfer of the number of blocks set in SD_SECCNT. However, a communication error or timeout might be generated if data is being received while writing to SD_BUF0 is in progress. CMD12 is automatically issued to stop multiblock transfer with the number of blocks which is set to SD_SECCNT and the response is received. At this point, CMD12 argument is automatically set to 0x0000_0000.

f. Operation complete
When all-block data transmit and the CRC status receive are completed, the ACEND bit (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the ACEND bit to 0 to read the response. This is the end of multiple block write operation. Additionally, perform error processing (clear the interrupt flag register) if a communications error or timeout occurs.

40.3.10 Multiple Block Write (MMC using external timer)

Figure 40.15 shows an example flow of a multiple block write using an external timer.

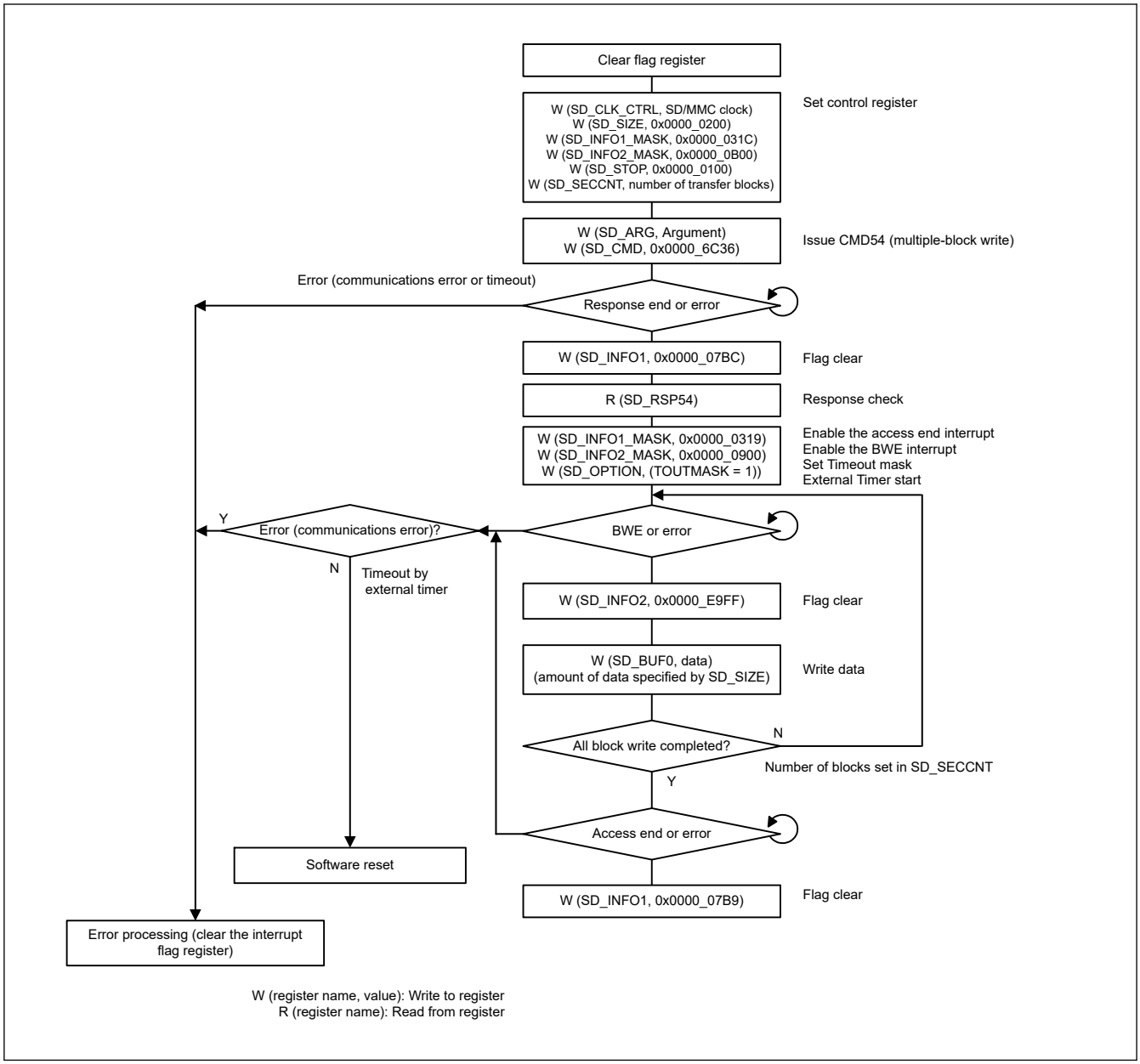


Figure 40.15 Example of multiple block write operation using external timer

40.3.10.1 Multiple block write operation using external timer

The operation of the multiple block write is described as follows:

a. Flag register clear
First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set
Set the MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).
Set the SEC bit in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.

c. Command issue (CMD54)
Set CMD54 Argument in SD_ARG and write 0x0000_6C36 to SD_CMD. CMD54 is issued and the multiple block write operation is started.

d. Response check
On receiving the response, the RSPEND bit (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the RSPEND bit to 0 and read the response from SD_RSP54. If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued

and the response received. If the command sequence is halted because the access end interrupt is enabled, an interrupt is generated by when the ACEND bit (access end) bit in SD_INFO1 sets to 1 on completion of response reception. Clear the ACEND bit to 0 and read the response.

e. Data write and data transmit to MMC
Write 0x0000_0319 to SD_INFO1_MASK to enable the access end interrupt, write 0x0000_0900 to SD_INFO2_MASK to enable the BWE interrupt and set 1 to TOUTMASK of SD_OPTION to inactivate timeout. In addition, start external timer. When SD_BUF0 is ready for the data to be written, the BWE bit in the SD_INFO2 resister is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified in SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the MMC. The CRC status and busy state are received from the MMC. Doing this repeats transfer of the number of blocks set in SD_SECCNT. However, a communication error or timeout might be generated if data is being received while writing to SD_BUF0 is in progress.

f. Operation complete
When all-block data transmit and the CRC status receive are completed, the ACEND bit (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the ACEND bit to 0 to read the response. This is the end of multiple block write operation. Additionally, perform error processing (clear the interrupt flag register) if a communications error or timeout occurs when receiving response. Execute software reset if a timeout by external timer occurs when transmitting data.

40.3.11 IO_RW_DIRECT Command (SD: CMD52)

Figure 40.16 shows an example flow of an IO_DIRECT command (CMD52) operation.

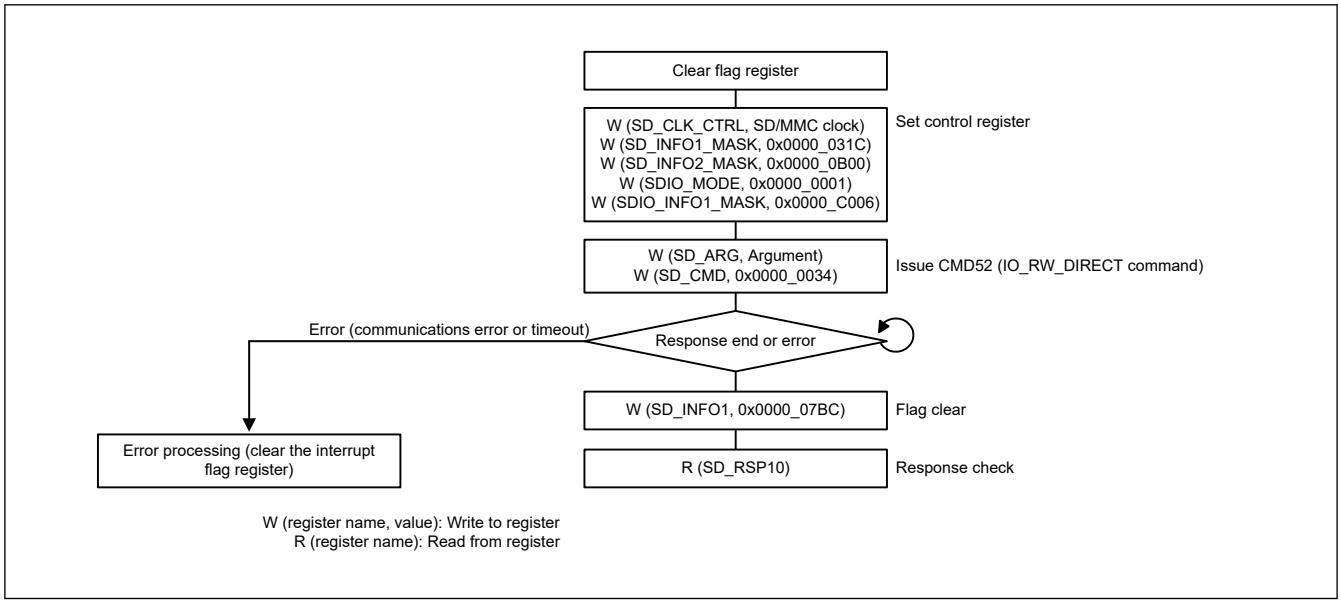


Figure 40.16 Example of IO_RW_DIRECT command (CMD52) operation

40.3.12 IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read)

Figure 40.17 shows an example flow for a CMD53 multiple block read operation.

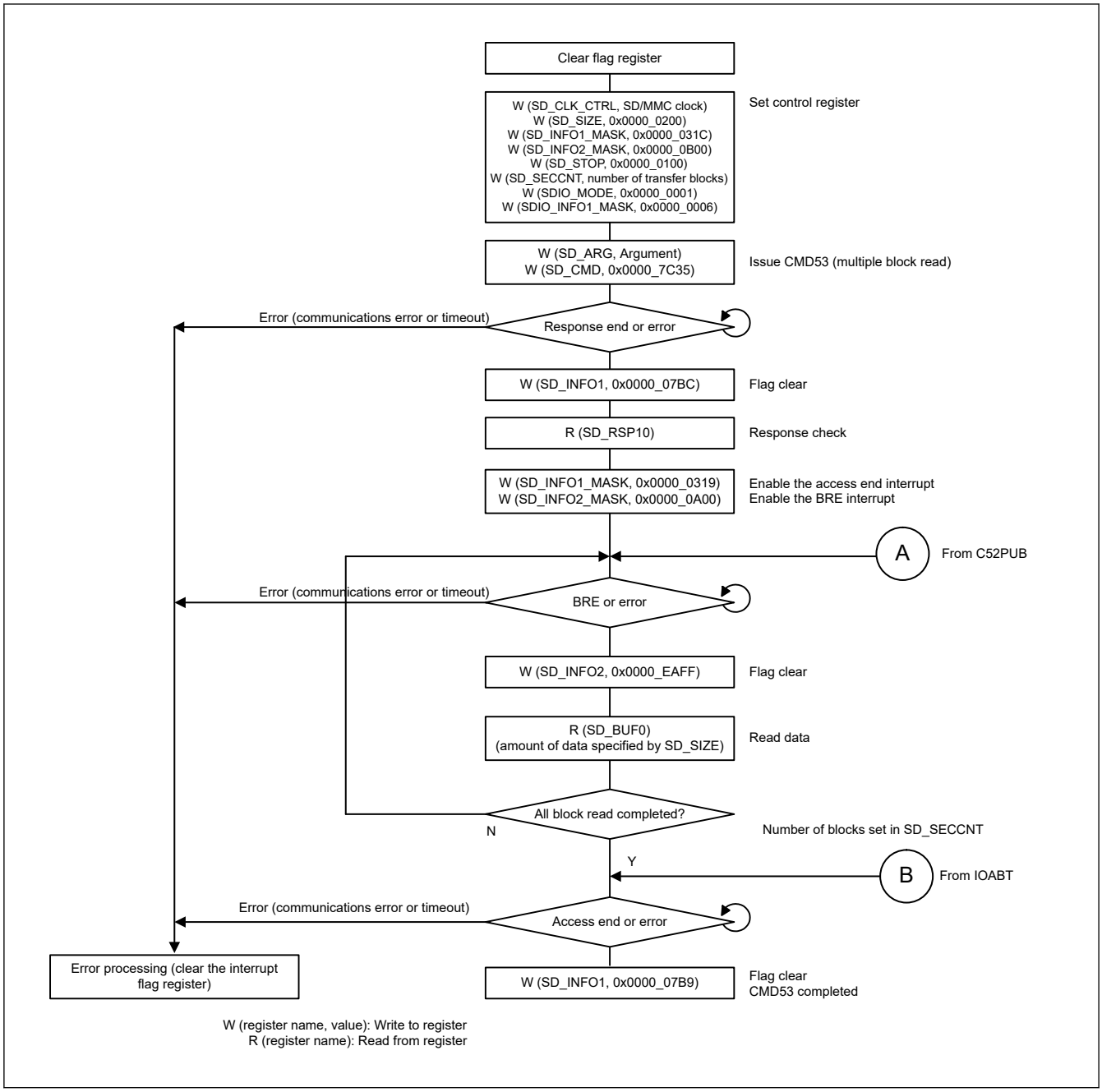


Figure 40.17 Example of IO_RW_EXTENDED command (CMD53) for multiple block read operation

Figure 40.18 shows an example flow when CMD52 (SDIO abort) is issued during a CMD53 multiple block read.

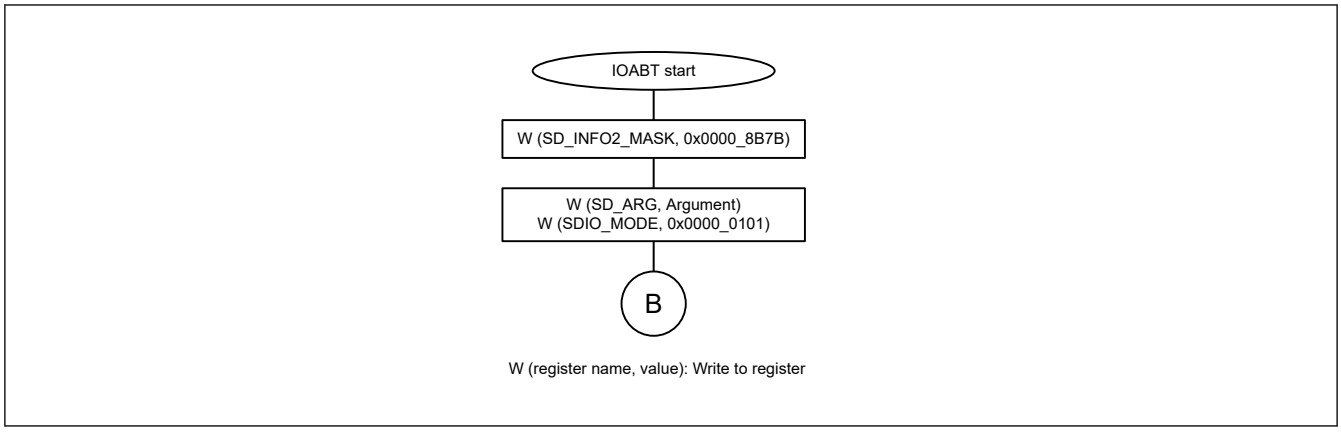


Figure 40.18 Flow when CMD52 (SDIO abort) is issued during a CMD53 multiple block read

Figure 40.19 shows an example flow when CMD52 (SDIO none abort) is issued at a CMD53 multiple block read while the SDHI is in the read wait state.

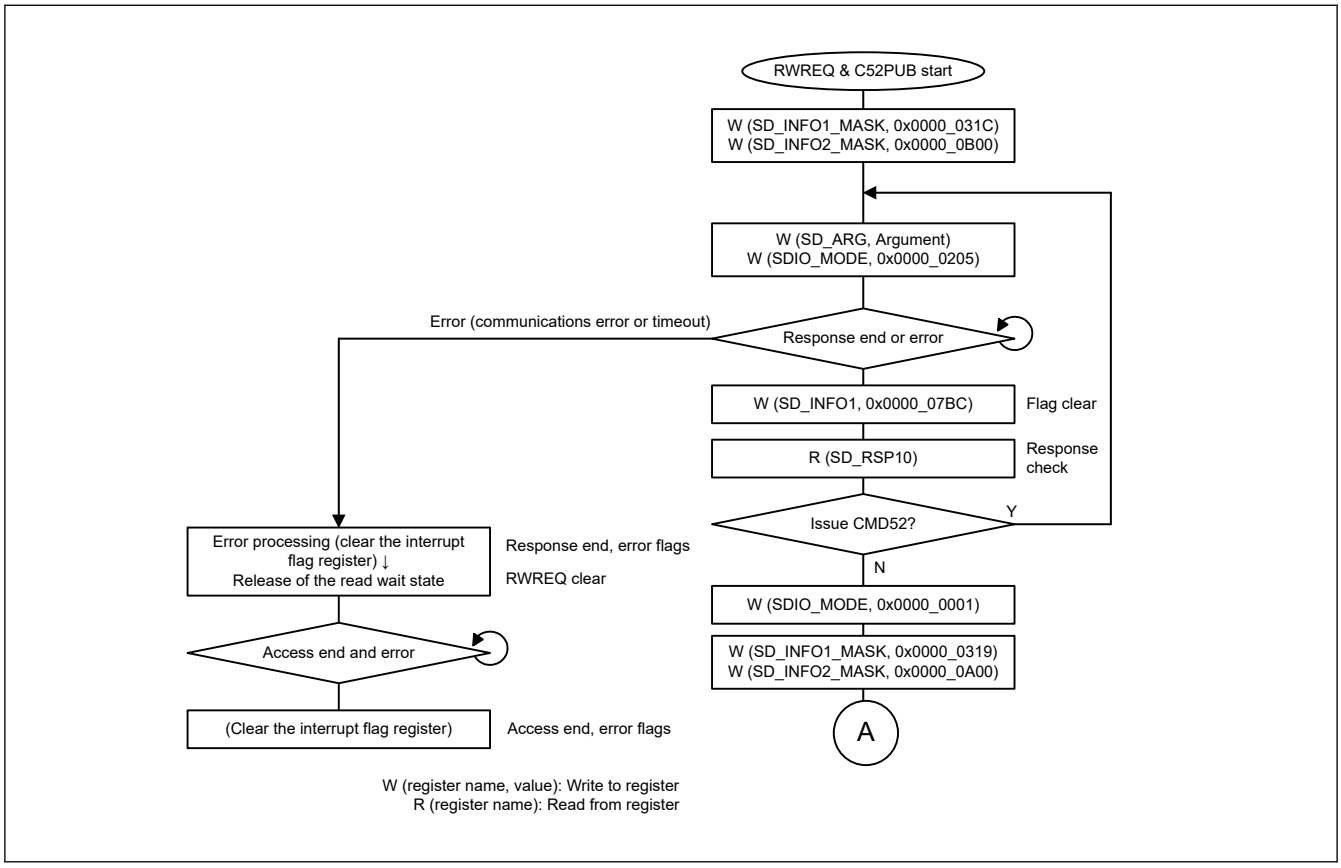


Figure 40.19 Flow when CMD52 (SDIO no abort) is issued during a CMD53 multiple block read while the SD Host Interface is in read wait state

40.3.13 IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write)

Figure 40.20 shows an example flow for a CMD53 multiple block write.

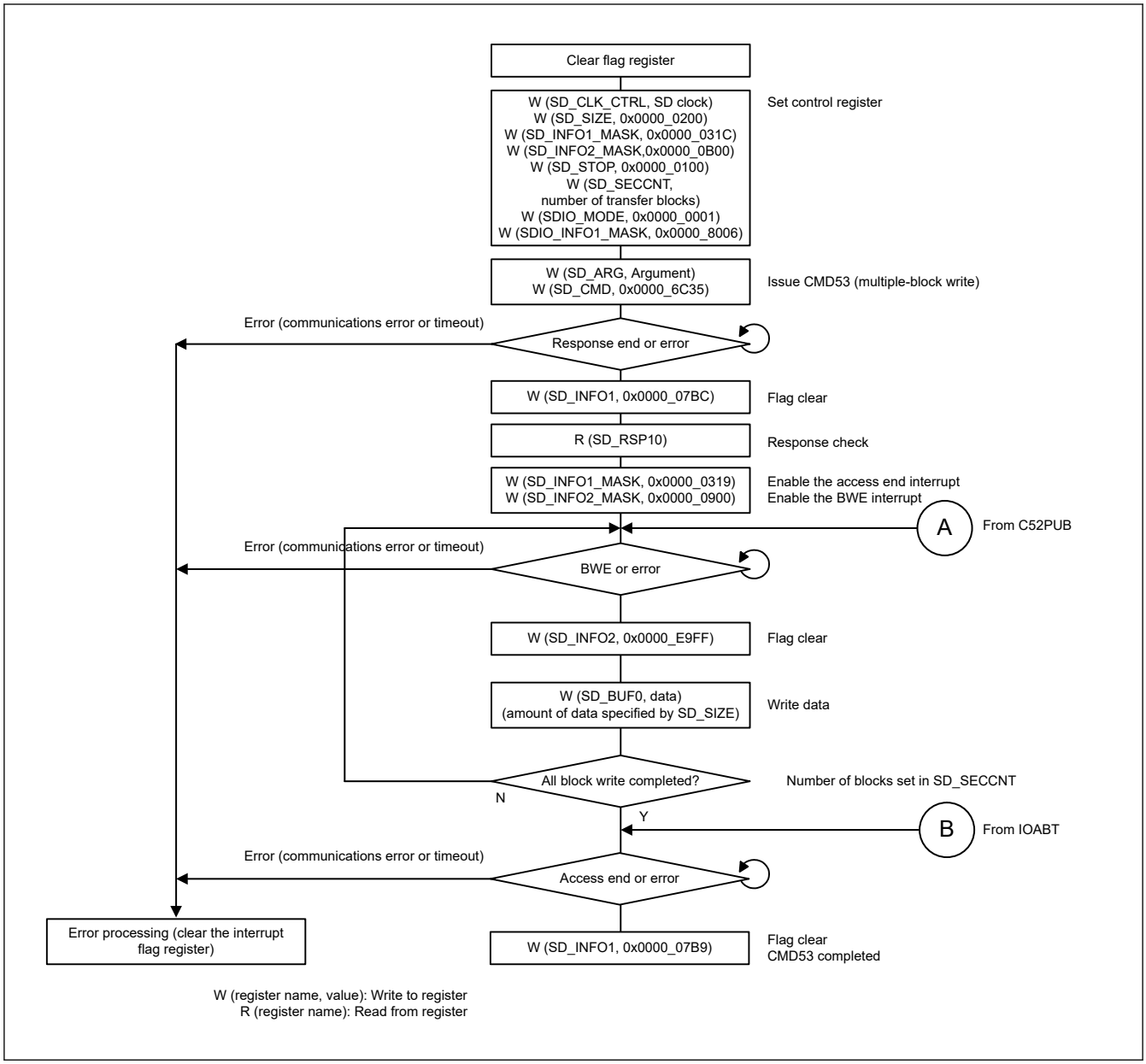


Figure 40.20 Example of IO_RW_EXTENDED command during a CMD53 multiple block write operation

Figure 40.21 shows an example flow when CMD52 (SDIO abort) is issued at CMD53 multiple block write.

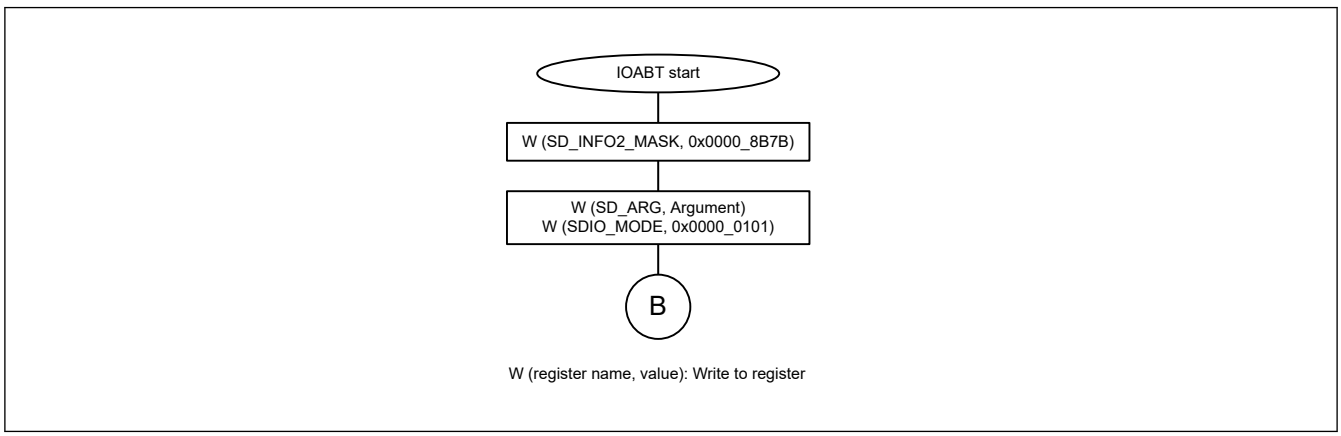


Figure 40.21 Flow when CMD52 (SDIO Abort) is issued during a CMD53 multiple block write

Figure 40.22 shows an example flow when CMD52 (SDIO none abort) is issued at CMD53 multiple block write.

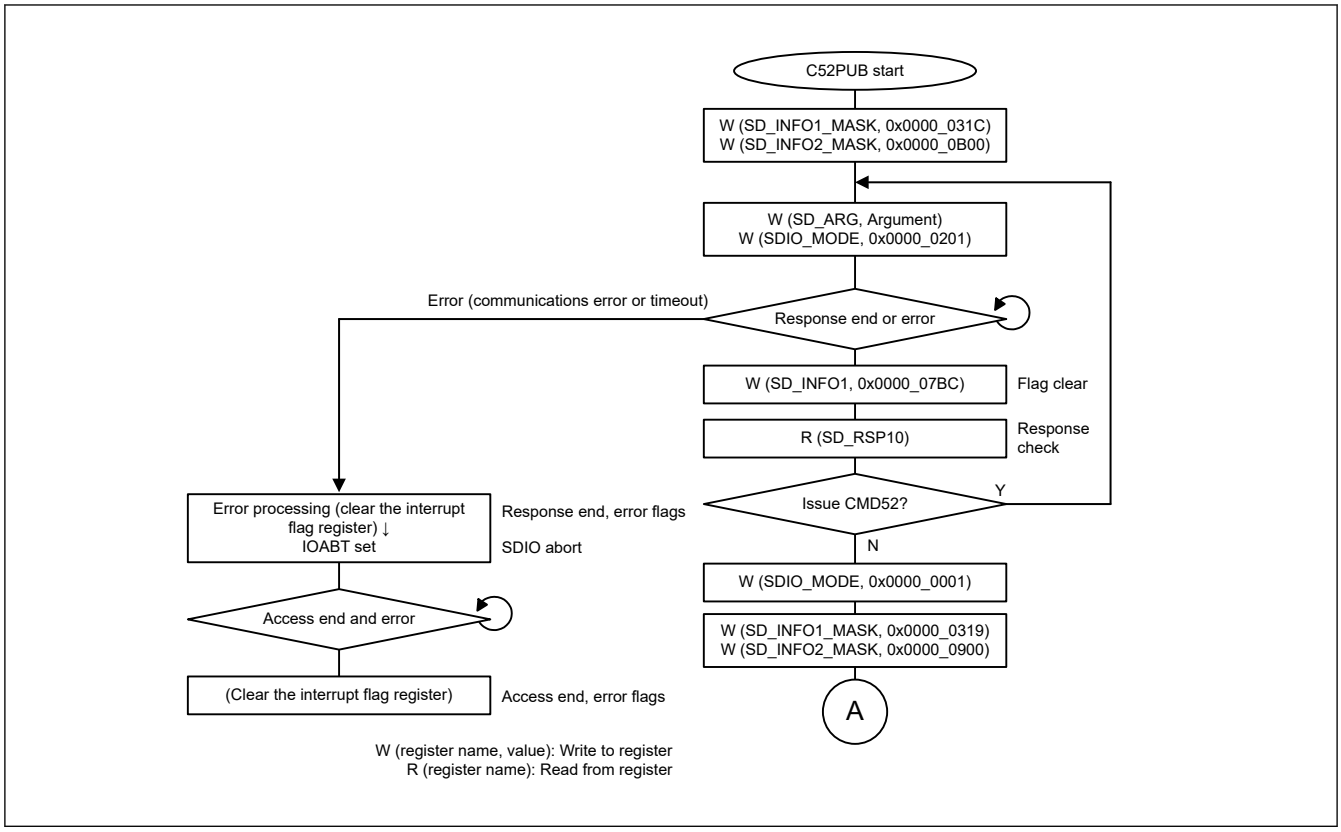


Figure 40.22 Flow when CMD52 (SDIO no abort) is issued during a CMD53 multiple block write

40.3.14 DMA Transfer (SD/MMC)

40.3.14.1 SD_BUF DMA transfer

Figure 40.23 shows an example flow for SD_BUF DMA read when CMD18 multiple block read is issued.

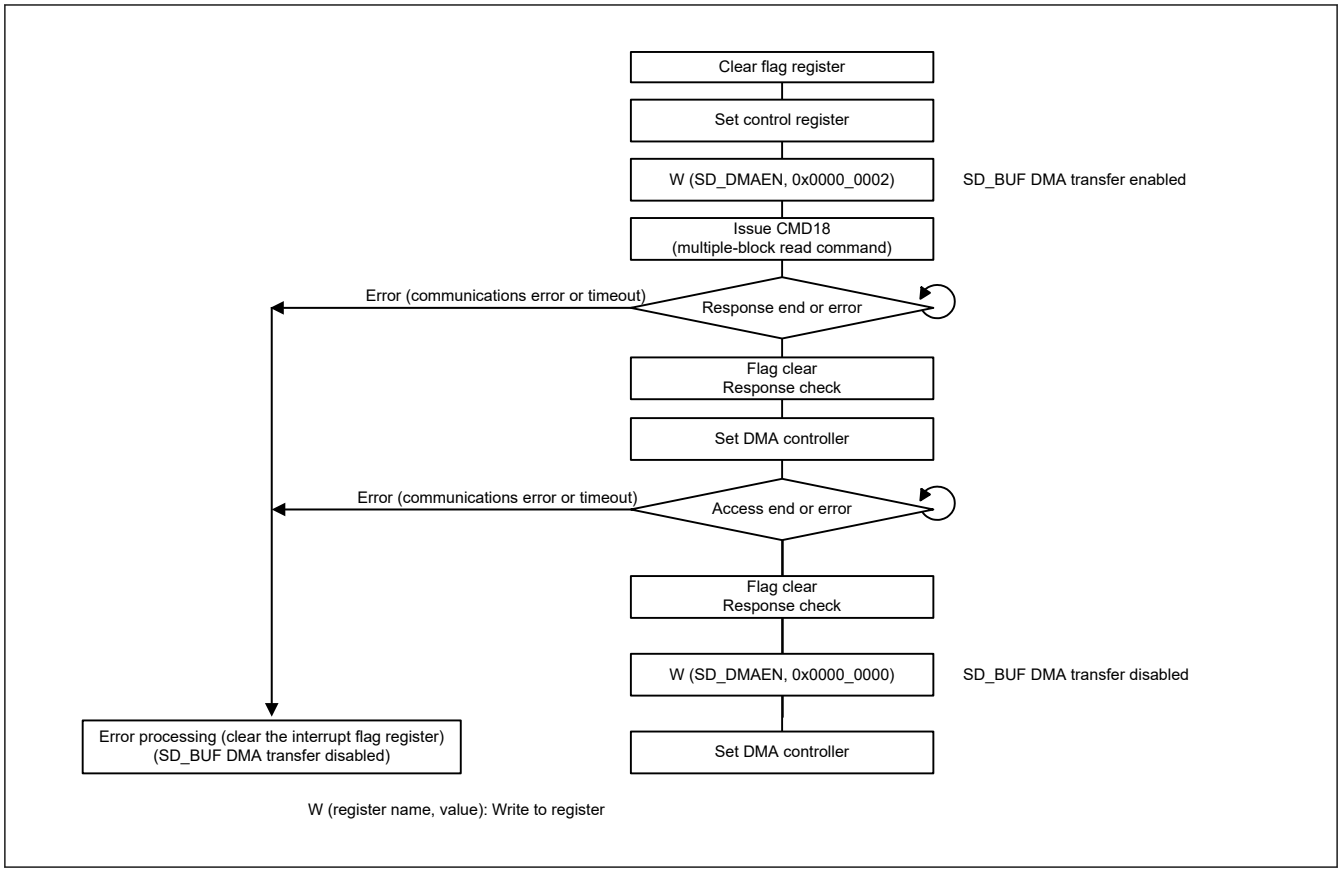


Figure 40.23 Example of SD_BUF_DMA read operation

Figure 40.24 shows an example flow for SD_BUF DMA write when CMD25 multiple block write is issued.

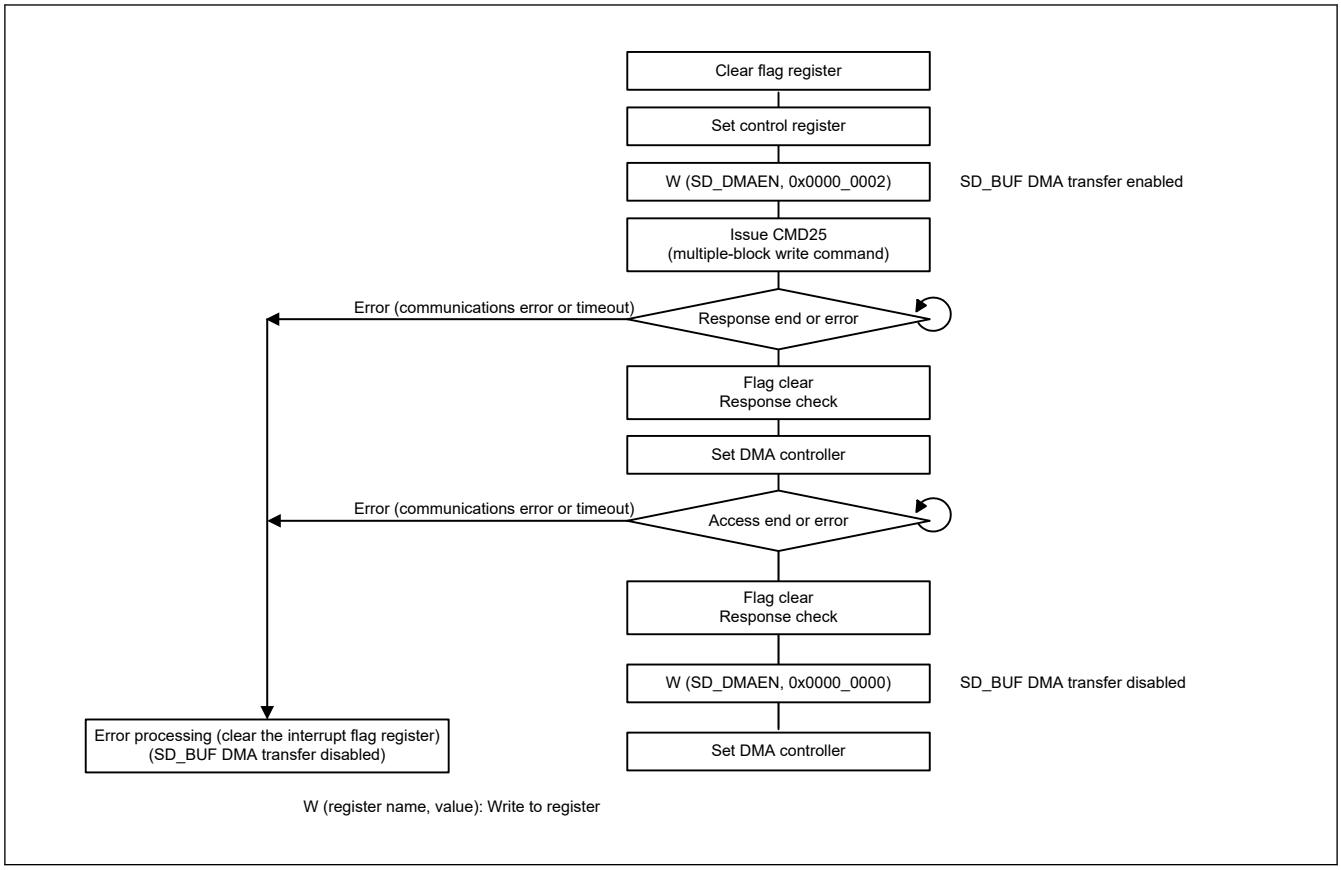


Figure 40.24 Example of SD_BUF_DMA write operation

40.3.15 Example of SD_CMD Register Setting

Table 40.8 and Table 40.9 list the SD_CMD register setting.

Table 40.8 Example SD_CMD register settings for SD (1 of 2)

Type	Command	Example SD_CMD register setting	Remark
CMD	CMD0	0x0000_0000	—
	CMD2	0x0000_0002	—
	CMD3	0x0000_0003	—
	CMD4	0x0000_0004	—
	CMD5	0x0000_0705 or 0x0000_0005	—
	CMD6	0x0000_1C06 or 0x0000_0006	—
	CMD7	0x0000_0007	When the card is placed in the deselected state, the response timeout flag sets because there is no response.
	CMD8	0x0000_0408 or 0x0000_0008	—
	CMD9	0x0000_0009	—
	CMD10	0x0000_000A	—
	CMD11	0x0000_040B or 0x0000_000B	—
	CMD12	0x0000_000C	—
	CMD13	0x0000_000D	—
	CMD15	0x0000_000F	—
	CMD16	0x0000_0010	—
	CMD17	0x0000_0011	—
	CMD18	0x0000_0012	With automatic CMD12
	CMD20	0x0000_0514 or 0x0000_0014	—
	CMD24	0x0000_0018	—
	CMD25	0x0000_0019	With automatic CMD12
	CMD27	0x0000_001B	—
	CMD28	0x0000_001C	—
	CMD29	0x0000_001D	—
	CMD30	0x0000_001E	—
	CMD32	0x0000_0020	—
	CMD33	0x0000_0021	—
	CMD38	0x0000_0026	—
	CMD42	0x0000_002A	—
	CMD52	0x0000_0434 or 0x0000_0034	—
	CMD53	0x0000_1C35	Single read
		0x0000_0C35	Single write
		0x0000_7C35	Multiple read
		0x0000_6C35	Multiple write
		0x0000_0035	The value on the left can be set for both single and multiple operations. However, the CF39 bit in SD_ARG must be set as follows. Read: 0 Write: 1
	CMD55	0x0000_0037	—
	CMD56	0x0000_0038	—

Table 40.8 Example SD_CMD register settings for SD (2 of 2)

Type	Command	Example SD_CMD register setting	Remark
ACMD	ACMD6	0x0000_0046	—
	ACMD13	0x0000_004D	—
	ACMD22	0x0000_0056	—
	ACMD23	0x0000_0057	—
	ACMD41	0x0000_0069	—
	ACMD42	0x0000_006A	—
	ACMD51	0x0000_0073	—

Table 40.9 Example SD_CMD register settings for MMC

Type	Command	Example SD_CMD register setting	Remark
CMD	CMD0	0x0000_0000	—
	CMD1	0x0000_0701	—
	CMD2	0x0000_0002	—
	CMD3	0x0000_0003	—
	CMD4	0x0000_0004	—
	CMD5	0x0000_0505	—
	CMD6	0x0000_0506	(with response busy)
		0x0000_0406	(without response busy)
	CMD7	0x0000_0007	When the card is placed in the deselected state, the response timeout flag sets because there is no response.
	CMD8	0x0000_1C08	—
	CMD9	0x0000_0009	—
	CMD10	0x0000_000A	—
	CMD12	0x0000_000C	—
	CMD13	0x0000_000D	—
	CMD14	0x0000_1C0E	Required setting: SD_IFMODE = 0x0000_0100 (CRC check is invalid)
	CMD15	0x0000_000F	—
	CMD16	0x0000_0010	—
	CMD17	0x0000_0011	—
	CMD18	0x0000_7C12	Pre-defined
	CMD19	0x0000_0C13	Required setting: SD_IFMODE = 0x0000_0100 (CRC check is invalid)
	CMD21	0x0000_1C15	DDR mode is inhibited
	CMD23	0x0000_0017	—
	CMD24	0x0000_0018	—
	CMD25	0x0000_6C19	Pre-defined
	CMD26	0x0000_0C1A	—
	CMD27	0x0000_001B	—
	CMD28	0x0000_001C	—
	CMD29	0x0000_001D	—
	CMD30	0x0000_001E	—
	CMD31	0x0000_1C1F	—
	CMD35	0x0000_0423	—
	CMD36	0x0000_0424	—
	CMD38	0x0000_0026	—
	CMD39	0x0000_0427	—
	CMD40	0x0000_0428	—
	CMD42	0x0000_002A	—
	CMD49	0x0000_0C31	—
	CMD53	0x0000_7C35	—
	CMD54	0x0000_6C36	—
	CMD55	0x0000_0037	—
	CMD56	0x0000_0038	—

40.4 Usage Notes

40.4.1 SD_BUF Illegal Write Access (SD/MMC)

When writing data to SD_BUF0 after the single block write or multi block write command is issued, the data of the size specified in SD_SIZE must be written.

If the data exceeds the size specified in SD_SIZE is written, the ERR4 bit in SD_INFO2 is set to 1. In addition, the data written to SD_BUF0 might not be transmitted and the SD_CLK_CTRLEN bit in SD_INFO2 is held at the value of 0. If this occurs, clearing the SDRST bit in SOFT_RST to 0 and then restoring its value to 1 clears the SD_CLK_CTRLEN bit to 1.

However, this does not apply to the single byte or three bytes when the SD_SIZE setting is odd, or to the fraction of bytes when the SD_SIZE setting is even (the 2 bytes that are not in a 4-byte unit), because the portion of dummy data writing is regarded as excess data and ignored.

40.4.2 Block Number Constraint for Multiple Block Read (SD)

When performing a multiple block read of one or two blocks, depending on the timing with which the SD card response register is read, the response value might not be read properly. To prevent this, do one of the following:

1. When receiving one or two blocks of data, use single block reading.
2. Read the response to CMD18 from SD_RSP54.

40.4.2.1 Mechanism of incorrect reading

Figure 40.25 shows the processing flows of the SDHI (hardware) operation and software operation when a multiple block read is performed on two blocks. As shown in the incorrect operation in Figure 40.25, when an interrupt is generated on reception of the CMD18 response and the timing with which the SD card response register (SD_RSP10) is read by the interrupt is delayed, the data during the CMD12 response reception or the CMD12 response might be read. This problem does not occur for multiple block reads of three or more blocks, because CMD12 is not issued until the block of data is read. The problem also does not occur for multiple block writes, because the CMD25 response is read before the block of data is sent.

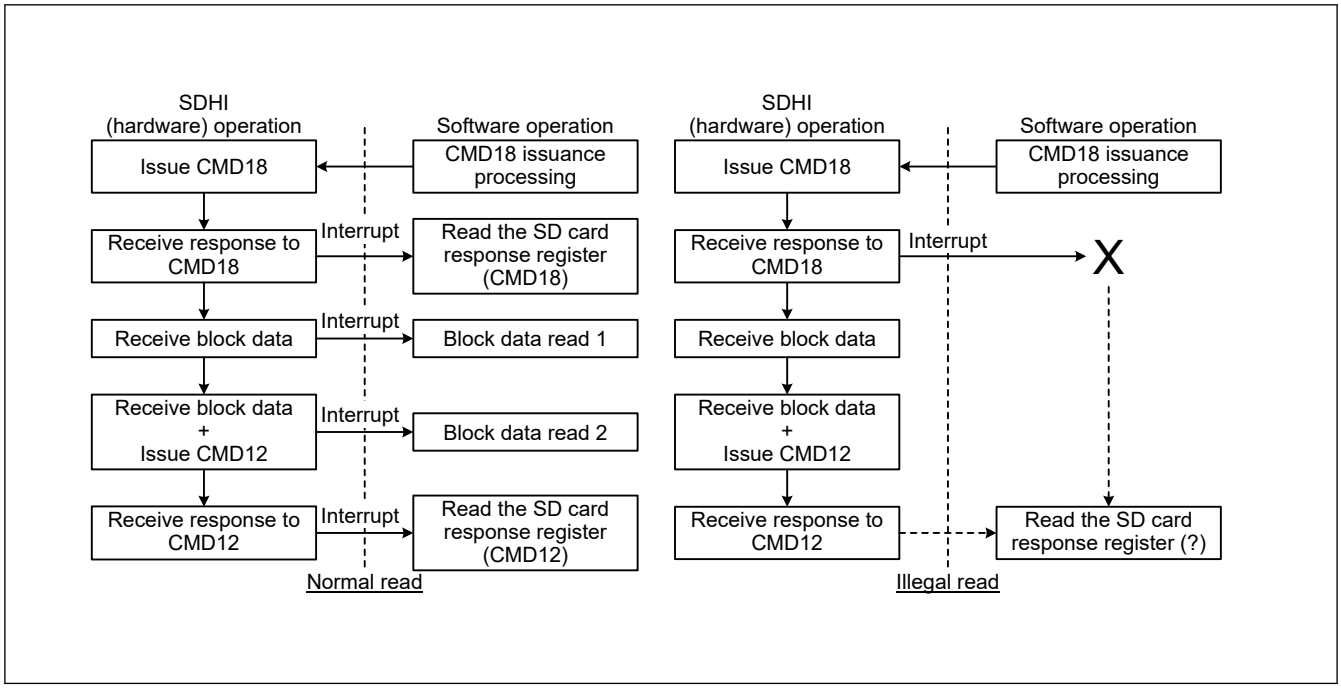


Figure 40.25 Multiple block read operation flow chart (two blocks)

40.4.3 Automatic Control of SD/MMC Clock Output (SD/MMC)

In the SD Card/MMC standard, 74 cycles of SD/MMC clock must be output before initialization of the card. For this reason, use automatic control of SD/MMC clock output after 74 cycles of SD/MMC clock are output. In addition, if

automatic control of SD/MMC clock output was in use, SD/MMC clock output is stopped on completion of the sequence for a communications error or timeout. When state transitions within the SD card/MMC are necessary after completion of the sequence, release automatic control of SD/MMC clock output and restart supply of the SD/MMC clock to the SD card/MMC.

40.4.4 Control of the C52PUB Setting for Multiple Block Write (SD)

If the C52PUB bit in SDIO_MODE is set to 1 during a sequence of multiple block write because of CMD53, CMD52 is not issued until SD_BUF becomes empty. For this reason, set the C52PUB bit after suspending writing to SD_BUF by using one of the following procedures, as appropriate:

(a) When DMA transfer is not in use

1. Before setting the C52PUB bit, suspend writing to SD_BUF by making the setting in SD_INFO2 to disable BWE interrupts.
2. Set the C52PUB bit in SDIO_MODE to 1 (so that CMD52 is issued when SD_BUF becomes empty).
3. After the RSPEND interrupt processing in SD_INFO1 because the issuing of CMD52 is completed, restart writing to SD_BUF by making the setting in SD_INFO2 to enable BWE interrupts.

(b) When DMA transfer is in use

1. Every time DMA transfer of the value set in SD_SIZE × n blocks (where n = 1, 2,…) proceeds, suspend writing to SD_BUF by DMA transfer before the C52PUB bit is set.
2. Set the C52PUB bit in SDIO_MODE to 1 (so that CMD52 is issued when SD_BUF becomes empty).
3. After the RSPEND interrupt processing in SD_INFO1 because the issuing of CMD52 is completed, restart writing to SD_BUF by DMA transfer.

40.4.5 Notes on SD_CLK_CTRL Register Settings (SD/MMC)

When the SD_CLK_CTRLEN bit in SD_INFO2 is 0, SD_CLK_CTRL cannot be written to. Before writing to SD_CLK_CTRL, you must check that the SD_CLK_CTRLEN bit in SD_INFO2 is 1.

40.4.6 Specification Limitations

1. The Suspend/Resume operation of the SDIO is not supported.
2. The SPI bus is not supported. (SD/MMC)
3. The shared bus and 8-bit SD bus of the embedded SDIO are not supported.
4. Stream transfer of MMC is not supported.
5. High Priority Interrupt (HPI) of MMC is not supported.
6. Boot Operation/Alternative Boot Operation of MMC is not supported.
7. Open-ended multiple block transfer of MMC is not supported.

40.4.7 STP Bit Setting during Multiple Block Read (SD/MMC)

During execution of multiple block read with automatic CMD12 execution by setting the SEC bit in SD_STOP to 1, even if the STP bit in SD_STOP is set to 1 to forcibly stop the execution, the command sequence might not stop depending on the timing of setting the STP bit.

To avoid this, when setting the STP bit in SD_STOP to 1 during multiple block transfer, clear the SEC bit in SD_STOP to 0 at the same time. (Even when the SD_CLK_CTRLEN bit in SD_INFO2 is 0, change the SEC bit from 1 to 0.)

When the command sequence does not stop because the SEC bit was not cleared to 0, the command sequence can be stopped by clearing the SDRST bit in SOFT_RST to 0.

When forcibly terminating the CMD53 multiple block transfer through the IOABT bit in SDIO_MODE, you must leave the SEC bit in SD_STOP as 1.

40.4.8 Register Setting Notes

1. All registers in section 40.2. Register Descriptions are accessed in 32-bit access-only.
2. When setting registers, set them after the I/O Port Register setting.

40.4.9 Module-stop function

SDHI operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The SDHI module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

41. Cyclic Redundancy Check (CRC)

41.1 Overview

The Cyclic Redundancy Check (CRC) calculator generates CRC codes to detect errors in the data. The bit order of CRC calculation results can be switched for LSB-first or MSB-first communication. Additionally, various CRC-generation polynomials are available. The snoop function allows monitoring reads from and writes to specific addresses. This function is useful in applications that require CRC code to be generated automatically in certain events, such as monitoring writes to the serial transmit buffer and reads from the serial receive buffer.

Table 41.1 lists the CRC calculator specifications and Figure 41.1 shows a block diagram.

Table 41.1 CRC calculator specifications

Item	Description	
Data size	8-bit	32-bit
Data for CRC calculation*1	CRC code generated for data in 8n-bit units (where n is a natural number)	CRC code generated for data in 32n-bit units (where n is a natural number)
CRC processor unit	Operation executed on 8 bits in parallel	Operation executed on 32 bits in parallel
CRC generating polynomial	One of three generating polynomials that is selectable: [8-bit CRC] • $X^8 + X^2 + X + 1$ (CRC-8) [16-bit CRC] • $X^{16} + X^{15} + X^2 + 1$ (CRC-16) • $X^{16} + X^{12} + X^5 + 1$ (CRC-CCITT)	One of two generating polynomials that is selectable: [32-bit CRC] • $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ (CRC-32) • $X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$ (CRC-32C)
CRC calculation switching	The bit order of CRC calculation results can be switched for LSB-first or MSB-first communication	
Module-stop function	Module-stop state can be set to reduce power consumption	
CRC snoop	Monitor reads from and writes to a certain register address	
TrustZone Filter	Security and Privilege attribution can be set	

Note 1. This function cannot divide data used in CRC calculations. Write data in 8-bit or 32-bit units.

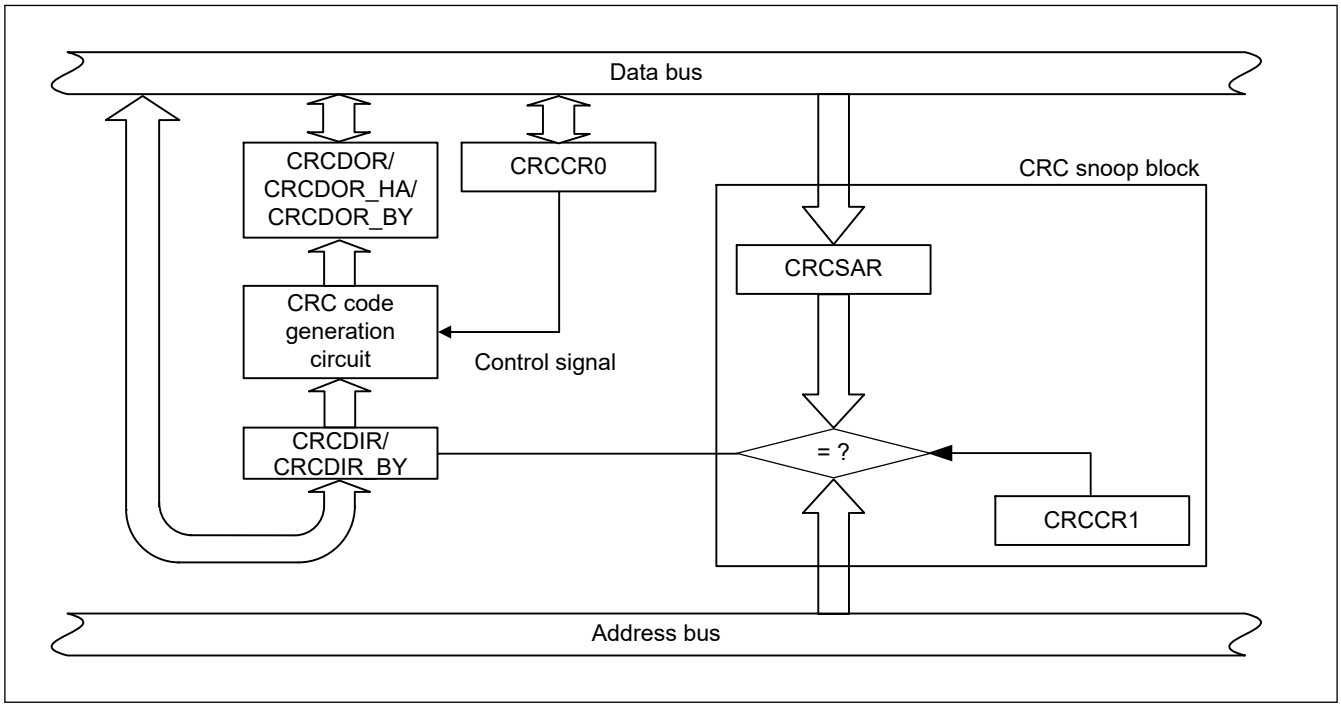


Figure 41.1 CRC calculator block diagram

41.2 Register Descriptions

41.2.1 CRCCR0 : CRC Control Register 0

Base address: CRC = 0x4031_0000
CRC_NS = 0x5031_0000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DORCLR	LMS	—	—	—	GPS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	GPS[2:0]	CRC Generating Polynomial Switching 0 0 1: 8-bit CRC-8 ($X^8 + X^2 + X + 1$) 0 1 0: 16-bit CRC-16 ($X^{16} + X^{15} + X^2 + 1$) 0 1 1: 16-bit CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) 1 0 0: 32-bit CRC-32 ($X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$) 1 0 1: 32-bit CRC-32C ($X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$) Others: No calculation is executed	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	LMS	CRC Calculation Switching 0: Generate CRC code for LSB-first communication 1: Generate CRC code for MSB-first communication	R/W
7	DORCLR	CRCDOR/CRCDOR_HA/CRCDOR_BY Register Clear 0: No effect 1: Clear the CRCDOR/CRCDOR_HA/CRCDOR_BY register	W

Note: S-TYPE3, P-TYPE3

GPS[2:0] bits (CRC Generating Polynomial Switching)

The GPS[2:0] bits select the CRC generating polynomial.

LMS bit (CRC Calculation Switching)

The LMS bit selects the bit order of generated CRC code. Transmit the lower byte of the CRC code first for LSB-first communication and the upper byte first for MSB-first communication. For details on transmitting and receiving CRC code, see section 41.3. Operation.

DORCLR bit (CRCDOR/CRCDOR_HA/CRCDOR_BY Register Clear)

Write 1 to the DORCLR bit to set the CRCDOR/CRCDOR_HA/CRCDOR_BY register to 0x00000000. This bit is read as 0. Only 1 can be written to it.

41.2.2 CRCCR1 : CRC Control Register 1

Base address: CRC = 0x4031_0000
CRC_NS = 0x5031_0000

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CRCSEN	CRCSWR	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	CRCSWR	Snoop-On-Write/Read Switch 0: Snoop-on-read 1: Snoop-on-write	R/W
7	CRCSEN	Snoop Enable 0: Disabled 1: Enabled	R/W

Note: S-TYPE3, P-TYPE3

CRCSWR bit (Snoop-On-Write/Read Switch)

The CRCSWR bit selects the direction of access in the CRC snoop function.

When this bit is set to 0 (initial value), the CRC snoop operation to read a specific register is enabled. Similarly, when this bit is set to 1, the CRC snoop operation to write a specific register is enabled.

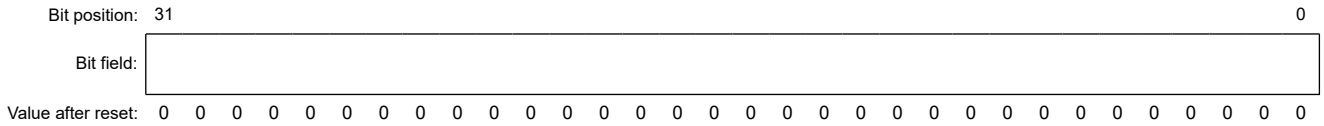
CRCSEN bit (Snoop Enable)

When the CRCSEN bit is set to 1, the CRC snoop operation is enabled. When this bit is set to 0, the CRC snoop operation is disabled.

41.2.3 CRCDIR/CRCDIR_BY : CRC Data Input Register

Base address: CRC = 0x4031_0000
CRC_NS = 0x5031_0000

Offset address: 0x04



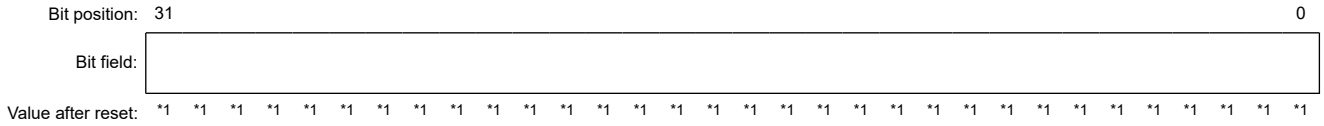
Bit	Symbol	Function	R/W
31:0	n/a	CRC input data The CRCDIR register is a 32-bit read/write register to write data for CRC-32 or CRC-32C calculation. The CRCDIR_BY (CRCDIR[31:24]) is an 8-bit read/write register to write data for CRC-8, CRC-16, or CRC-CCITT calculation.	R/W

Note: S-TYPE3, P-TYPE3

41.2.4 CRCDOR/CRCDOR_HA/CRCDOR_BY : CRC Data Output Register

Base address: CRC = 0x4031_0000
CRC_NS = 0x5031_0000

Offset address: 0x08



Bit	Symbol	Function	R/W
31:0	n/a	CRC output data The CRCDOR register is a 32-bit read/write register for CRC-32 or CRC-32C calculation. The CRCDOR_HA (CRCDOR[31:16]) register is a 16-bit read/write register for CRC-16 or CRC-CCITT calculation. The CRCDOR_BY (CRCDOR[31:24]) register is an 8-bit read/write register for CRC-8 calculation. Rewrite the CRCDOR/CRCDOR_HA/CRCDOR_BY register to perform the calculations using a value other than the initial value. Data written to the CRCDIR/CRCDIR_BY register is CRC calculated and the result is stored in the CRCDOR/CRCDOR_HA/CRCDOR_BY register. If the CRC code is calculated following the transferred data and the result is 0x00000000, there is no CRC error.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. If FSBL is executed in reset, then value after reset is 1, if FSBL is skipped in reset, then values after reset is 0.

41.2.5 CRCSAR : Snoop Address Register

Base address: CRC = 0x4031_0000
CRC_NS = 0x5031_0000

Offset address: 0x0C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CRCSA[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	CRCSA[13:0]	Register Snoop Address These bits store the TDR or RDR address in the SCI module to snoop	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

CRCSA[13:0] bits (Register Snoop Address)

The CRCSA[13:0] bits specify the lower address 14 bits of the register monitored by the CRC snoop operation.

Only the following addresses can be used for the CRCSA[13:0] bits:

[secure region address]

- 0x4035_8004: SCI0.TDR, 0x4035_8000: SCI0.RDR
- 0x4035_8104: SCI1.TDR, 0x4035_8100: SCI1.RDR
- 0x4035_8204: SCI2.TDR, 0x4035_8200: SCI2.RDR
- 0x4035_8304: SCI3.TDR, 0x4035_8300: SCI3.RDR
- 0x4035_8404: SCI4.TDR, 0x4035_8400: SCI4.RDR
- 0x4035_8904: SCI9.TDR, 0x4035_8900: SCI9.RDR

[non-secure region address]

- 0x5035_8004: SCI0.TDR, 0x5035_8000: SCI0.RDR
- 0x5035_8104: SCI1.TDR, 0x5035_8100: SCI1.RDR
- 0x5035_8204: SCI2.TDR, 0x5035_8200: SCI2.RDR
- 0x5035_8304: SCI3.TDR, 0x5035_8300: SCI3.RDR
- 0x5035_8404: SCI4.TDR, 0x5035_8400: SCI4.RDR
- 0x5035_8904: SCI9.TDR, 0x5035_8900: SCI9.RDR

41.3 Operation

41.3.1 Basic Operation

The CRC calculator generates CRC codes for use in LSB-first or MSB-first transfer.

The following examples show CRC code generation for input data (0xF0) using the 16-bit CRC-CCITT generating polynomial ($X^{16} + X^{12} + X^5 + 1$). In these examples, the value of the CRC Data Output Register (CRCDOR_HA) is cleared before CRC calculation.

When an 8-bit CRC (with the polynomial $X^8 + X^2 + X + 1$) is in use, the valid bits of the CRC code are obtained in CRCDOR_BY. When a 32-bit CRC is in use, the valid bits of the CRC code are obtained in CRCDOR.

Figure 41.2 and Figure 41.3 show the LSB-first and MSB-first data transmission examples respectively. Figure 41.4 and Figure 41.5 show the LSB-first and MSB-first data reception examples.

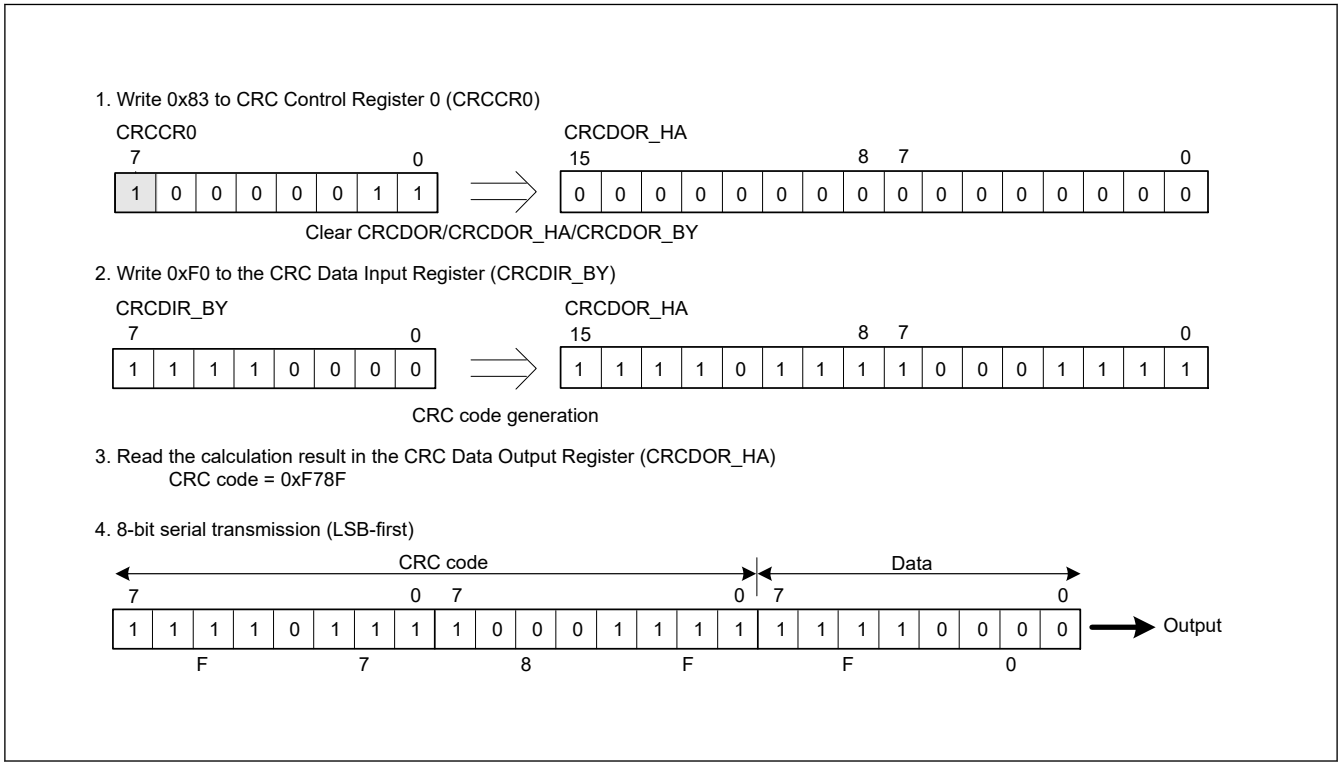


Figure 41.2 LSB-first data transmission

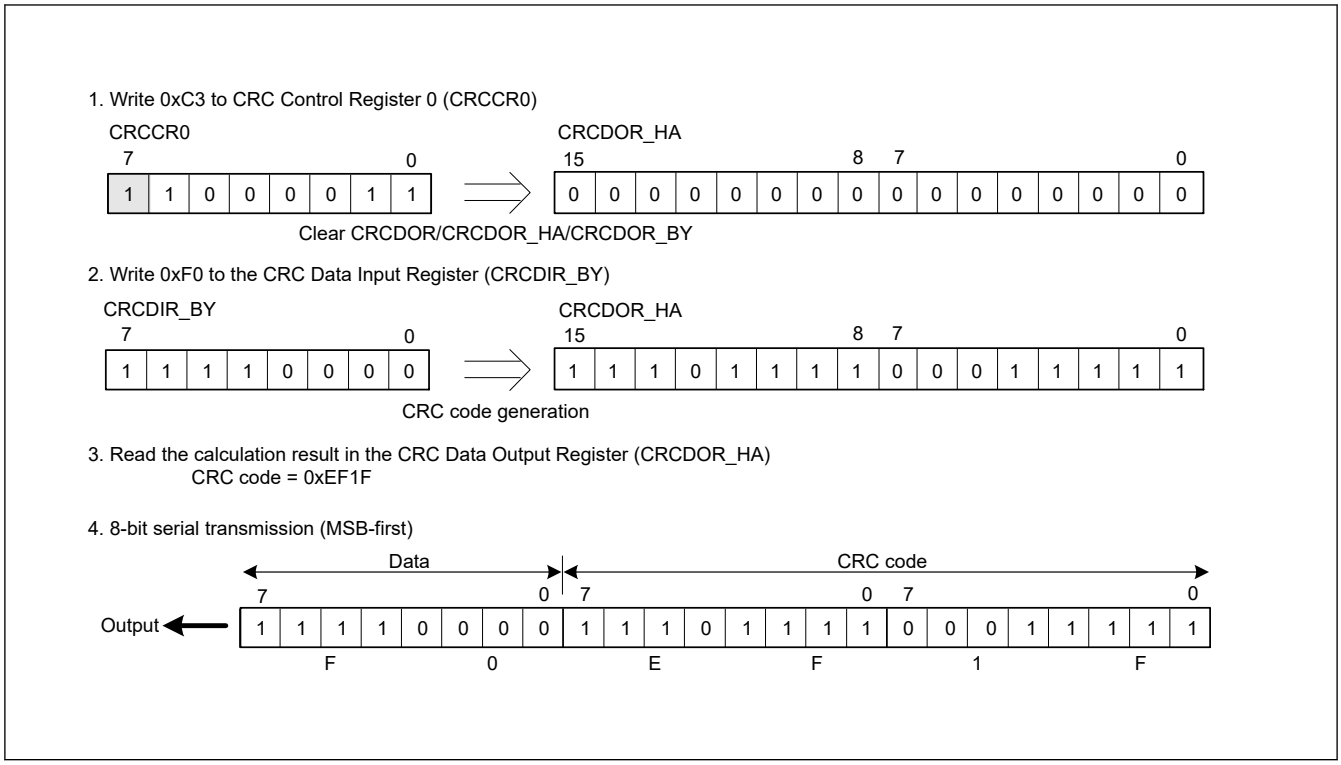


Figure 41.3 MSB-first data transmission

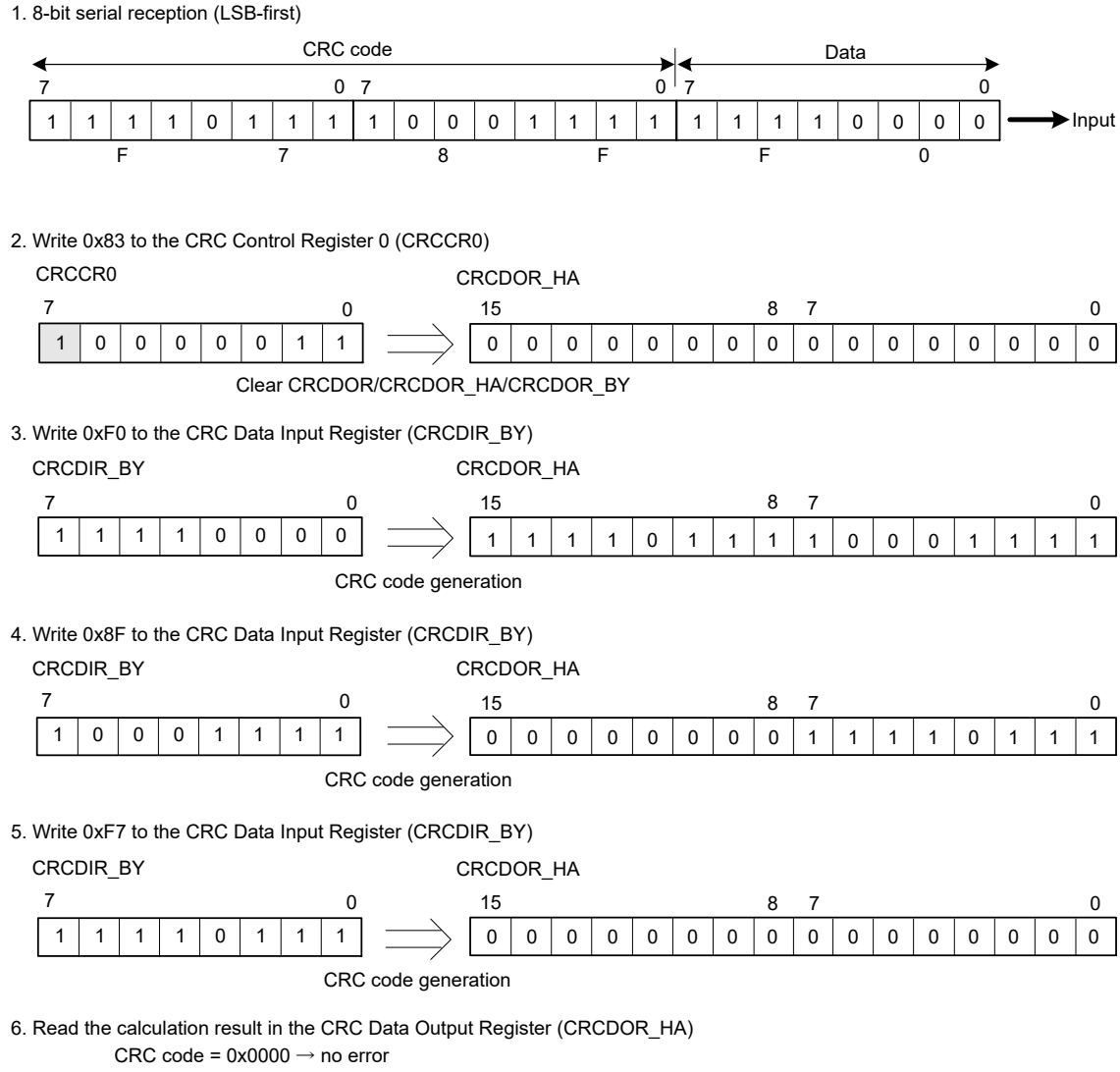


Figure 41.4 LSB-first data reception

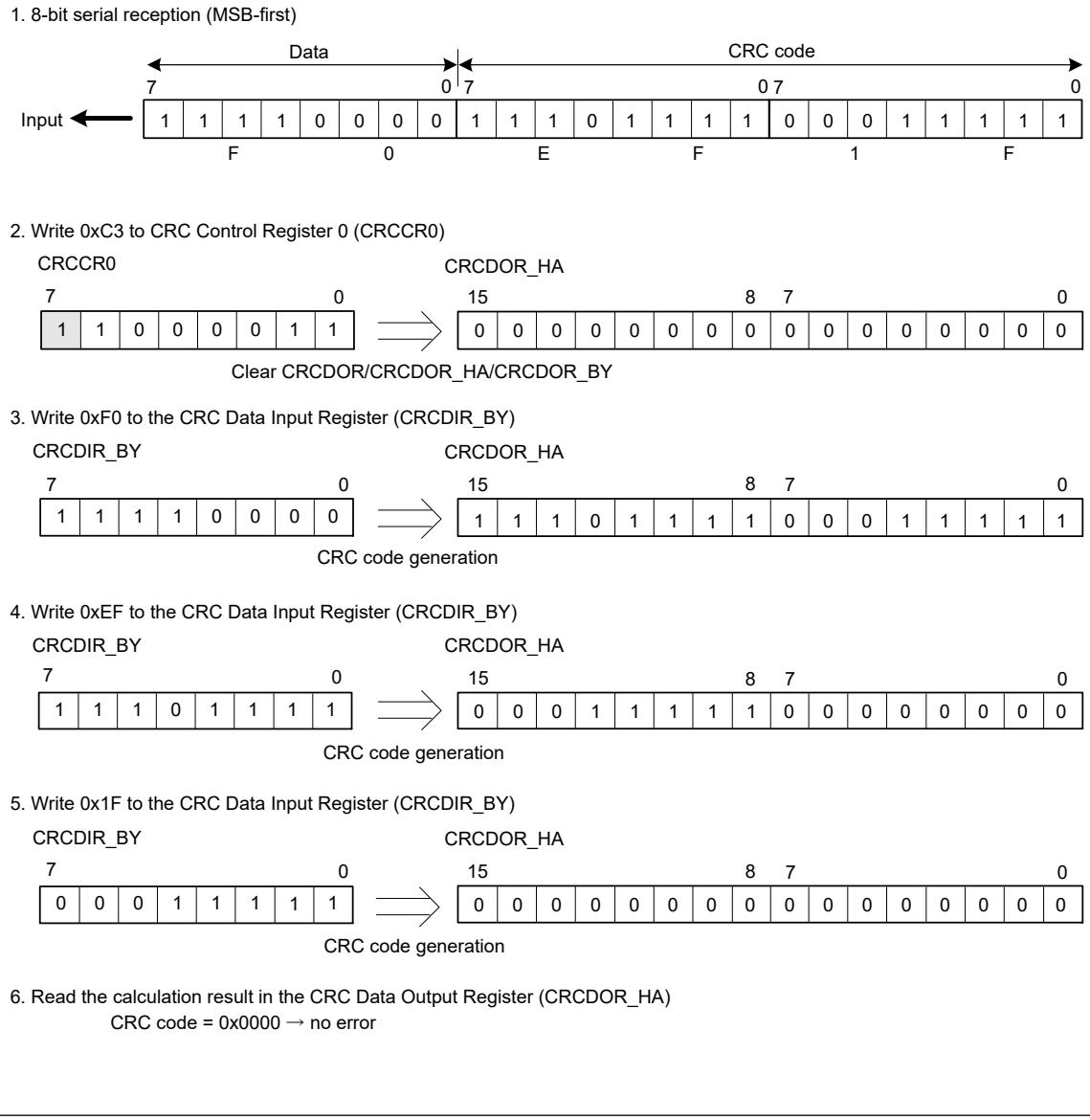


Figure 41.5 MSB-first data reception

41.3.2 CRC Snoop Function

The CRC snoop function monitors reads from and writes to a specific register and performs CRC calculation on the monitored data automatically. Because the CRC snoop function recognizes writes to and reads from a specific register address as a trigger to automatically perform CRC calculation, there is no need to write data to the CRCDIR register. All I/O register specified in the section 41.2.5. CRCSAR : Snoop Address Register are subject to the CRC snoop. The CRC snoop is useful in monitoring writes to the SCIn.TDR (n = 0 to 4, 9) register, and reads from the SCIn.RDR (n = 0 to 4, 9) register.

To use this function, write the lower address 14 bits of a specific register to bits CRCSA13 to CRCSA0 in the CRCSAR register, and set CRCSEN bit in the CRCCR1 register to 1. Then, set the CRCSWR bit in the CRCCR1 register to 1 to enable snooping on writes to the target register, or set the CRCSWR bit in the CRCCR1 register to 0 to enable snooping on reads from the target register. It is possible that access to a target I/O register may be executed before the CRCSWR bit write completed. In this case, the data is not stored in the CRCDIR register. To avoid this issue, before accessing I/O register, read back the CRCSWR bit that was written to confirm that the write completed.

When both the CRCSEN and CRCSWR bits are set to 1, and data is written to a target register in a bus master module such as the CPU, DMAC, and DTC, the CRC calculator stores the data in the CRCDIR register and performs CRC calculation. Similarly, when the CRCSEN bit is set to 1, CRCSWR bit to 0, and data is read from a target register in a bus master

module such as the CPU, DMAC, and DTC, the CRC calculator stores the data in the CRCDIR register and performs CRC calculations.

When the CRC code is generated by using CRC-8, CRC-16, and CRC-CCITT generating polynomial, the target register is accessed in 1 byte (8 bits). RDR_BY and TDR_LL should be used to access RDR and TDR. Similarly, when the CRC code is generated by using CRC-32 and CRC-32C generating polynomial, the target register is accessed in words (32 bits). Note that for RDR and TDR, CRC codes are generated that contain data other than RDAT and TDAT.

When CRC is marked as secure by PSARC.PSARC1 bit, the CRC snoop function is available for secure access to the specified I/O registers. When CRC is marked as non-secure by PSARC.PSARC1 bit, the CRC snoop function is available for non-secure access to the specified I/O registers.

41.4 Usage Notes

41.4.1 Settings for the Module-Stop State

The Module Stop Control Register C (MSTPCRC) can enable or disable CRC calculator operation. The CRC calculator is initially stopped after a reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

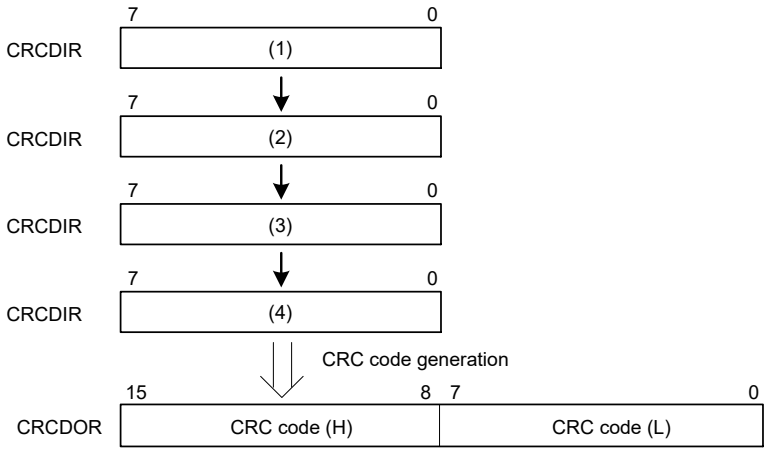
41.4.2 Note on Transmission

The transmission sequence for the CRC code differs based on whether the transmission is LSB-first or MSB-first. Figure 41.6 shows an LSB-first and MSB-first data transmission.

When transmitting 32-bit data (for operation executed on 8 bits in parallel)

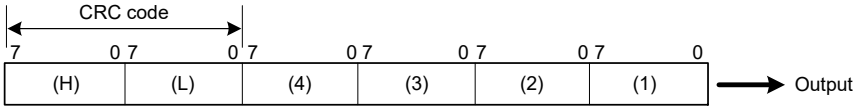
1. CRC code

After specifying the method for generation calculation, write data to CRCDIR in order of (1), (2), (3), and (4).



2. Transmit data

(i) When transmission is LSB-first



(ii) When transmission is MSB-first

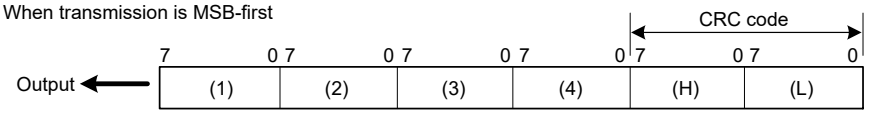


Figure 41.6 LSB-first and MSB-first data transmission

42. Boundary Scan

42.1 Overview

The boundary scan function provides a serial I/O interface based on the JTAG (Joint Test Action Group), IEEE Std.1149.1, and IEEE Standard Test Access Port and Boundary Scan Architecture. Table 42.1 lists the boundary scan specifications, Figure 42.1 shows a block diagram, and Table 42.2 lists the I/O pins.

Table 42.1 Boundary scan specifications

Parameter	Specifications
Execution condition	Boundary scan must be executed when the RES pin is driven low.
Test modes	• BYPASS mode • EXTEST mode • SAMPLE/PRELOAD mode • CLAMP mode • HIGHZ mode • IDCODE mode

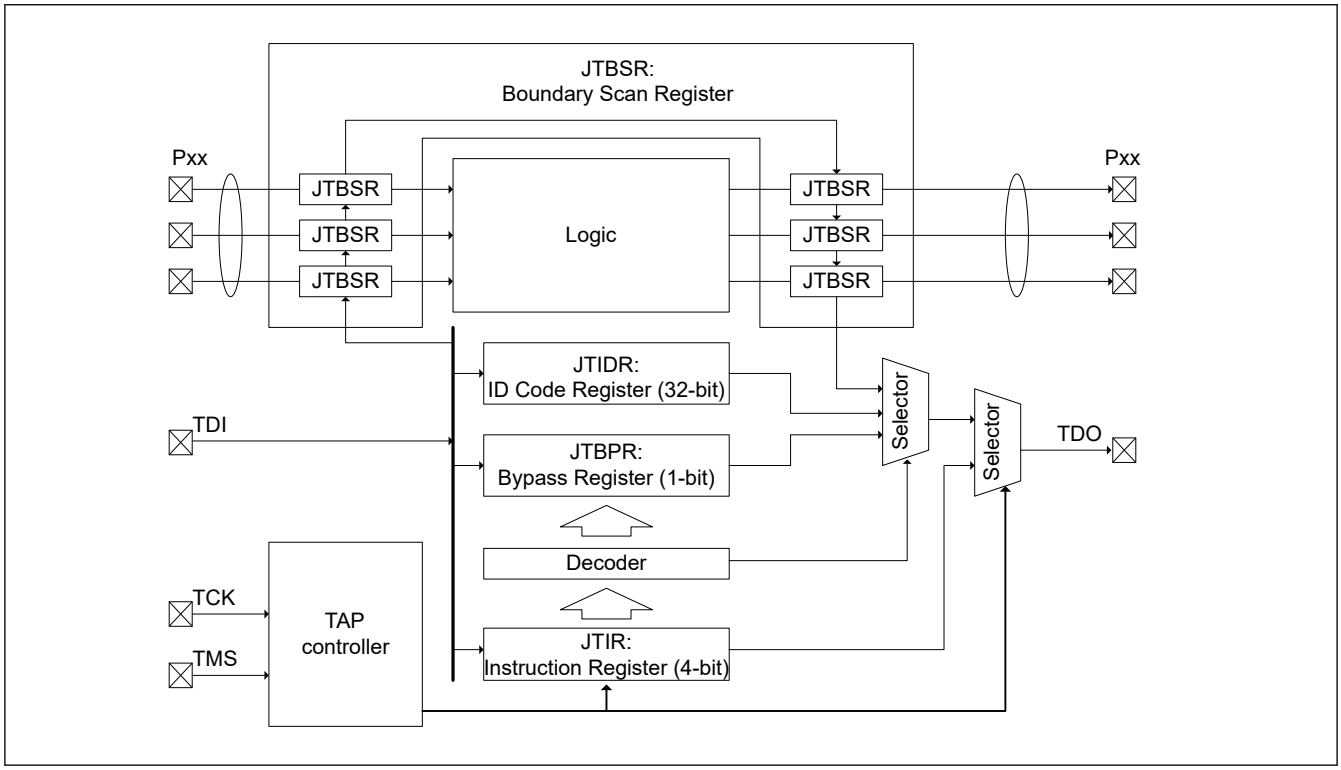


Figure 42.1 Boundary scan function block diagram

Table 42.2 Boundary scan I/O pins

Pin name	I/O	Description
TCK	Input	Test clock input pin Clock signal for boundary scan. The input clock duty cycle is 50% when the boundary scan function is used.
TMS	Input	Test mode select pin
TDI	Input	Test data input pin
TDO	Output	Test data output pin

Note: This device does not support the TRST pin for the JTAG interface.

42.2 Register Descriptions

Table 42.3 lists the boundary scan registers.

Table 42.3 Boundary scan registers

Register name	Symbol	Value after reset
Instruction Register	JTIR	0xE
ID Code Register	JTIDR	0x0841_9447
Bypass Register	JTBPR	Undefined
Boundary Scan Register	JTBSR	Undefined

Usage notes for the boundary scan registers:

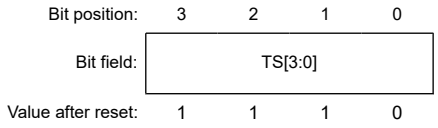
- Instructions can be input to the Instruction Register (JTIR) through the TDI pin by serial transfer.
- The Bypass Register (JTBPR), which is a 1-bit register, is connected between the TDI and TDO pins in BYPASS mode.
- The Boundary Scan Register (JTBSR), which is configured according to the BSDL description, is connected between the TDI and TDO pins when test data is being shifted in.

Table 42.4 shows the availability of serial transfer for the registers.

Table 42.4 Serial transfer for registers

Register name	Serial input	Serial output
Instruction Register (JTIR)	Available	Available
ID Code Register (JTIDR)	Available	Available
Bypass Register (JTBPR)	Available	Available
Boundary Scan Register (JTBSR)	Available	Available

42.2.1 JTIR : Instruction Register

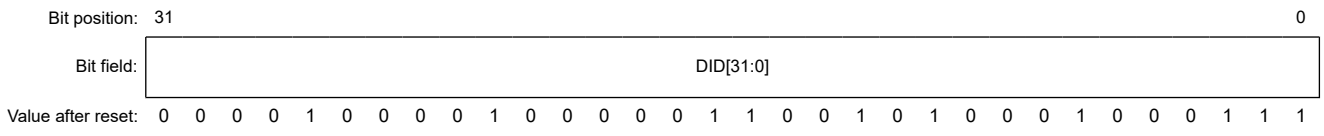


Bit	Symbol	Function	R/W
3:0	TS[3:0]	Test Bit Set The command configuration for these bits	—

TS[3:0]	Instruction
0x0	EXTEST
0x1	SAMPLE/PRELOAD
0x3	IDCODE (Renesas code)
0x5	CLAMP
0x6	HIGHZ
0xF	BYPASS
Others	Reserved

JTAG instructions can be transferred to the JTIR register by serial input from the TDI pin. The JTIR register is initialized when a power-on reset occurs, or when the TAP controller is in the Test-Logic-Reset state.

42.2.2 JTIDR : ID Code Register



Bit	Symbol	Function	R/W
31:0	DID[31:0]	Device ID These bits store the fixed value that indicates the device IDCODE (0x0841_9447).	—

The JTIDR register data is output from the TDO pin when the IDCODE instruction is executed. After a reset release, the DID[31:0] of JTIDR changes into the Arm® debug code. See the *Arm® CoreSight™ SoC-400 Technical Reference Manual* (ARM DDI 0480F).

42.2.3 JTBPR : Bypass Register

The JTBPR register is a 1-bit register and is connected between the TDI and TDO pins when the JTIR register is set to BYPASS mode. The JTBPR register cannot be read from or written to by the CPU.

42.2.4 JTBSR : Boundary Scan Register

The JTBSR register is a shift register for controlling the external input and output pins of this device, and is distributed across the pads. To apply the JTBSR register in boundary-scan testing, issue the EXTEST, SAMPLE/PRELOAD, CLAMP and HIGHZ instructions. The BSDL file describes the associations between the JTBSR register bits and the pins of this device. The value after reset is undefined.

42.3 Operation

During a reset, the JTAG ports, TCK, TMS, TDI and TDO are assigned as default pin functions. The TCK, TMS and TDI pins are pulled up by the pull-up resistors. Boundary scan testing can be executed after the setup time elapses when POR is negated and RES is driven low.

42.3.1 TAP Controller

Figure 42.2 shows the state transition diagram of the TAP controller. All transitions are controlled by the TMS signal.

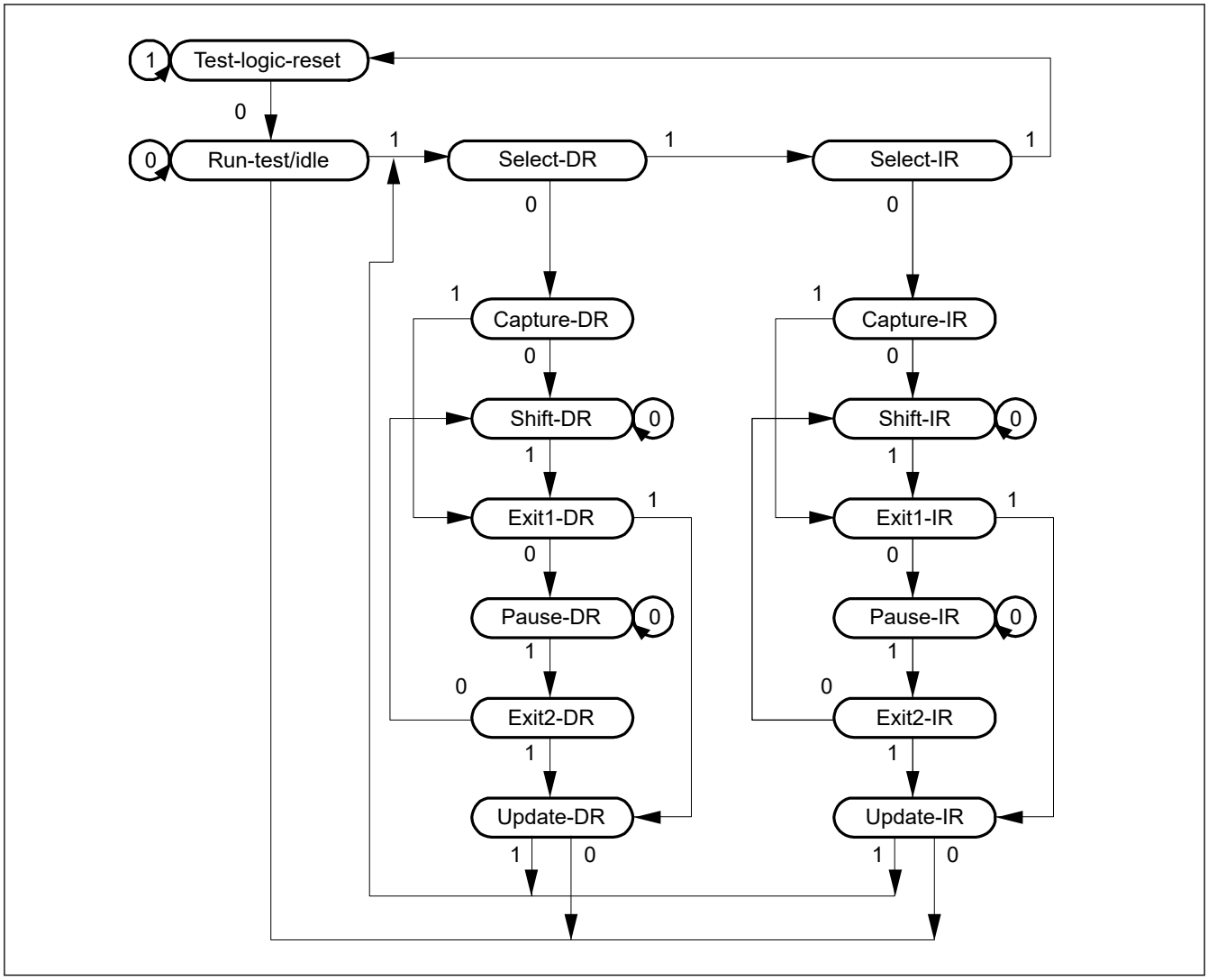


Figure 42.2 State transition diagram of TAP controller

42.3.2 Commands

(1) BYPASS

The BYPASS instruction drives the Bypass Register (JTBPR). This instruction shortens the shift path, facilitating the transfer of serial data to other LSIs on a printed circuit board at higher speeds. While this instruction is being executed, the test circuit has no effect on the system circuits.

The JTBPR register is connected between the TDI and TDO pins. Bypass operation is initiated from the Shift-DR operation. The TDO is low in the first clock cycle in the Shift-DR state. In the subsequent clock cycles, values input to the TDI pin are output from the TDO pin.

(2) EXTEST

The EXTEST instruction is used to test external circuits when this device is installed on the printed circuit board. If this instruction is executed, output pins are used to output test data (specified in the SAMPLE/PRELOAD instruction) from the Boundary Scan Register (JTBSR) to the other devices, and input pins are used to input the test result.

(3) SAMPLE/PRELOAD

The SAMPLE/PRELOAD instruction is used to input data from the internal circuits of this device to the JTBSR register, output data from the scan path, and reload the data to the scan path. While this instruction is executed, input signals are directly input to this device and output signals are also directly output to the external circuits. This device system circuit is not affected by this instruction.

In SAMPLE operation, the JTBSR register latches a snapshot of the data transferred from the input pins to the internal circuit or data transferred from the internal circuit to the output pins. The latched data is read from the scan path. The JTBSR register latches the data snapshot on the rising edge of the TCK pin in the Capture-DR state. The data snapshot is only transferred from the internal circuit to the output pins during a reset.

In PRELOAD operation, the initial value is written from the scan path to the parallel output latch of the JTBSR register prior to the EXTEST instruction execution. If EXTEST is executed without executing this PRELOAD operation, undefined values are output from the beginning to the end (transfer to the output latch) of the EXTEST sequence. (In the EXTEST instruction, output parallel latches are always output to the output pins.)

(4) IDCODE

When the IDCODE instruction is selected, the ID Code Register (JTIDR) value is output to the TDO pin in the Shift-DR state of the TAP controller. In this case, the JTIDR register value is output LSB-first. During this instruction execution, the test circuit does not affect the system circuit.

(5) CLAMP

When the CLAMP instruction is selected, output pins output the JTBSR register value that was specified in the SAMPLE/PRELOAD instruction in advance. While the CLAMP instruction is selected, the status of the JTBSR register is maintained regardless of the TAP controller state.

The JTBPR register is connected between the TDI and TDO pins, leading to the same operation as when the BYPASS instruction is selected.

(6) HIGHZ

When the HIGHZ instruction is selected, all output pins enter high-impedance state. While the HIGHZ instruction is selected, the JTBSR register is maintained regardless of the state of the TAP controller.

The JTBPR register is connected between the TDI and TDO pins, leading to the same operation as when the BYPASS instruction is selected.

42.4 Usage Notes

The boundary scan function is subject to the following constraints:

- The boundary scan must be executed when the RES pin is driven low
- Serial data input/output is in LSB order, as shown in Figure 42.3

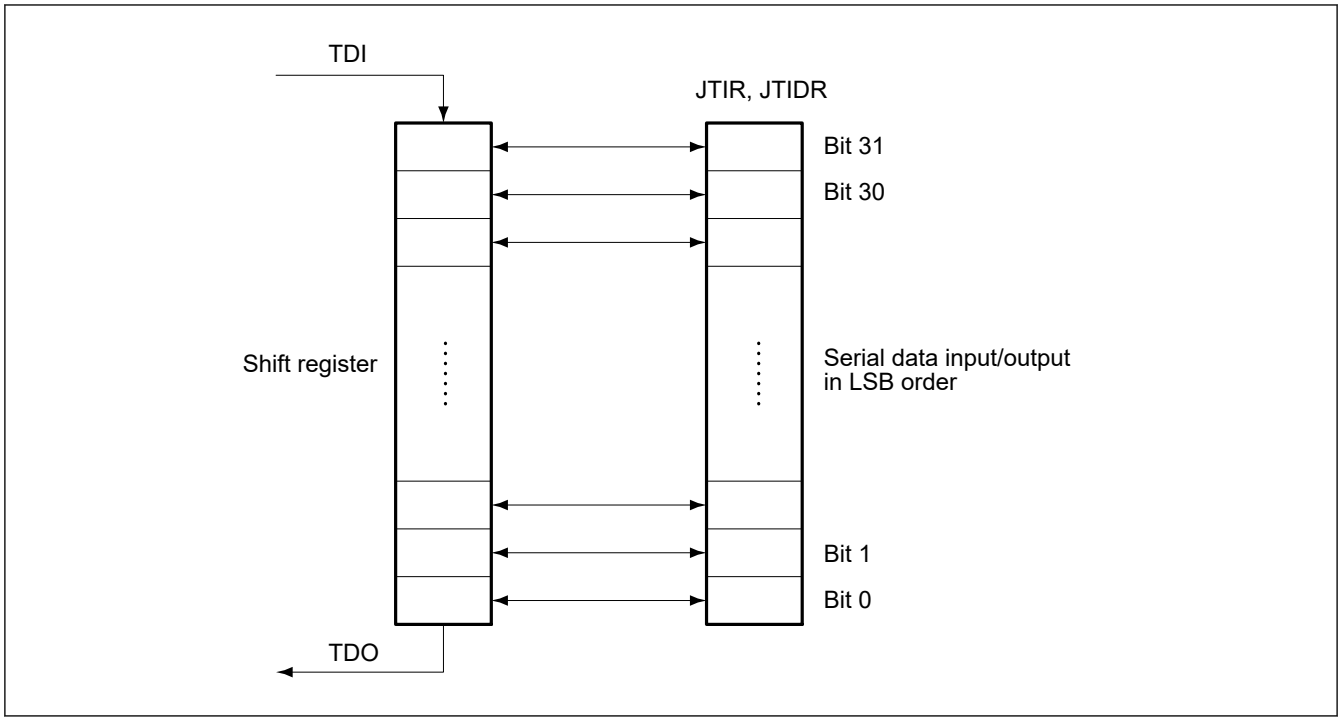


Figure 42.3 Serial data input/output

The following pins cannot be boundary-scanned:

- Power supply pins (VCC, VCC2, VCC_DCDC, VCL, VSS, VSS_DCDC, VBATT, AVCC0, AVSS0, VCC_USB, VSS_USB, VSS1_USBHS, VSS2_USBHS, AVCC_MIPI, VCC18_MIPI and VSS_MIPI)
- Analog reference pins (VREFH0, VREFL0, VREFH, VREFL)
- Clock pins (EXTAL, XTAL, XCIN, and XCOUT)
- Reset pin (RES)
- USBHS-dedicated pins (USBHS_DP, USBHS_DM)
- USBFS pin (USB_DP, USB_DM)
- The boundary-scan pins (TCK, TMS, TDI, and TDO).
- MIPI-dedicated pins (MIPI_CL_N, MIPI_CL_P, MIPI_DL0_N, MIPI_DL0_P, MIPI_DL1_N, MIPI_DL1_P)
- The switching regulator pin (VLO)

43. Security Features

43.1 Features

- Armv8-M TrustZone Technology
  - Implementation Defined Attribution Unit (IDAU) is implemented
  - Security Attribution Unit (SAU) is implemented
    - 8 regions
  - Master Security Attribution Unit (MSAU) is implemented (IDAU for master other than CPU)
  - Code flash
    - Up to 2 regions (secure/non-secure) in linear mode
    - Up to 4 regions (2 secure regions/2 non-secure regions) in dual mode
  - Data flash
    - Up to 2 regions (secure/non-secure)
  - SRAM
    - Up to 2 regions (secure/non-secure) per SRAM
  - Standby SRAM
    - Up to 2 regions (secure/non-secure)
  - VBATT backup registers
    - Up to 2 regions (secure/non-secure)
  - Peripheral
    - Security attributes can be set individually for each unit/channel
  - CSC[*1]
    - CSC region is defined as non-secure
  - SDRAM[*1]
    - SDRAM region is defined as non-secure
  - OSPI[*1]
    - OSPI region is defined as non-secure
- Privileged control
  - Access permissions for memory except Standby SRAM and VBATT backup register are controlled by MPU managed by the privileged code
  - Privilege attributes of Standby SRAM and VBATT backup register are controlled by the registers of each controller
  - Individual privileged or Unprivileged attribution for each peripheral.
- Device lifecycle management
- Three debug levels
  - AL2: Non-secure and secure debug functions are enabled and accessible from the debugger
  - AL1: Only non-secure debug functions is enabled, and the debugger can access only defined non-secure debug accessible regions
  - AL0: No debug functions are available.
- Key injection
- Secure factory programming
  - Supports image programming in ciphertext format in an untrusted factory
- Secure boot

  - Verify the integrity and authenticity of image to be programmed
  - Verify the integrity and authenticity of the executable image by the immutable (ROM) First Stage Bootloader (FSBL) prior to executing the image.
- Cryptographic accelerator
  - See section 44, Renesas Secure IP (RSIP-E51A).
- Secure pin multiplexing
  - All I/O port pins can be configured individually as secure or non-secure
  - Peripheral pin function is valid when the security attribute of peripheral and I/O port match. See section 19, I/O Ports.

Note 1. There is no TrustZone filter in the external RAM and external device area. Therefore, even if these regions are marked as secure in the SAU settings, non-secure masters except the CPU can access them.

43.2 Tamper Detection

The device is provided with I/O ports capable of detecting an external tamper attempt. In the case of a tamper event:

- Timestamping
  The RTC can take a timestamp to log the event, and an interrupt can be generated. See section 24, Realtime Clock (RTC).
- VBATT backup registers
  VBATT backup registers are zeroized (set to zero) following detection of a tamper event. Seesection 11, Battery Backup Function.

43.3 Arm Security Features

43.3.1 Arm TrustZone Technology

Arm TrustZone technology divides the system and the application into secure and non-secure domains. A secure application can issue both secure and non-secure transactions, but a non-secure application can only issue non-secure transactions. Secure transactions can only access secure memory and resources, and non-secure transactions can only access non-secure memory and resources. Secure transactions can be issued only using secure region addresses and non-secure transactions can be issued only using non-secure region addresses. Figure 43.1shows the security attributes of transactions that can be issued by each master.

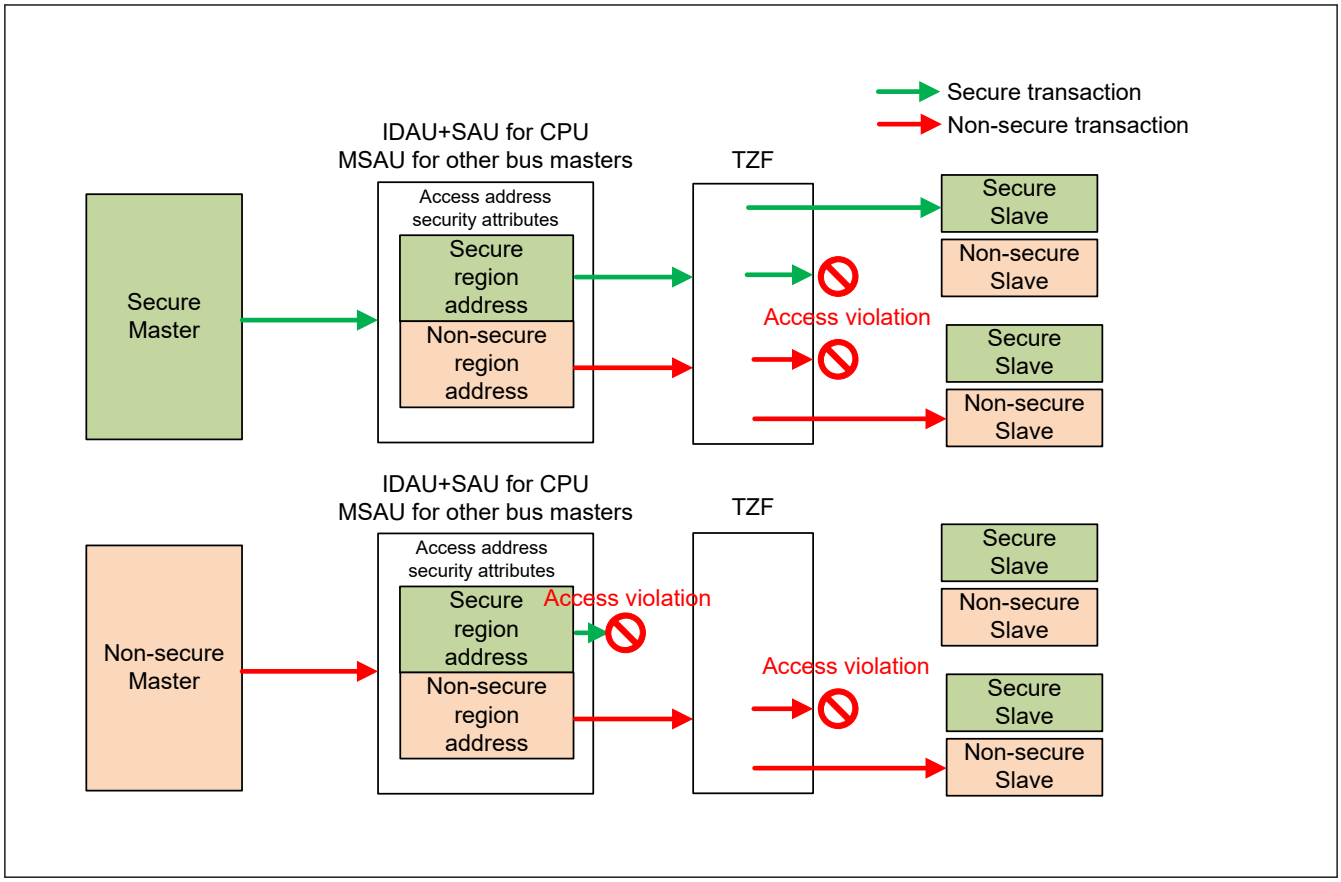


Figure 43.1 Transactions that can be issued by each master

For more details, see the *Arm® v8-M Architecture Reference Manual and Arm® Platform Security Architecture Trusted Base System Architecture for Arm®v8-M 1.0*.

43.3.2 Privileged Control

Systems and applications are divided into privileged and unprivileged domains. The CPU can limit or exclude access to some resources by executing code in privileged or unprivileged mode. Privileged mode can access both privileged and unprivileged domains, but unprivileged mode can access only the unprivileged domain.

For more details, see the *Arm® v8-M Architecture Reference Manual and Arm® Platform Security Architecture Trusted Base System Architecture for Arm®v8-M 1.0*.

43.3.3 Security Attribution

The TrustZone for Armv8.1-M implementation consists of the Security Attribution Unit (SAU) and Implementation Defined Attribution Unit (IDAU). The 4GB memory space is partitioned into Secure (S) and Non-secure (NS) memory regions. The Secure memory space is further divided into two types, Non-secure Callable (NSC) and Secure.

Note:
- S = Secure addresses are used for memory and peripherals that are only accessible by secure software or secure masters.
- NSC = A special type of secure location. This type of memory is the only type in which an Armv8.1-M processor permits to hold a Secure Gateway (SG) instruction that enables software to transition from Non-secure to Secure state.
- NS = Non-secure addresses are used for memory and peripherals accessible by all software running on the device.

43.3.3.1 Implementation Defined Attribution Unit (IDAU)

The IDAU defines the code, SRAM and peripheral region into the secure alias region and the non-secure alias region by the address bit [28]. The secure code region and secure SRAM region are assigned the NSC security attributes. The security map defined by IDAU is fixed in hardware and cannot be changed by software.

43.3.3.2 Master Security Attribution Unit (MSAU)

MSAU is IDAU that defines system-specific security address map for masters other than CPU. The MSAU defines secure and non-secure alias regions but does not define the Non-secure Callable (NSC) and region number. Masters other than the CPU can issue a security transaction using the secure alias address defined by MSAU. However, non-secure masters are prohibited from issuing secure transactions using an address in the secure alias region. Figure 43.2 shows the defined security map of IDAU and MSAU. The security map defined by MSAU is fixed in hardware and cannot be changed by software.

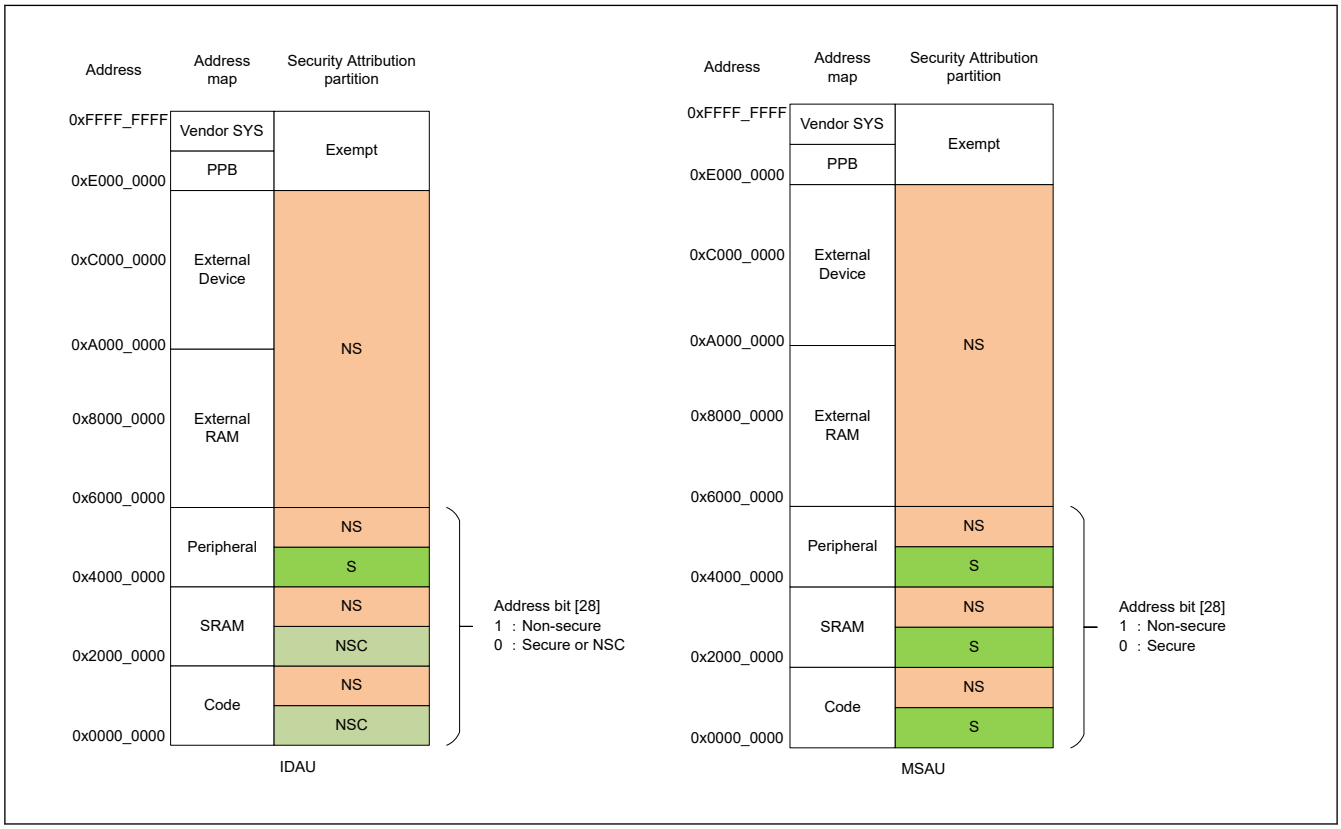


Figure 43.2 IDAU and MSAU defined security map

43.3.3.3 Secure Attribution Unit (SAU)

The SAU is a programmable unit that determines the security of an address. The SAU is programmable in the Secure state and has a programmers' model similar to the MPU. If an address maps to regions defined by both IDAU and SAU, the region of the highest security level is selected. Secure master can issue secure and non-secure transactions using the address of each security alias region. Non-secure master cannot issue secure transactions using the address of the secure alias region. Figure 43.3 shows the final determination of the address map security level.

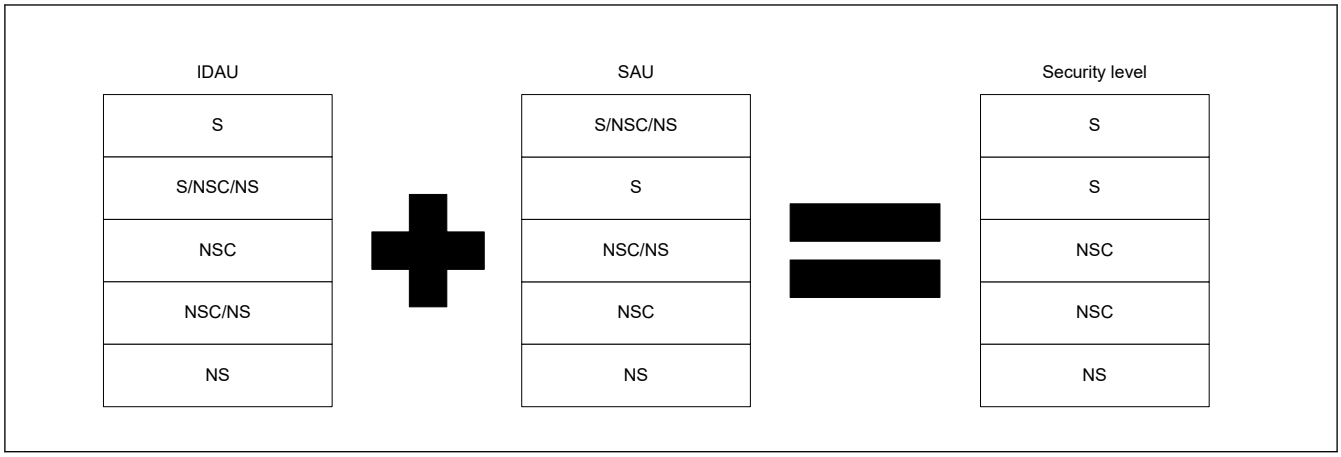


Figure 43.3 Final determination of address map security level

When using TrustZone to perform secure and non-secure region separation, SAU MUST be set according to the following.

The regions set as NS attribute in IDAU MUST be set to NS in the SAU as well. The regions set to NS attribute in IDAU are:

0x1000_0000 to 0x1FFF_FFFF (SAU Region 1 in Figure 43.4)

0x3000_0000 to 0x3FFF_FFFF (SAU Region 3 in Figure 43.4)

0x5000_0000 to 0xDFFF_FFFF (SAU Region 3 in Figure 43.4)

At least one NSC region MUST be created within any region defined as NSC by the IDAU. The regions set to NSC attribute by the IDAU are:

0x0000_0000 to 0x0FFF_FFFF (SAU Region 0 in Figure 43.4)

0x2000_0000 to 0x2FFF_FFFF (SAU Region 2 in Figure 43.4)

If you do not wish to define any isolation by means of using TrustZone, do not change SAU_CTRL.ALLNS = 0 and SAU_CTRL.ENABLE = 0 (initial value).

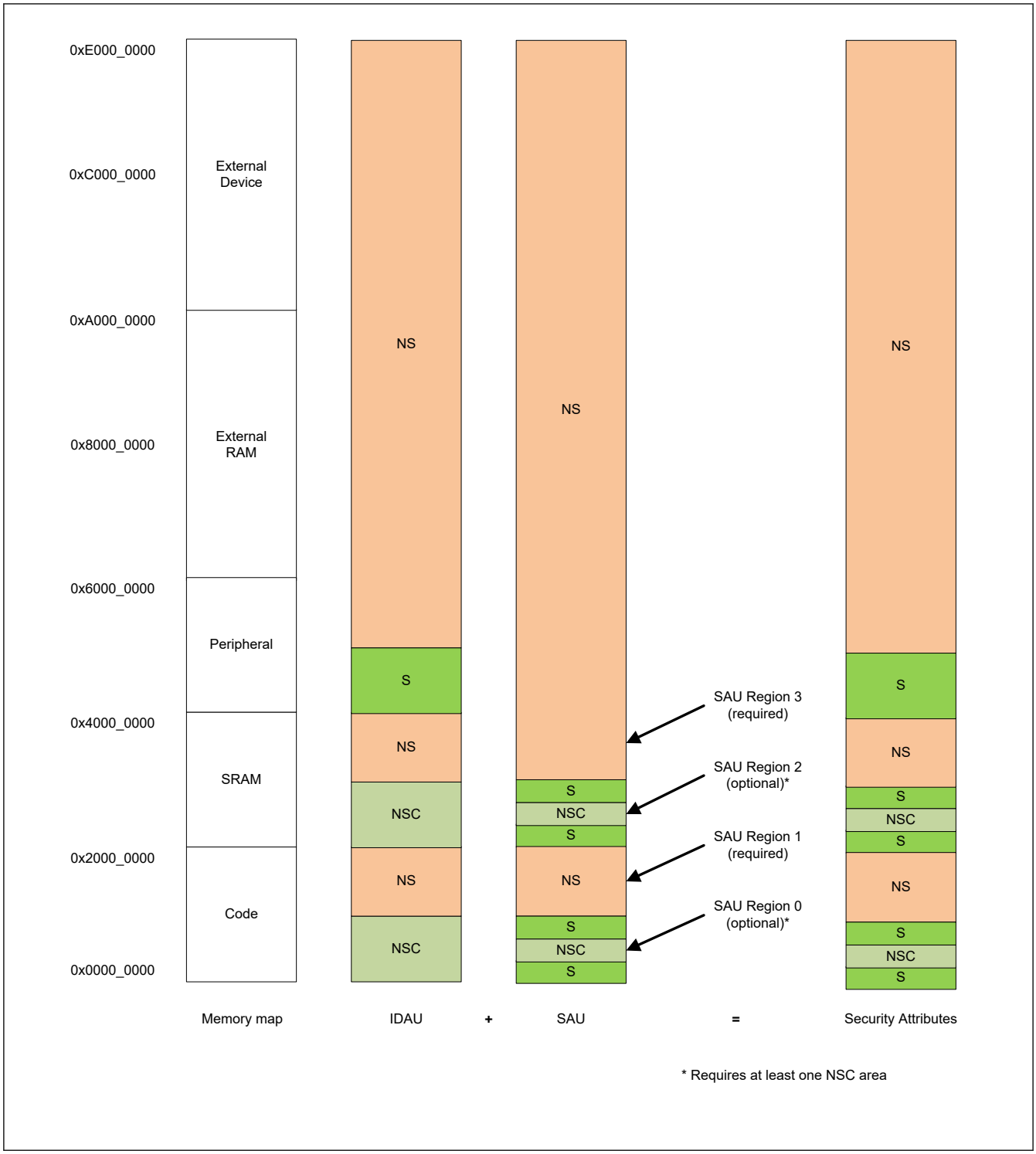


Figure 43.4 SAU settings and resulting security attributes

43.3.3.4 Region Number

The SAU and IDAU also define region numbers for each of the memory regions and security attributes. This region number is used by software to determine if a contiguous range of memory shares common security attributes. Figure 43.5 shows the defined region number of IDAU.

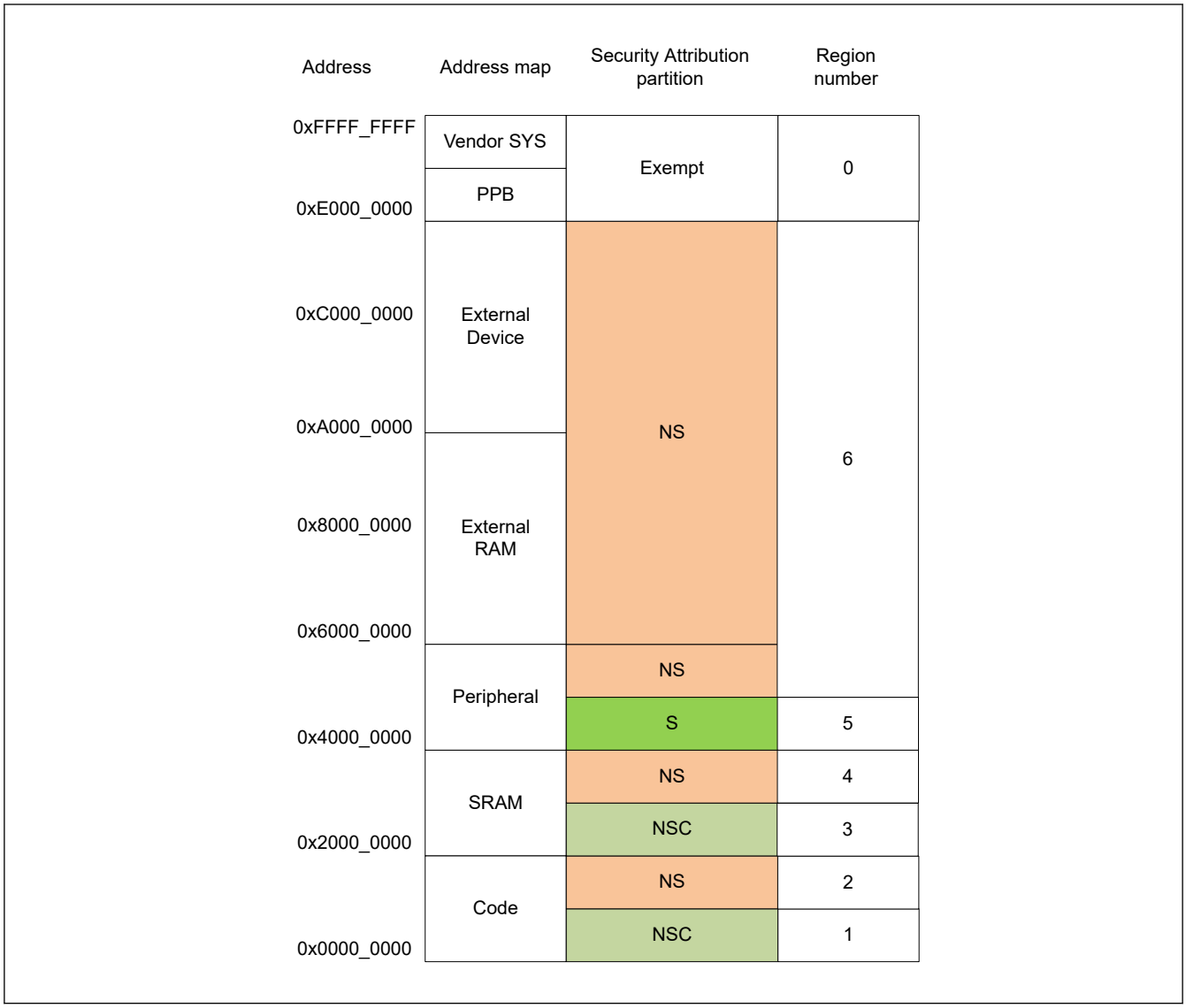


Figure 43.5 IDAU defined region numbers

43.3.3.5 Memory Security Attribution of TrustZone filter

The memories are divided into S and NS regions. The memory security attribution of code flash and data flash is stored into nonvolatile memory by a boot firmware command when the device lifecycle is in the OEM state and the authentication level is AL2. These memory security attributions are applied before application execution. They cannot be updated by the application but are readable using dedicated registers. The memory security attributions of SRAM, standby SRAM, and VBATT backup registers are set by a dedicated security attribution register writable only by secure access.

The code flash can be divided in up to two regions in linear mode and four regions in dual mode. The partitioning is the same between bank0 and bank1 in the dual mode. The data flash can be divided in up to two regions. SRAM, standby SRAM, and VBATT backup registers can be divided in up to two regions. Figure 43.6 shows the memory mapping. Table 43.1 shows size of the memory region and Table 43.2 shows access permission of the memory region.

It is prohibited to place Secure or Non-secure Callable regions in a block swappable area in linear mode because the secure application would be placed in the non-secure region after block swapping.

Bank swapping by the Non-secure user may change the code in the Secure or Non-secure callable areas. The following is recommended to prevent this.

- Set BANKSEL_SEL.BANKSWP[2:0] = 000b to disable the bank swapping for Non-secure users.
- Make the code in Secure or Non-secure callable areas for both banks the same when allow Non-secure user to swap banks.

See section 43.8. Field Updating in Dual Mode.

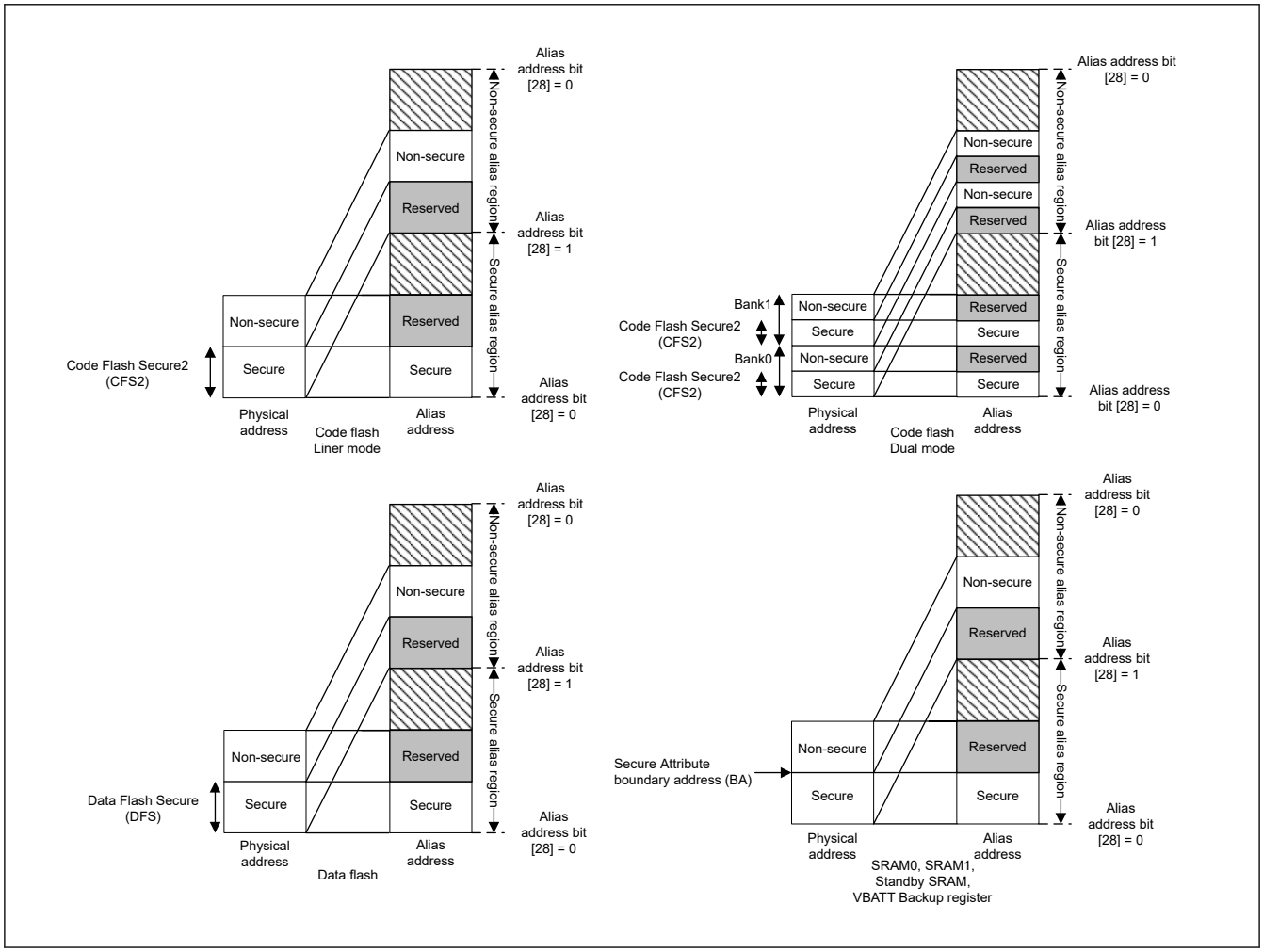


Figure 43.6 Memory mapping

Table 43.1 Size of memory region (1 of 2)

Memory region		Start address	Size
Linear mode	Code flash secure	0x0200_0000	CFS2 size (32 KB)
	Code flash non-secure	0x1200_0000 + CFS2 size (32 KB)	Code flash size - CFS2 size (32 KB)
Dual mode	Code flash bank0 secure	0x0200_0000	CFS2 size (32 KB)
	Code flash bank0 non-secure	0x1200_0000 + CFS2 size (32 KB)	Code flash size/2 - CFS2 size (32 KB)
	Code flash bank1 secure	0x0220_0000	CFS2 size (32 KB)
	Code flash bank1 non-secure	0x1220_0000 + CFS2 size (32 KB)	Code flash size/2 - CFS2 size (32 KB)
Data flash secure		0x2700_0000	DFS size (1 KB)
Data flash non-secure		0x3700_0000 + DFS size (1 KB)	Data flash size - DFS size (1 KB)
SRAM0 secure		0x2200_0000	BA (8 KB)
SRAM0 non-secure		0x3200_0000 + BA (8 KB)	SRAM0 size - BA (8 KB)
SRAM1 secure		0x2206_0000	BA (8 KB)
SRAM1 non-secure		0x3206_0000 + BA (8 KB)	SRAM1 size - BA (8 KB)
Standby SRAM secure		0x2600_0000	BA (128 bytes)
Standby SRAM non-secure		0x3600_0000 + BA (128 bytes)	Standby SRAM size - BA (128 bytes)
VBATT backup register secure		0x4001_ED00	BA (32 bytes)

Table 43.1 Size of memory region (2 of 2)

Memory region	Start address	Size
VBATT backup register non-secure	0x5001_ED00 + BA (32 bytes)	Backup register size - BA (32 bytes)

Note: The number in parentheses indicates the smallest unit that can be set by the user.
Note: BA is the setting value of the security attribution boundary address register for each memory region.

Table 43.2 Access permission of the memory region

Memory region	Secure transaction	Non-secure transaction
Each memory region configured as S or NSC	Allowed	Write ignored/Read ignored TrustZone access error is generated
Each memory region configured as NS	Write ignored/Read ignored TrustZone access error is generated	Allowed

43.3.3.6 Peripheral Security Attribution of TrustZone filter

Each peripheral can be configured as secure or non-secure and privileged or unprivileged.

Peripherals are divided into two types.

Type1 peripherals has one security and privilege attribution. Access to all registers is controlled by one security and privilege attribution. Type1 peripheral security and privilege attribution is set to the PSARx (x = B to E) and the PPARx (x = B to E) register by the secure and privileged application.

Type2 peripheral has the security attribution for each register or for each bit and access to each register or bit field is controlled according to these security attributions. Type2 peripheral security attribution is set to the Security Attribution register in each module by the secure application. For details on the Security Attribution register, see each section.

Table 43.3 shows the classification of peripheral type.

Table 43.3 Classification of peripheral type

Type1	Type2
SCI, SPI, OSPI, DOTF, ETHERC, EDMAC, USBHS, USBFS, IIC, I3C,RSIP-E51A, CANFD, CEU, DRW, GLCDC, MIPI-DSI, DOC, SDHI, SSIE, CRC, CAC, ACMPHS, TSN, ADC12, DAC12, POEG, AGT, GPT, ULPT, RTC, IWDT, and WDT	System control (Resets, PVD, Clock Generation Circuit, Low Power Modes, Battery Backup Function), Flash memory controller, Flash cache, SRAM controller, CPU cache, DMAC, DTC, ICU, MPU, BUS, Security setting, ELC, and I/O ports

Table 43.4 shows the access permission of type1 peripherals. The access permission of type2 peripherals is different by peripherals. Each register description of type2 peripherals includes the note of S-TYPE and P-TYPE indicating the specification of the access permission. For details on S-TYPE and P-TYPE, see section Appendix 4, Note for Register R/W.

Table 43.4 Access permission of type1 peripherals

	Secure and Privileged access	Secure and Unprivileged access	Non-secure and Privileged access	Non-secure and Unprivileged access
Peripheral configured as Secure and Privileged	Allowed	Write ignored/Read ignored TrustZone Access error is generated	Write ignored/Read ignored TrustZone Access error is generated	Write ignored/Read ignored TrustZone Access error is generated
Peripheral configured as Secure and Unprivileged	Allowed	Allowed	Write ignored/Read ignored TrustZone Access error is generated	Write ignored/Read ignored TrustZone Access error is generated
Peripheral configured as Non-secure and Privileged	Write ignored/Read ignored TrustZone Access error is generated	Write ignored/Read ignored TrustZone Access error is generated	Allowed	Write ignored/Read ignored TrustZone Access error is generated
Peripheral configured as Non-secure and Unprivileged	Write ignored/Read ignored TrustZone Access error is generated	Write ignored/Read ignored TrustZone Access error is generated	Allowed	Allowed

43.3.4 TrustZone Access Error

Table 43.5 shows the behavior when TrustZone access error occurs. The behavior varies depending on the master or the slave area to be accessed.

Table 43.5 Behavior of TrustZone access error

DAP	CPU	DMAC/DTC	EDMAC	DRW	MIPI	GLCDC/CEU
• Only error response is returned*1	• IDAU/SAU detects SecureFault exception • TrustZone filter detects BusFault exception • Can issue a reset*2	• Stop transfer • Issue NMI or reset*2 • Generate interrupt (DMA0_TRANSERR)	• Issue NMI or reset*2 • Generate interrupt (ETHER_EINT0)*3	• Stop next transfer • Issue NMI or reset*2 • Generate interrupt (DRW_IRQ)*4	• Stop next transfer • Issue NMI or reset*2 • Generate interrupt (DSI_SEQ0, DSI_SEQ1)*5	• Issue NMI or reset*2

Note: This behavior is not applicable for bufferable write access. For more information on when a bufferable write access error is detected, see section 14.7.2. Operations When a Bus Error Occurs.
Note 1. When a TrustZone access error occurs by a debugger access, exception, NMI, or reset does not occur. Only the error response is returned.
Note 2. The operation after error detection is selected by the BUSOAD.SRERROAD bit.
Note 3. Address Error flag in EESR.ADE bit is set. The interrupt occurs when it is enabled in EESIPR.ADEIP bit.
Note 4. Access error flag in STATUS.BUSIRQ is set. The interrupt occurs when it is enabled in IRQCTL.BUSIRQEN bit.
Note 5. Access error flag in SQCH0SR.TXIBERR, SQCH1SR.TXIBERR is set. The interrupt occurs when it is enabled in SQCH0IER.TXIBERR, or SQCH1IER.TXIBERR bits.

43.4 Device Lifecycle Management

Device Lifecycle Management (DLM) identifies the current development/production/deployment phase of the device and controls the capabilities of the debug function, the serial programming interface, and Renesas test mode. Figure 43.7 shows the available device lifecycle states and Table 43.6 shows the lifecycle state definitions and capabilities in each state.

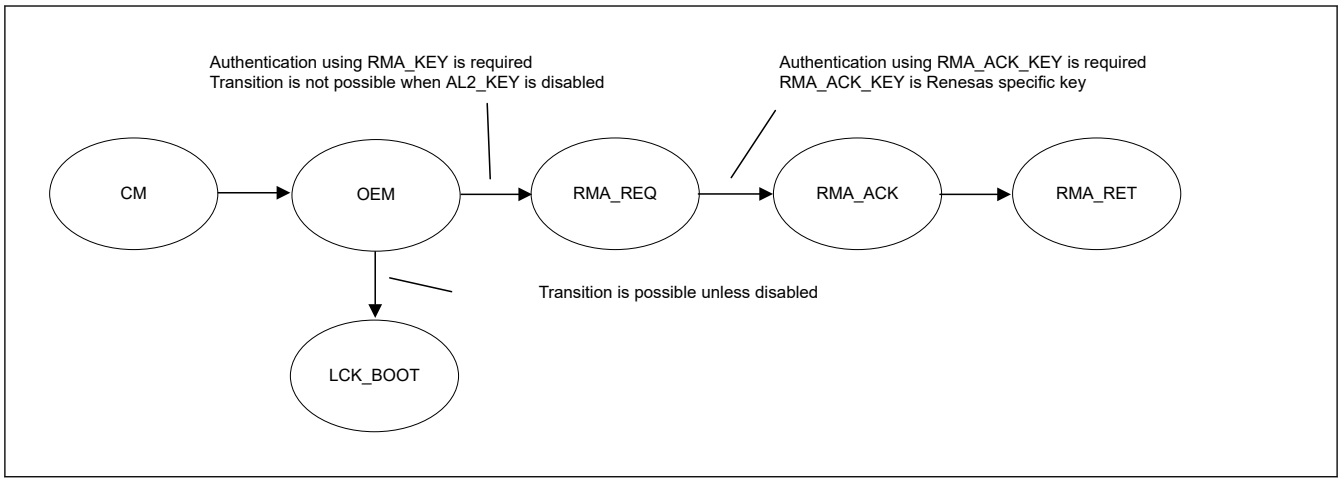


Figure 43.7 Device lifecycle states

Table 43.6 Lifecycle state definition and capabilities in each state (1 of 2)

Lifecycle	Definition	Protection level	Debug function	Serial programming	Renesas test mode
CM	“Chip Manufacturing”	PL2	Available in the secure and non-secure debug	Available Cannot access code/data flash area	Not available
OEM	“Original Equipment Manufacturer” The device is owned by the customer.	PL2 or PL1 or PL0	Depend on the authentication level		Not available
LCK_BOOT	“LoCKed BOOT interface” The debug interface and the serial programming interface are permanently disabled.	PL0	Not available	Not available	Not available

Table 43.6 Lifecycle state definition and capabilities in each state (2 of 2)

Lifecycle	Definition	Protection level	Debug function	Serial programming	Renesas test mode
RMA_REQ	“Return Material Authorization REQuest” Request for RMA. The customer must send the device to Renesas in this state.	PL0	Not available	Available Cannot access code/data flash area	Not available
RMA_ACK	“Return Material Authorization ACKnowledged” Failure analysis in Renesas	PL2	Available in the secure and non-secure debug	Available Cannot access code/data flash area	Available
RMA_RET	“Return Material Authorization RETurn” The device is back to the customer. The device does not boot.	PL0	Not available	Not available	Not available

Note: The customer receives the device in either CM or OEM state (Protection level: PL2).

43.4.1 Changing the Lifecycle State

Use the boot firmware commands to change the device lifecycle state. These commands are available via the boot firmware's UART and USB interfaces, plus SWD/JTAG. See the boot firmware application note for details of the command. The lifecycle state cannot be updated by an application, but the current lifecycle state can be read through the dedicated registers.

As shown in Figure 43.7, each transition is one-way and the state cannot be regressed.

Transition from OEM to RMA_REQ requires key authentication using the RMA_KEY which must have been previously injected by the customer. The key length of the RMA_KEY is 128 bits. Inject the RMA_KEY as shown in section 43.5. Secure Key Injection. RMA_KEY can be injected in AL2.

The key authentication uses a challenge and response authentication or authentication using the MCU's unique ID. The response (challenge and response authentication) or the authentication code (using the MCU's unique ID) can be calculated as follows:

Response = AES-128 CMAC (RMA_KEY, 128-bit challenge)

Authentication code = AES-128 CMAC (RMA_KEY, 128-bit MCU unique ID)

The contents of the flash memory except permanently locked blocks or registers are erased when transitioning to RMA_REQ. The contents of the permanently locked blocks or registers can be read by Renesas at failure analysis. A flash block can permanently locked by setting the PBPS/PBPS_SEC and BPS_SEL registers to permanently disable programming and erasure of the block. The SAS register can be permanently locked by the FSPR bit, permanently disabling programming and erasure of the register. Transition to RMA_REQ is not possible if the AL2_KEY is disabled. The MCU does not respond after changing the device lifecycle state to RMA_REQ. To continue to use boot firmware commands, you must enter boot mode again after a reset. See the boot firmware application note for details.

Transition from OEM to LCK_BOOT is possible unless that transition has been explicitly disabled. Use the parameter setting command in AL2 or AL1 to prohibit the transition to LCK_BOOT. The LCK_BOOT transition prohibition is a permanent setting and cannot be undone. The debug interface and serial programming interface are permanently disabled in LCK_BOOT.

Note: For proper DLM state management, it is required to issue an initial DLM state transition command to OEM in case of receiving in CM state.

43.4.2 Protection and Authentication Level

The protection level (PL) and the authentication level (AL) determine the availability of the debug function and the serial programming interface. PL and AL are fixed except in the OEM state. For PL and for AL, three levels can be set for each in the OEM state. AL indicates a temporary authentication status and is initialized to PL after an MCU power-on reset. Both PL and AL can be changed only by the boot firmware and cannot be changed by the application. Figure 43.8 shows the available Protection Level and Authentication Level states and transitions. Table 43.7 shows the availability of the debug function and the serial programming interface in each level.

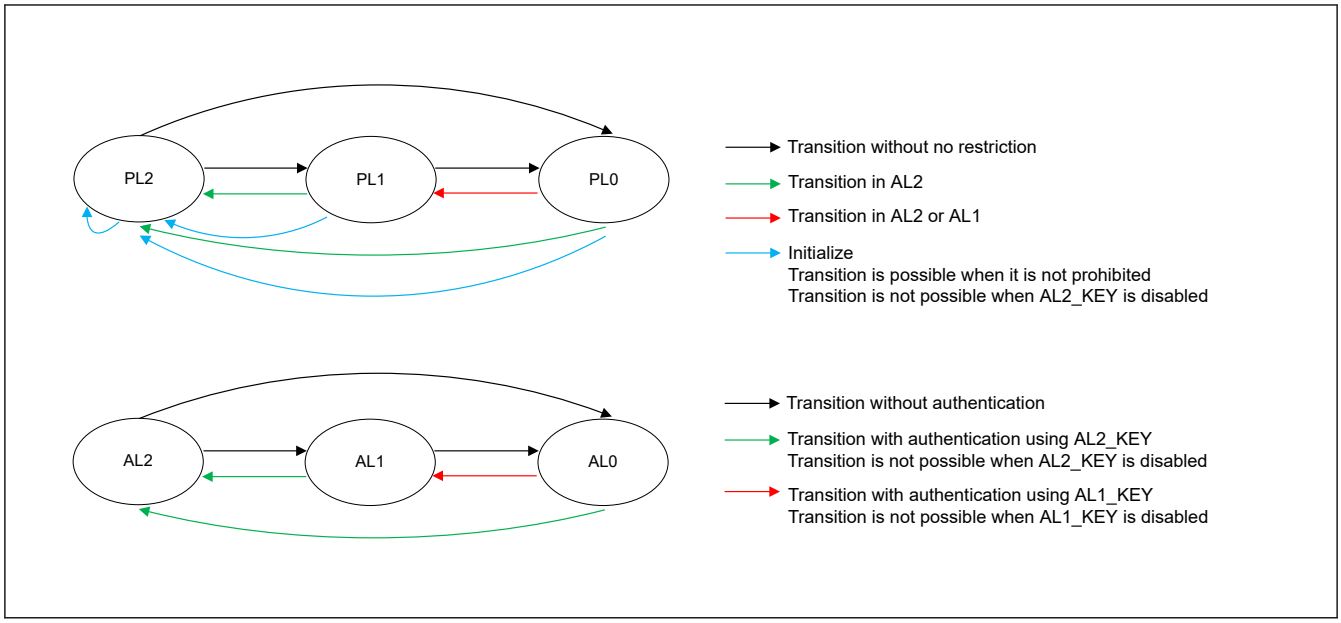


Figure 43.8 PL and AL states and transitions

Table 43.7 Availability of the debug function and the serial programming interface in each authentication level

AL	Debug function	Serial programming interface
AL2	Non-secure and Secure debug functions are enabled and accessible from the debugger	All functions are available
AL1	Only non-secure debug function is enabled, and debugger can access only defined non-secure debug accessible regions	Available but cannot program, erase, or read secure code or data flash area
AL0	No debug functions are available	Available but cannot access code or data flash area

Use the boot firmware commands to change PL and AL. See the boot firmware application note for details of the command.

Changing to a lower PL can be done with no restriction. Changing to a higher PL requires that the MCU be at that AL or higher. For example, when the current AL is AL1, changing to PL1 is possible but changing to PL2 is impossible. PL can be reset by the Initialize command unless the command itself is disabled. The Initialize command will set the PL to PL2 and the content in the flash memory will be erased. If there are any permanently locked blocks or registers, the Initialize command does not execute. In other words, when all the bits of the PBPS and PBPS_SEC registers are 1 and the FSPR bit is 1, the Initialize command is executable. The Initialize command can be disabled permanently in all AL states by the parameter setting command to prevent users from erasing the contents of flash memory. The Initialize command is disabled also when AL2_KEY is disabled. The MCU does not respond after executing the Initialize command. To continue to use the boot firmware commands, enter boot mode again after a reset. See the boot firmware application note for details.

Changing to a lower AL can be done without authentication. Changing to a higher AL requires key authentication using AL2_KEY or AL1_KEY, as appropriate. These keys are 128-bit keys. Injection of AL2_KEY or AL1_KEY is performed as shown in section 43.5. Secure Key Injection. AL2_KEY can be injected in AL2, AL1_KEY can be injected in AL2 or AL1. The key authentication uses a challenge and response authentication. The response can be calculated as follows:

Response = AES-128 CMAC (KEY, 128 bits challenge)

AL2_KEY can be disabled permanently in AL2 by parameter setting command. AL1_KEY can be disabled permanently in AL2 or AL1 by parameter setting command.

43.4.3 Serial Programming

Whether a serial programmer can be connected and the range of flash memory that can be accessed depends on the device lifecycle state and AL as shown in Table 43.6 and Table 43.7. In addition, the accepted serial programming commands differ depending on the device lifecycle state and AL. See the boot firmware application note for details of the commands.

43.4.4 Device Lifecycle State and PL Change Example

The following section describes a typical device lifecycle state and PL change example.

- Secure developer
  - Change the device lifecycle state from CM to OEM using the boot firmware command
  - Set the memory security attribution of the code flash and data flash using the boot firmware command
  - Program and debug the secure application. Debug is possible if the lifecycle is CM, but it is not possible to set the memory security attribution in CM state. If the memory security attribution is not set, all area of the code flash and data flash is secure. Inject application AES, RSA, ECC, HMAC keys listed in Table 43.8 (if required)
  - If the AL2_KEY and RMA_KEY are required, inject them using the boot firmware commands.
  - Prepare the MCU for the Non-secure developer
    - If the Non-secure Developer will not be permitted to use the Initialize command, disable it using the boot firmware command.
    - If the AL2_KEY will not be used and the Non-secure Developer will not be permitted to use the Initialize function, disable the AL2_KEY using the boot firmware command.
    - Change the PL from PL2 to PL1 using the boot firmware command
- Non-secure developer
  - Program and debug the non-secure application. Inject application AES, RSA, ECC, HMAC keys listed in Table 43.8 (if required)
  - If the AL1_KEY is required, inject it using the boot firmware command.
  - Prepare the MCU for end-product deployment
    - Disable the Initialize command using the boot firmware command (if required)
    - If the AL1_KEY will not be used, disable the AL1_KEY using the boot firmware command.
    - Change the PL from PL1 to PL0 using the boot firmware command

43.4.5 Failure Analysis

If the customer requests failure analysis by Renesas, it is necessary to send the device after changing the device lifecycle state to RMA_REQ. If the device lifecycle state is not RMA_REQ, Renesas cannot perform the failure analysis. After failure analysis, Renesas changes the lifecycle to RMA_RET and returns the device to the customer.

Note: RMA_KEY is required to change the device lifecycle state to RMA_REQ or the MCU's Unique ID must be used as part of an authentication code. See section 43.4.1. Changing the Lifecycle State for details.

43.5 Secure Key Injection

To inject a user key into the MCU, perform the following steps.

Renesas provides the Security Key Management Tool, available on the Renesas web site, to assist with key injection preparation.

1. Create a 256-bit installation key.
   This key is called the User Factory Programming Key (UFPK) and is used to wrap a user key
2. Get the wrapped version (W-UFPK) of the UFPK through the Renesas Key Wrapping Service.
3. Wrap the user key using the UFPK.
4. Send the W-UFPK and the wrapped user key to the MCU using a boot firmware interface.
   The user key is unwrapped, wrapped with the MCU's hardware unique key, and stored in nonvolatile memory. DLM keys are stored in upmapped flash. Application keys are stored at the address specified with the key injection command.

Figure 43.9 shows an example of key injection and Table 43.8 shows the keys that can be injected.

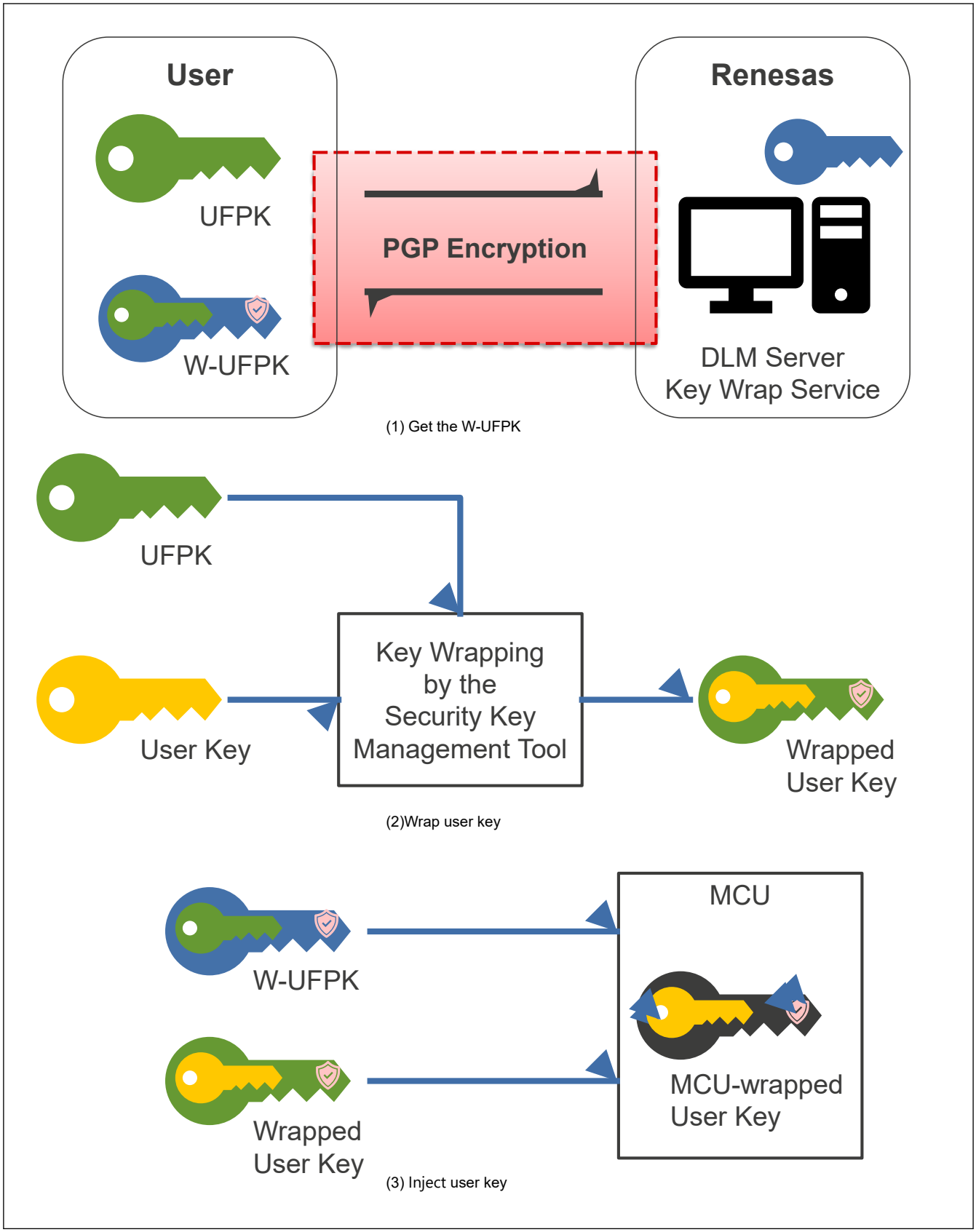


Figure 43.9 Key injection

Table 43.8 Keys that can be injected

DLM transition	AL transition	AES	RSA	ECC	HMAC	EdDSA	Secure boot	Other
RMA_KEY	AL2_KEY, AL1_KEY	AES-128, AES-192, AES-256, AES-XTS-128, AES-XTS-256	RSA-1024, RSA-2048, RSA-3072, RSA-4096 (Public and Private) RSA-2048 Public Key for TLS	secp192r1, secp224r1, secp256r1, secp384r1, secp521r1, Brainpool P256r1, Brainpool P384r1, Brainpool P521r1, Koblitz secp256k1 (Public and Private)	HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512, HMAC-SHA512/224, HMAC-SHA512/256	Ed25519 (Public and Private)	OEM_ROOT_PK	Key-Update Key

43.6 Secure Factory Programming

In addition to secure key injection support for injecting user keys (DLM, debug authentication, application, and secure boot keys), the MCU supports the programming of a firmware image in ciphertext format to prevent assets from being leaked during production programming. This enables secure factory programming in a non-secure environment. Secure factory programming is supported by boot firmware. The maximum capacity that can be programmed with the secure factory programming is almost 16 MB. See the security key management tool user's manual or the boot firmware application note for details of the capacity limitation. Figure 43.10 shows an example of secure factory programming of an encrypted firmware image. The customer wraps the Image Encryption Key with the UFPK and encrypts the image with the Image Encryption Key using AES128-CCM. When the customer sends the W-UFPK, the wrapped Image Encryption Key, and encrypted image to the MCU through the serial programming interface, the MCU decrypts and programs the firmware image.

To further support secure factory programming in a non-secure environment, the DLM state, Protection Level, and Authentication Keys can all be set via a single boot firmware command. Points to note about this boot firmware command include:

- Encrypted firmware programming can be performed only when the MCU is in the OEM state.
- This command will change the Protection Level of the MCU. The initial PL must be PL2. The final PL must be PL0.
- If the DLM state will remain in the OEM state, AL2 key must be injected. Do not inject AL1 key.
- The MCU can be transitioned to the LCK_BOOT state. In this case, AL keys cannot be injected.
- AL2 key must be wrapped with the same UFPK as the Image Encryption Key.
- This command will erase all code and data flash area except option-setting memory before programming encrypted firmware image. If there are any permanently locked blocks, this command can not be executed.
- Prohibit to execute this command if the current register setting or write value to the registers related to startup area selection and startup bank selection are other than the following.
  - SAS.BTFLG = 1b
  - BANKSEL.BANKSWP[2:0] = 111b
  - BANKSEL_SEC.BANKSWP[2:0] = 111b
- All option-setting memory values must be included in the encrypted firmware image, including those not changed from their default values. However, if the following areas are write-protected, the write data in these areas should not be included in the image. If included, this command will terminate with an error.
  - SAS register
  - Lockable Area 0 to 2 in the data flash option setting memory

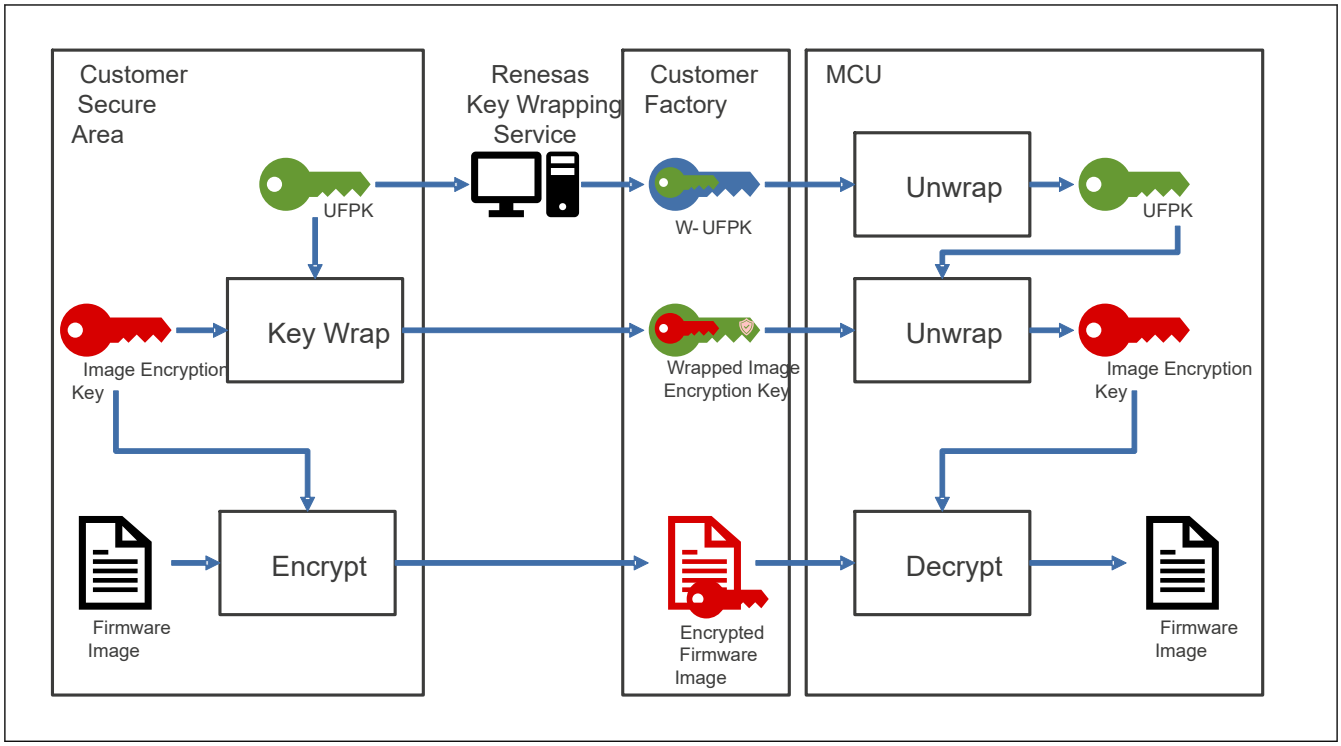


Figure 43.10 Secure factory encrypted image programming

43.7 Secure Boot

Secure boot verifies the integrity and authenticity of an OEM boot loader (called OEM_BL) starting at the beginning of application executable memory.

The OEM_BL is verified when it is initially programmed and prior to execution.

OEM_BL is verified when it is programmed using the boot firmware. Two certificates are used for validation. One is the key certificate, which is used to certify the OEM_BL verification key (called the OEM_BL_PK, the public part of the OEM_BL verification key pair). The key certificate is signed by the secret part of OEM_BL verification root key pair (called OEM_ROOT_SK). The public part of OEM_BL verification root key pair (called OEM_ROOT_PK) is registered with the MCU as the root of trust key using secure key injection as described in section 43.5. Secure Key Injection.
The OEM_ROOT_SK and OEM_ROOT_PK are a 256-bit ECC (secp256r1 curve) key pair. The SHA2-256 hash of the OEM_ROOT_PK is stored during the secure key injection process.

The other certificate for validation during programming is the code certificate, which is used to certify the OEM_BL.
The code certificate is signed by the secret part of OEM_BL verification key pair (called OEM_BL_SK). After successful validation of the OEM_BL, the HMAC value of OEM_BL and the code certificate (called OEM_BL_digest) is generated and programmed to flash. HMAC SHA2-256 is used to generate the HMAC value. A derived key from the HUK is used as the HMAC key, so the HMAC value is unique in each MCU.

In single-chip mode, the immutable first stage boot loader (called FSBL) in ROM is executed after reset when the FSBL is enabled in the FSBLCTRL0 register. When secure boot is selected in FSBLCTRL1 register, the FSBL generates the HMAC value of the OEM_BL and the code certificate, and compares it to the expected HMAC value. When the HMAC value is the same, the FSBL jumps to the OEM_BL. When CRC boot is selected in the FSBLCTRL1 register, the FSBL calculates the CRC of the OEM_BL and compares it to the expected CRC value, which is located in the code certificate. If the CRC values match, the FSBL jumps to the OEM_BL.

The Code certificate must be programmed into flash, and its location in flash must be programmed into the SACC0 register or SACC1 register. The SACC0 register specifies the start address of the code certificate when BANKSWP[2:0] = 111b in dual mode or when BTFLG = 1 in linear mode. The SACC1 register specifies the start address of the code certificate when BANKSWP[2:0] = 000b in dual mode or when BTFLG = 0 in linear mode.

When measurement reporting is enabled in the FSBLCTRL1 register, the FSBL stores the measurement report to the SRAM address specified by the SAMR register.

FSBL execution can be skipped after a software reset or deep software standby reset, as configured by the FSBLCTRL0 register.

If the generated HMAC or CRC value is different from the expected value, the high level is output to the port set by the FSBLCTRL2 register and the MCU goes to CPU Sleep mode.

Programming the hash of OEM_ROOT_PK and FSBL related registers can be permanently locked. It is strongly recommended to lock these items during production programming.

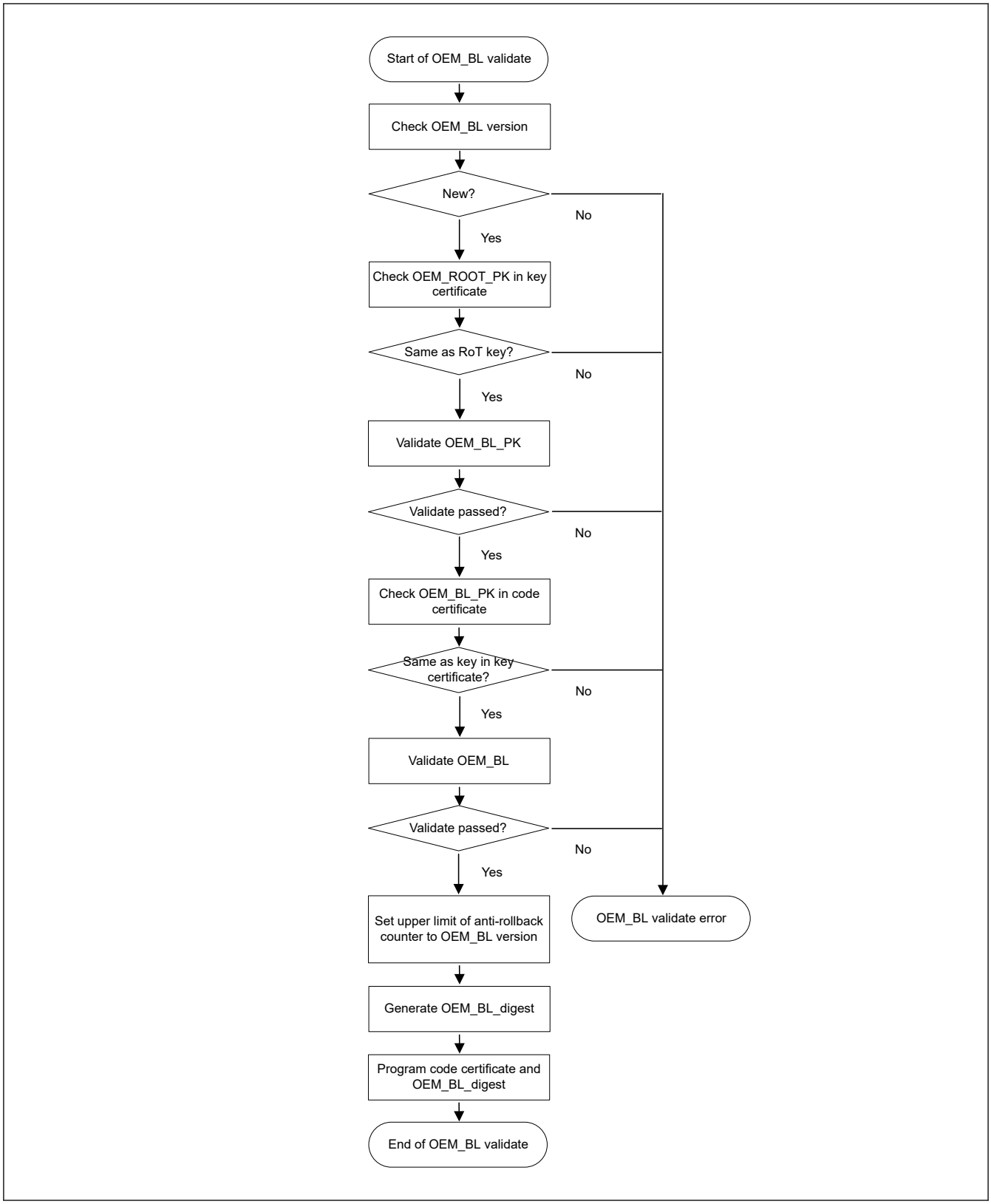


Figure 43.11 OEM_BL validation flow in serial programming mode

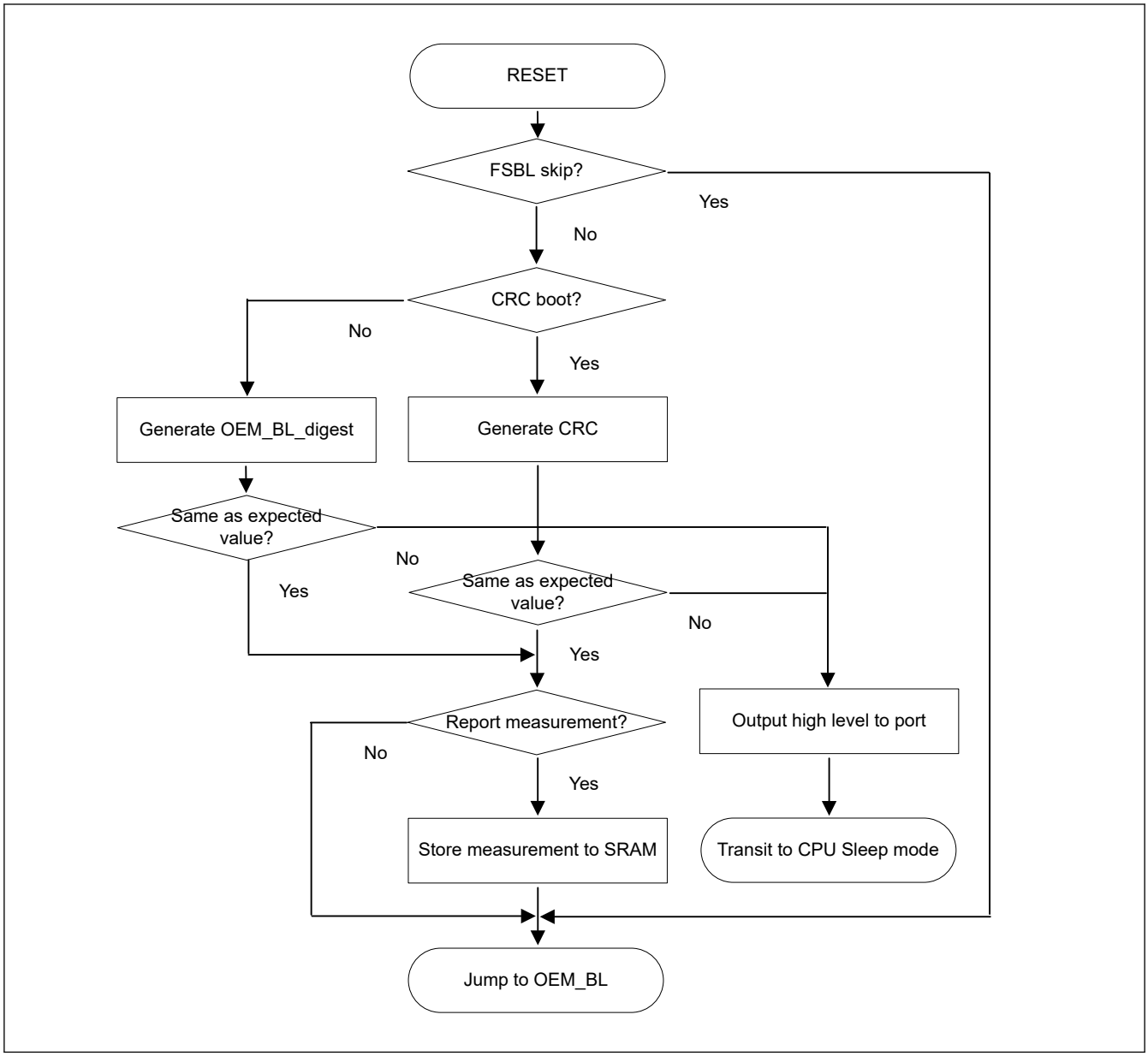


Figure 43.12 FSBL flow in single-chip mode

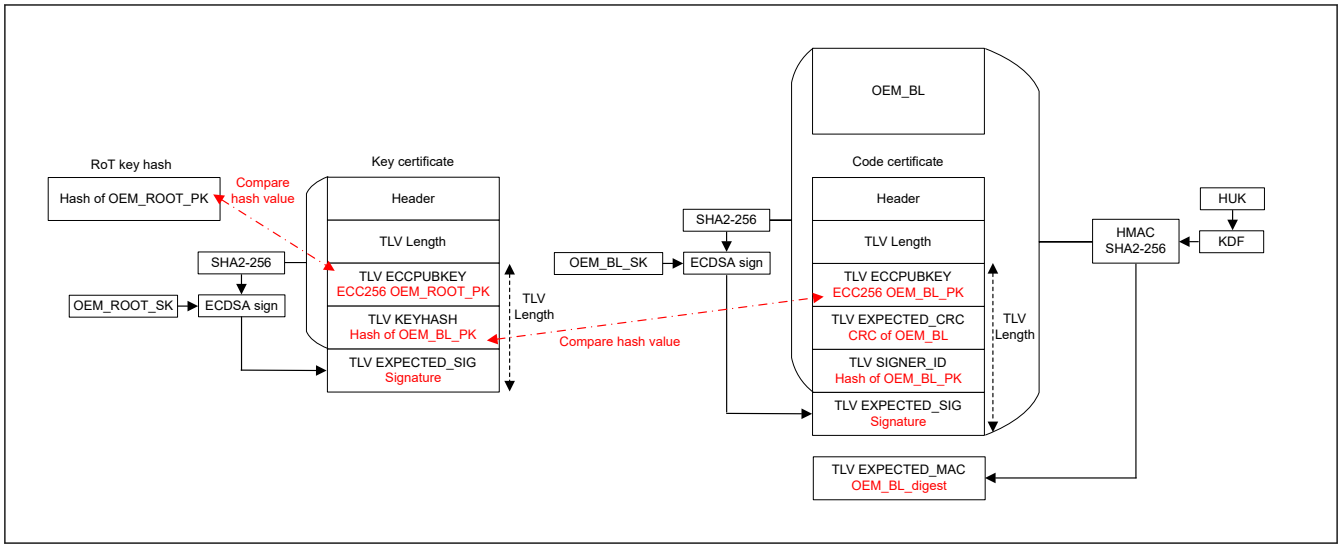


Figure 43.13 Validation structure and certificate format

Table 43.9 Detail of key certificate

Field		Size [byte]	Description
Header	Magic	4	Set to 0x6B657963
	Manifest version	4	Set to 0x00010000
	Flags	4	Reserved for future use
	Reserved	20	Reserved for future use
TLV Length		4	Length in bytes of TLV field
TLV ECCPUBKEY	Type & Length	4	Set to 0x00088010
	Value	64	ECC P-256 OEM_ROOT_PK
TLV KEYHASH	Type & Length	4	Set to 0x10144008
	Value	32	SHA2-256 hash value of OEM_BL_PK
TLV EXPECTED_SIG	Type & Length	4	Set to 0x20088410
	Value	64	Signature

Table 43.10 Detail of code certificate (1 of 2)

Field		Size [byte]	Description
Header	Magic	4	Set to 0x636F6463
	Manifest version	4	Set to 0x00010000
	Flags	4	Set to 0x00000000
	Load Addr	4	Set to 0x02000000
	Dest Addr	4	Set to 0x02000000
	Image size	4	Size in bytes of OEM_BL set in multiples of 16, minimum size is 64 bytes
	Image version	4	Version number of OEM_BL. Can be specified from 1 to 64
	Build number	4	Set to 0x00000000
TLV length		4	Length in bytes of TLV field
TLV ECCPUBKEY	Type & Length	4	Set to 0x01088010
	Value	64	ECC P-256 OEM_BL_PK
TLV EXPECTED_CRC	Type & Length	4	Set to 0x40000001
	Value	4	CRC32 of OEM_BL

Table 43.10 Detail of code certificate (2 of 2)

Field		Size [byte]	Description
TLV SIGNER_ID	Type & Length	4	Set to 0x10144008
	Value	32	SHA2-256 hash value of OEM_BL_PK
TLV EXPECTED_SIG	Type & Length	4	Set to 0x25088410
	Value	64	Signature of [Code Certificate | OEM_BL], signed by the OEM_BL_SK When generating the signature, enter the code certificate and OEM_BL in the hash function in this order.

Table 43.11 Detail of OEM_BL_digest

Field		Size [byte]	Description
TLV EXPECTED_MAC	Type & Length	4	Fixed value 0x30184008
	Value	32	Unique OEM_BL_digest in each MCU

Note: OEM_BL_digest should be programmed in the area contiguous with the code certificate.

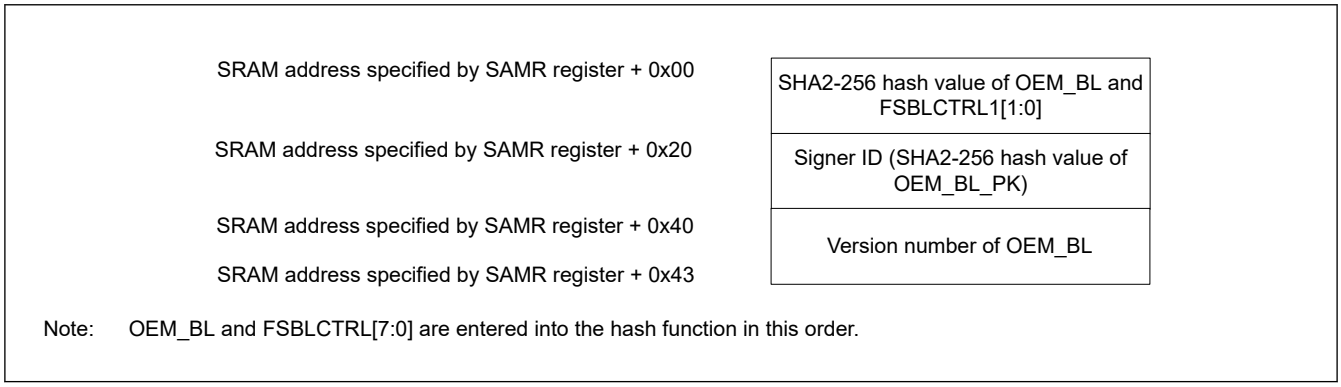
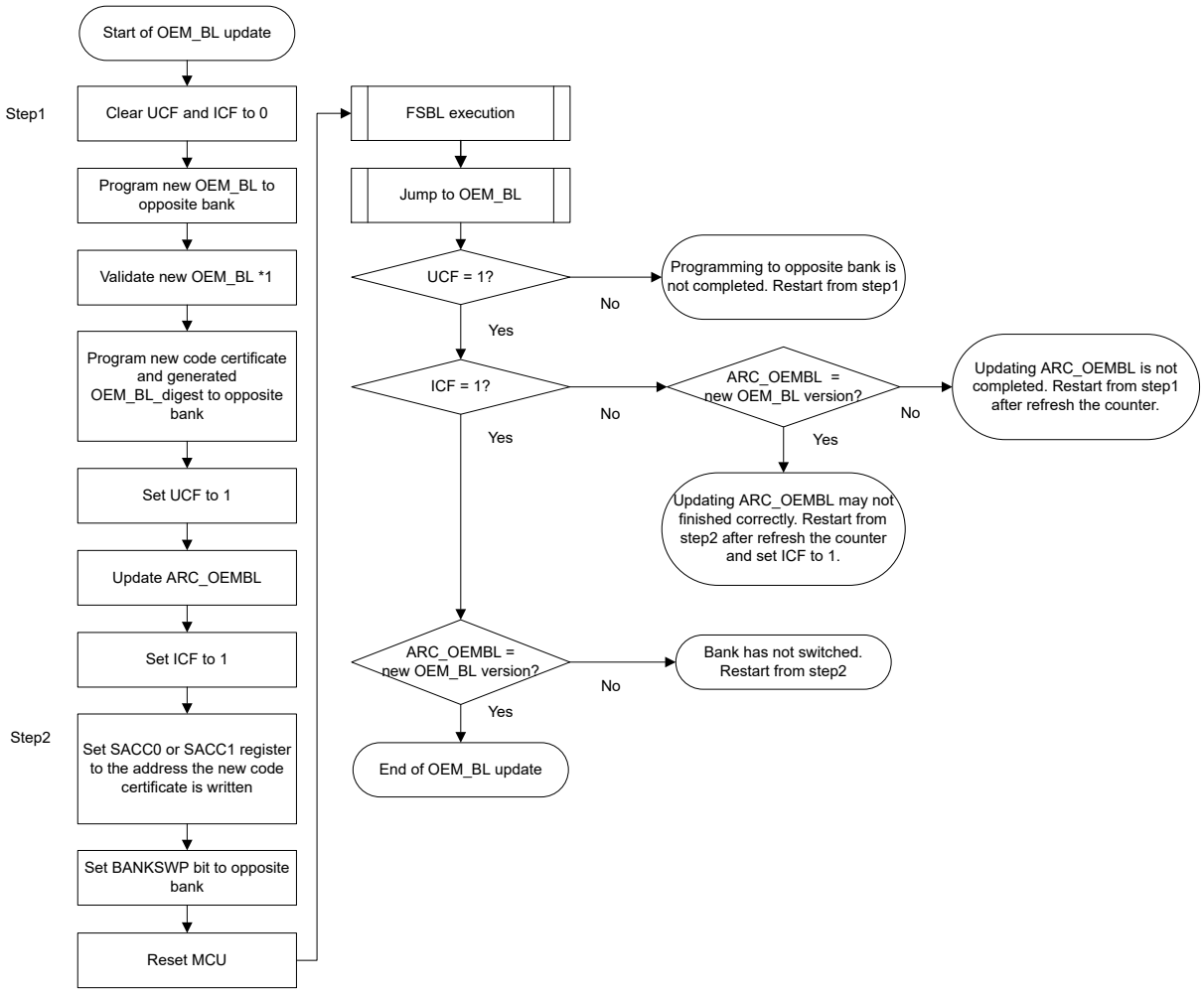


Figure 43.14 Format and location of the measurement report

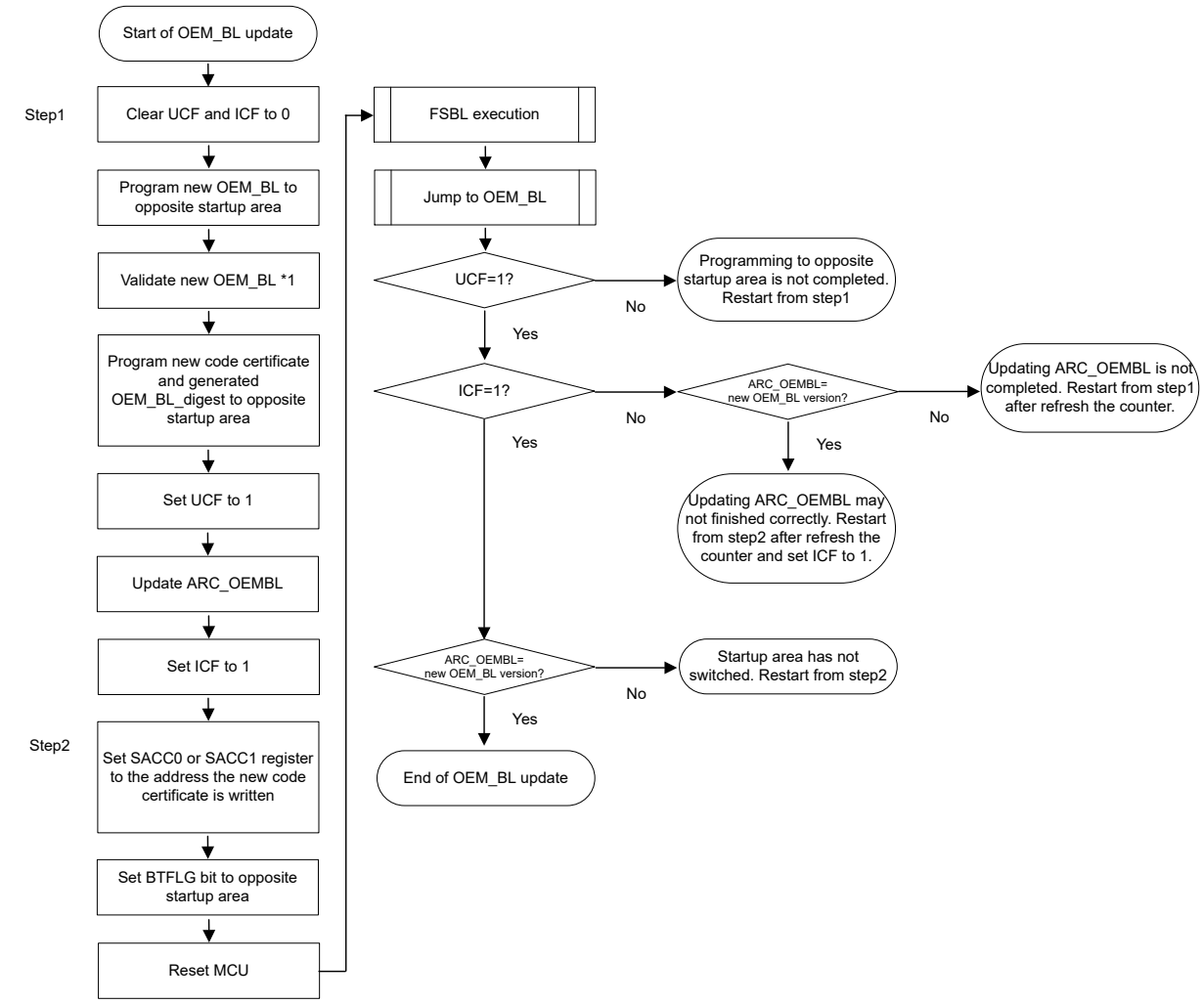
OEM_BL can be updated by application code. In this case, the application code must validate the updated bootloader, generate the OEM_BL_digest, and store it immediately after the new code certificate, as per the flow described by Figure 43.11.Figure 43.15 and Figure 43.16 show the OEM_BL update flow in dual mode or linear mode. When using linear mode, if the total size of OEM_BL, code certificate, and OEM_BL_digest is larger than the startup area size of 8 KB, you cannot use this update flow. It is recommended to use dual mode.

The implementation of the OEM_BL is user defined and application specific. However, it is recommended to implement status flags such as Update Complete Flag (UCF) and Increment Complete Flag (ICF) in the user lockable area of the data flash option-setting memory as a countermeasure against update errors due to unexpected power failure. If an update error occurs, it can be recovered as shown in the following figures. If an update error occurs, the current OEM_BL is executed after FSBL execution. Note that the update error determination program is required for both the current OEM_BL and the new OEM_BL.



Note 1. Validate new OEM_BL the same way as Figure 43.11.

Figure 43.15 OEM_BL update flow in dual mode



Note 1. Validate new OEM_BL the same way as Figure 43.11.

Figure 43.16 OEM_BL update flow in linear mode

43.8 Field Updating in Dual Mode

Be careful when performing in-field firmware update to keep the contents of secure (S or NSC) or Non-secure (NS) regions after bank swapping in the dual mode. Figure 43.17 and Figure 43.18 show the performing in-field update of information flow of secure or non-secure regions in the dual mode.

Figure 43.17 and Figure 43.18 show updating the Secure and the Non-secure regions respectively when operating in dual bank mode. Initially, BANK0 contains Version 1 of the firmware, and BANK1 is blank. As shown in Figure 43.17, to update the S or NSC firmware to Version 2, first copy the Version 1 NS firmware in BANK0 to BANK1, update the S or NSC to Version 2, and then execute the bank swap.

As shown in Figure 43.18, to update the NS firmware to Version 2, first copy the Version 1 S and NSC firmware from BANK0 to BANK1. Note that this process must be done by a secure service. After that, non-secure users should update the NS firmware to Version 2 and execute the bank swap.

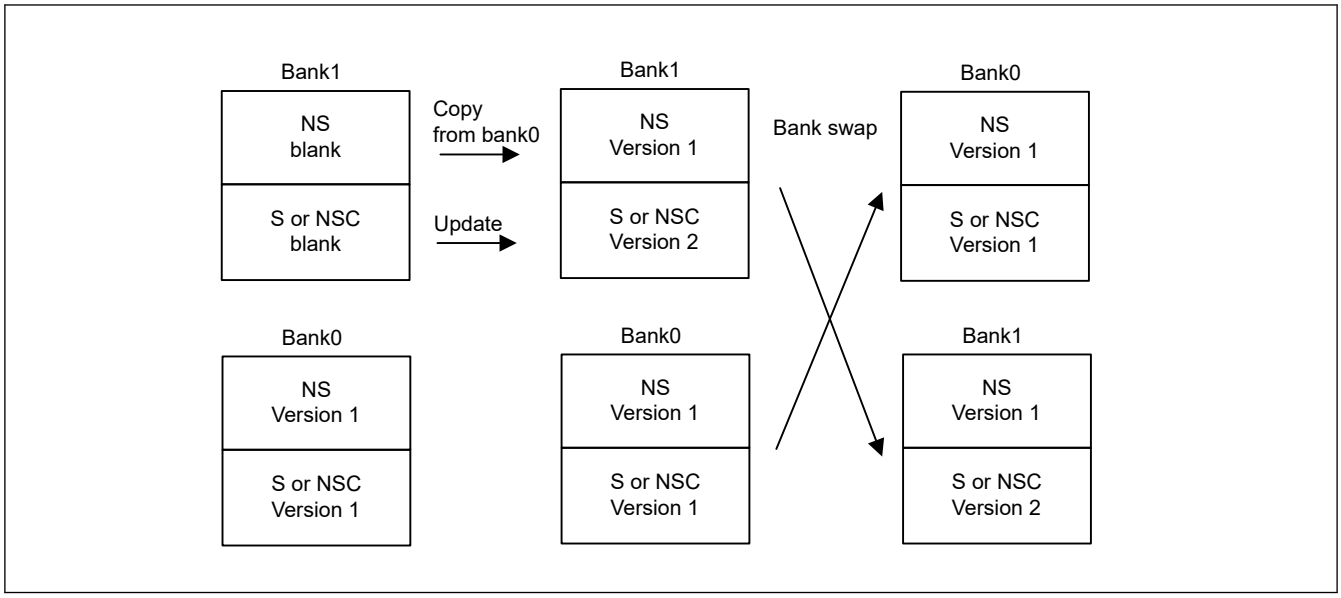


Figure 43.17 Field updating flow of secure or non-secure callable regions

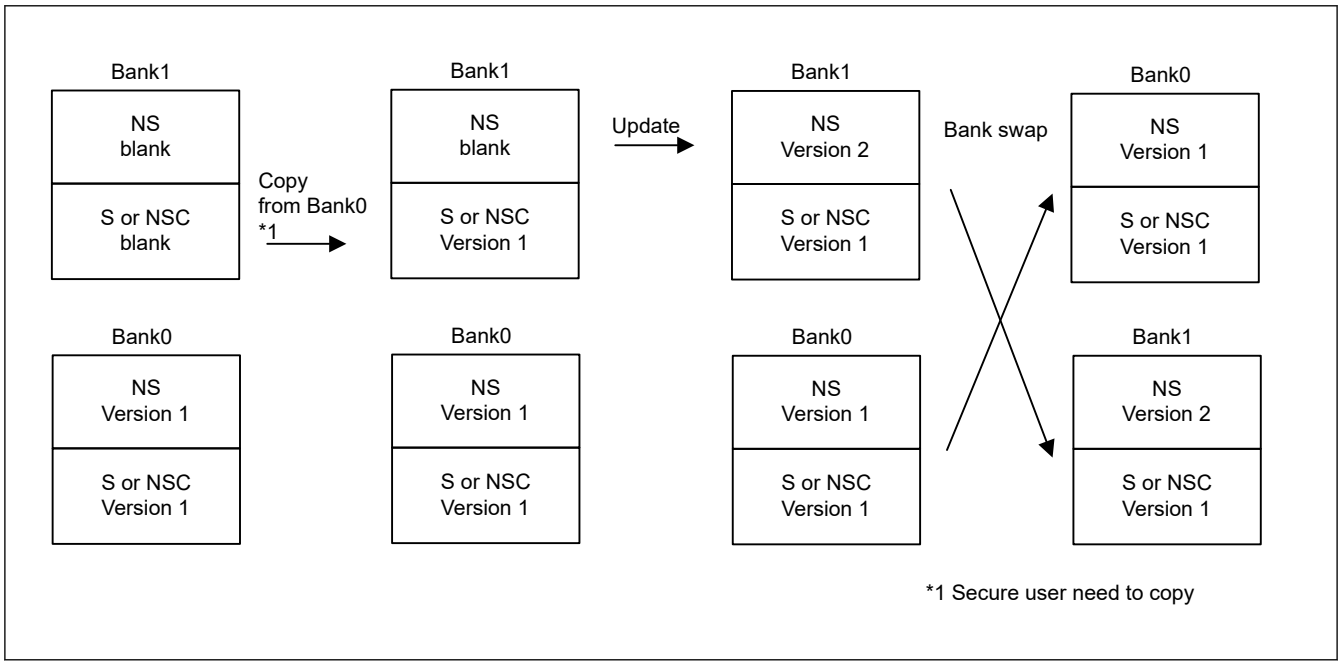


Figure 43.18 Field updating flow of non-secure region

43.9 Register Description

43.9.1 PSARB : Peripheral Security Attribution Register B

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PSARB31	PSARB30	PSARB29	PSARB28	PSARB27	—	—	—	—	PSARB22	—	—	PSARB19	PSARB18	—	PSARB16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PSARB15	—	—	PSARB12	PSARB11	—	PSARB9	PSARB8	—	—	—	PSARB4	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	PSARB4	I3C Bus Interface Security Attribution Target module: I3C and the MSTPCRB.MSTPB4 bit 0: Secure 1: Non-secure	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	PSARB8	I2C Bus Interface 1 Security Attribution Target module: IIC1 and the MSTPCRB.MSTPB8 bit 0: Secure 1: Non-secure	R/W
9	PSARB9	I2C Bus Interface 0 Security Attribution Target module: IIC0 and the MSTPCRB.MSTPB9 bit 0: Secure 1: Non-secure	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	PSARB11	Universal Serial Bus 2.0 FS Interface 0 Security Attribution Target module: USBFS0 and the MSTPCRB.MSTPB11 bit 0: Secure 1: Non-secure	R/W
12	PSARB12	Universal Serial Bus 2.0 HS Interface Security Attribution Target module: USBHS and the MSTPCRB.MSTPB12 bit 0: Secure 1: Non-secure	R/W
14:13	—	These bits are read as 0. The write value should be 0.	R/W
15	PSARB15	ETHERC/EDMAC Controller Security Attribution Target module: ETHERC/EDMAC, the MSTPCRB.MSTPB15 bit, and the PFENET.PHYMODE0 bit 0: Secure 1: Non-secure	R/W
16	PSARB16	Octa Memory Controller Security Attribution Target module: OSPI (+DOTF) and the MSTPCRB.MSTPB16 bit 0: Secure 1: Non-secure	R/W
17	—	This bit is read as 0. The write value should be 0.	R/W
18	PSARB18	Serial Peripheral Interface 1 Security Attribution Target module: RSPI1 and the MSTPCRB.MSTPB18 bit 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
19	PSARB19	Serial Peripheral Interface 0 Security Attribution Target module: RSPI0 and the MSTPCRB.MSTPB19 bit 0: Secure 1: Non-secure	R/W
21:20	—	These bits are read as 0. The write value should be 0.	R/W
22	PSARB22	Serial Communication Interface 9 Security Attribution Target module: SCI9 and the MSTPCRB.MSTPB22 bit 0: Secure 1: Non-secure	R/W
26:23	—	These bits are read as 0. The write value should be 0.	R/W
27	PSARB27	Serial Communication Interface 4 Security Attribution Target module: SCI4 and the MSTPCRB.MSTPB27 bit 0: Secure 1: Non-secure	R/W
28	PSARB28	Serial Communication Interface 3 Security Attribution Target module: SCI3 and the MSTPCRB.MSTPB28 bit 0: Secure 1: Non-secure	R/W
29	PSARB29	Serial Communication Interface 2 Security Attribution Target module: SCI2 and the MSTPCRB.MSTPB29 bit 0: Secure 1: Non-secure	R/W
30	PSARB30	Serial Communication Interface 1 Security Attribution Target module: SCI1 and the MSTPCRB.MSTPB30 bit 0: Secure 1: Non-secure	R/W
31	PSARB31	Serial Communication Interface 0 Security Attribution Target module: SCI0 and the MSTPCRB.MSTPB31 bit 0: Secure 1: Non-secure	R/W

Note: S-TYPE-1, P-TYPE-1.

The PSARB register specifies the security attribution for each module and the corresponding bit in Module Stop Control Register.

43.9.2 PSARC : Peripheral Security Attribution Register C

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PSARC31	—	—	—	PSARC27	PSARC26	—	—	—	—	—	—	—	—	—	PSARC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PSARC15	—	PSARC13	PSARC12	PSARC11	—	—	PSARC8	PSARC7	—	—	—	—	—	PSARC1	PSARC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PSARC0	Clock Frequency Accuracy Measurement Circuit Security Attribution register specifies the security attribution for each module and the corresponding bit in Module Stop Control Register. Target module: CAC and the MSTPCRC.MSTPC0 bit 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
1	PSARC1	Cyclic Redundancy Check Calculator Security Attribution Target module: CRC and the MSTPCRC.MSTPC1 bit 0: Secure 1: Non-secure	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	PSARC7	Serial Sound Interface Enhanced (channel 1) Security Attribution Target module: SSIE1 and the MSTPCRC.MSTPC7 bit 0: Secure 1: Non-secure	R/W
8	PSARC8	Serial Sound Interface Enhanced (channel 0) Security Attribution Target module: SSIE0 and the MSTPCRC.MSTPC8 bit 0: Secure 1: Non-secure	R/W
10:9	—	These bits are read as 0. The write value should be 0.	R/W
11	PSARC11	Secure Digital Host IF 1 Security Attribution Target module: SDHI1 and the MSTPCRC.MSTPC11 bit 0: Secure 1: Non-secure	R/W
12	PSARC12	Secure Digital Host IF 0 Security Attribution Target module: SDHI0 and the MSTPCRC.MSTPC12 bit 0: Secure 1: Non-secure	R/W
13	PSARC13	Data Operation Circuit Security Attribution Target module: DOC and the MSTPCRC.MSTPC13 bit 0: Secure 1: Non-secure	R/W
14	—	This bit is read as 0. The write value should be 0.	R/W
15	PSARC15	Graphic (GLCDC, MIPI-DSI, DRW) Security Attribution Target module: GLCDC, MIPI-DSI, DRW and the MSTPCRC.MSTPC4, MSTPCRC.MSTPC6, MSTPCRC.MSTPC10 bits 0: Secure 1: Non-secure	R/W
16	PSARC16	CEU Security Attribution Target module: CEU and the MSTPCRC.MSTPC16 bit 0: Secure 1: Non-secure	R/W
25:17	—	These bits are read as 0. The write value should be 0.	R/W
26	PSARC26	Controller Area Network with Flexible Data-Rate 1 Security Attribution Target module: CANFD1 and the MSTPCRC.MSTPC26 bit 0: Secure 1: Non-secure	R/W
27	PSARC27	Controller Area Network with Flexible Data-Rate 0 Security Attribution Target module: CANFD0 and the MSTPCRC.MSTPC27 bit 0: Secure 1: Non-secure	R/W
30:28	—	These bits are read as 0. The write value should be 0.	R/W
31	PSARC31	RSIP-E51A Security Attribution Target module:RSIP-E51A and the MSTPCRC.MSTPC31 bit 0: Secure 1: Non-secure	R/W

Note: S-TYPE-1, P-TYPE-1.

The PSARC register specifies the security attribution for each module and the corresponding bit in Module Stop Control Register.

43.9.3 PSARD : Peripheral Security Attribution Register D

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	PSARD28	PSARD27	—	—	—	—	PSARD22	—	PSARD20	—	—	—	PSARD16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PSARD15	PSARD14	PSARD13	PSARD12	PSARD11	—	—	—	—	—	PSARD5	PSARD4	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	PSARD4	Low Power Asynchronous General Purpose Timer 1 Security Attribution Target module: AGT1 and the MSTPCRD.MSTPD4 bit 0: Secure 1: Non-secure	R/W
5	PSARD5	Low Power Asynchronous General Purpose Timer 0 Security Attribution Target module: AGT0 and the MSTPCRD.MSTPD5 bit 0: Secure 1: Non-secure	R/W
10:6	—	These bits are read as 0. The write value should be 0.	R/W
11	PSARD11	Port Output Enable for GPT Group 3 Security Attribution Target module: POEG Group D and the MSTPCRD.MSTPD11 bit 0: Secure 1: Non-secure	R/W
12	PSARD12	Port Output Enable for GPT Group 2 Security Attribution Target module: POEG Group C and the MSTPCRD.MSTPD12 bit 0: Secure 1: Non-secure	R/W
13	PSARD13	Port Output Enable for GPT Group 1 Security Attribution Target module: POEG Group B and the MSTPCRD.MSTPD13 bit 0: Secure 1: Non-secure	R/W
14	PSARD14	Port Output Enable for GPT Group 0 Security Attribution Target module: POEG Group A and the MSTPCRD.MSTPD14 bit 0: Secure 1: Non-secure	R/W
15	PSARD15	12-Bit A/D 1 Converter Security Attribution Target module: ADC12_1 and the MSTPCRD.MSTPD15 bit 0: Secure 1: Non-secure	R/W
16	PSARD16	12-Bit A/D 0 Converter Security Attribution Target module: ADC12_0 and the MSTPCRD.MSTPD16 bit 0: Secure 1: Non-secure	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	PSARD20	12-Bit D/A Converter Security Attribution Target module: DAC12 and the MSTPCRD.MSTPD20 bit 0: Secure 1: Non-secure	R/W
21	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
22	PSARD22	Temperature Sensor Security Attribution Target module: TSN and the MSTPCRD.MSTPD22 bit 0: Secure 1: Non-secure	R/W
26:23	—	These bits are read as 0. The write value should be 0.	R/W
27	PSARD27	High Speed Analog Comparator 1 Security Attribution Target module: ACMPHS1 and the MSTPCRD.MSTPD27 bit 0: Secure 1: Non-secure	R/W
28	PSARD28	High Speed Analog Comparator 0 Security Attribution Target module: ACMPHS0 and the MSTPCRD.MSTPD28 bit 0: Secure 1: Non-secure	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1.

The PSARD register specifies the security attribution for each module and the corresponding bit in Module Stop Control Register.

43.9.4 PSARE : Peripheral Security Attribution Register E

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PSARE31	PSARE30	PSARE29	PSARE28	PSARE27	PSARE26	PSARE25	PSARE24	PSARE23	PSARE22	PSARE21	PSARE20	PSARE19	PSARE18	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PSARE9	PSARE8	—	—	—	—	PSARE3	PSARE2	PSARE1	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	PSARE1	WDT Security Attribution Target module: WDT 0: Secure 1: Non-secure	R/W
2	PSARE2	IWDT Security Attribution Target module: IWDT 0: Secure 1: Non-secure	R/W
3	PSARE3	Real Time Clock Security Attribution Target module: RTC 0: Secure 1: Non-secure	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	PSARE8	ULPT1 Security Attribution Target module: ULPT1 and the MSTPCRE.MSTPE8 bit 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
9	PSARE9	ULPT0 Security Attribution Target module: ULPT0 and the MSTPCRE.MSTPE9 bit 0: Secure 1: Non-secure	R/W
17:10	—	These bits are read as 0. The write value should be 0.	R/W
18	PSARE18	General PWM Timer Channel 13 Security Attribution Target module: GPT13 and the MSTPCRE.MSTPE18 bit 0: Secure 1: Non-secure	R/W
19	PSARE19	General PWM Timer Channel 12 Security Attribution Target module: GPT12 and the MSTPCRE.MSTPE19 bit 0: Secure 1: Non-secure	R/W
20	PSARE20	General PWM Timer Channel 11 Security Attribution Target module: GPT11 and the MSTPCRE.MSTPE20 bit 0: Secure 1: Non-secure	R/W
21	PSARE21	General PWM Timer Channel 10 Security Attribution Target module: GPT10 and the MSTPCRE.MSTPE21 bit 0: Secure 1: Non-secure	R/W
22	PSARE22	General PWM Timer Channel 9 Security Attribution Target module: GPT9 and the MSTPCRE.MSTPE22 bit 0: Secure 1: Non-secure	R/W
23	PSARE23	General PWM Timer Channel 8 Security Attribution Target module: GPT8 and the MSTPCRE.MSTPE23 bit 0: Secure 1: Non-secure	R/W
24	PSARE24	General PWM Timer Channel 7 Security Attribution Target module: GPT7 and the MSTPCRE.MSTPE24 bit 0: Secure 1: Non-secure	R/W
25	PSARE25	General PWM Timer Channel 6 Security Attribution Target module: GPT6 and the MSTPCRE.MSTPE25 bit 0: Secure 1: Non-secure	R/W
26	PSARE26	General PWM Timer Channel 5 Security Attribution Target module: GPT5 and the MSTPCRE.MSTPE26 bit 0: Secure 1: Non-secure	R/W
27	PSARE27	General PWM Timer Channel 4 Security Attribution Target module: GPT4 and the MSTPCRE.MSTPE27 bit 0: Secure 1: Non-secure	R/W
28	PSARE28	General PWM Timer Channel 3 Security Attribution Target module: GPT3 and the MSTPCRE.MSTPE28 bit 0: Secure 1: Non-secure	R/W
29	PSARE29	General PWM Timer Channel 2 Security Attribution Target module: GPT2 and the MSTPCRE.MSTPE29 bit 0: Secure 1: Non-secure	R/W
30	PSARE30	General PWM Timer Channel 1 Security Attribution Target module: GPT1 and the MSTPCRE.MSTPE30 bit 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
31	PSARE31	General PWM Timer Channel 0 Security Attribution Target module: GPT0, GPT_OPS and the MSTPCRE.MSTPE31 bit 0: Secure 1: Non-secure	R/W

Note: S-TYPE-1, P-TYPE-1.

The PSARE register specifies the security attribution for each module and the corresponding bit in Module Stop Control Register.

43.9.5 MSSAR : Module Stop Security Attribution Register

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSSAR31	—	—	—	—	—	—	—	—	MSSAR22	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MSSAR15	—	—	—	—	—	—	—	—	—	—	—	—	—	MSSAR1	MSSAR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MSSAR0	SRAM0 Clock Stop Security Attribution Target module: the MSTPCRA.MSTPA0 bit 0: Secure 1: Non-secure	R/W
1	MSSAR1	SRAM1 Clock Stop Security Attribution Target module: the MSTPCRA.MSTPA1 bit 0: Secure 1: Non-secure	R/W
14:2	—	These bits are read as 0. The write value should be 0.	R/W
15	MSSAR15	Standby RAM Clock Stop Security Attribution Target module: the MSTPCRA.MSTPA15 bit 0: Secure 1: Non-secure	R/W
21:16	—	These bits are read as 0. The write value should be 0.	R/W
22	MSSAR22	DMAC/DTC Clock Stop Security Attribution Target module: the MSTPCRA.MSTPA22 bit 0: Secure 1: Non-secure	R/W
30:23	—	These bits are read as 0. The write value should be 0.	R/W
31	MSSAR31	ELC Clock Stop Security Attribution Target module: the MSTPCRC.MSTPC14 bit 0: Secure 1: Non-secure	R/W

Note: S-TYPE-1, P-TYPE-1.

The MSSAR register specifies the security attribution for the corresponding bit in Module Stop Control Register.

43.9.6 PPARB : Peripheral Privilege Attribution Register B

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x1C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PPARB31	PPARB30	PPARB29	PPARB28	PPARB27	—	—	—	—	PPARB22	—	—	PPARB19	PPARB18	—	PPARB16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PPARB15	—	—	PPARB12	PPARB11	—	PPARB9	PPARB8	—	—	—	PPARB4	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 1. The write value should be 1.	R/W
4	PPARB4	I3C Bus Interface Privilege Attribution Target module: I3C and the MSTPCRB.MSTPB4 bit 0: Privileged 1: UnPrivileged	R/W
7:5	—	These bits are read as 1. The write value should be 1.	R/W
8	PPARB8	I2C Bus Interface 1 Privilege Attribution Target module: IIC1 and the MSTPCRB.MSTPB8 bit 0: Privileged 1: UnPrivileged	R/W
9	PPARB9	I2C Bus Interface 0 Privilege Attribution Target module: IIC0 and the MSTPCRB.MSTPB9 bit 0: Privileged 1: UnPrivileged	R/W
10	—	This bit is read as 1. The write value should be 1.	R/W
11	PPARB11	Universal Serial Bus 2.0 FS Interface 0 Privilege Attribution Target module: USBFS0 and the MSTPCRB.MSTPB11 bit 0: Privileged 1: UnPrivileged	R/W
12	PPARB12	Universal Serial Bus 2.0 HS Interface 0 Privilege Attribution Target module: USBHS and the MSTPCRB.MSTPB12 bit 0: Privileged 1: UnPrivileged	R/W
14:13	—	These bits are read as 1. The write value should be 1.	R/W
15	PPARB15	ETHERC/EDMAC Controller Privilege Attribution Target module: ETHERC/EDMAC, the MSTPCRB.MSTPB15 bit, and the PFENET.PHYMODE0 bit 0: Privileged 1: UnPrivileged	R/W
16	PPARB16	Octa Memory Controller Privilege Attribution Target module: OSPI (+DOTF) and the MSTPCRB.MSTPB16 bit 0: Privileged 1: UnPrivileged	R/W
17	—	This bit is read as 1. The write value should be 1.	R/W
18	PPARB18	Serial Peripheral Interface 1 Privilege Attribution Target module: RSPI1 and the MSTPCRB.MSTPB18 bit 0: Privileged 1: UnPrivileged	R/W

Bit	Symbol	Function	R/W
19	PPARB19	Serial Peripheral Interface 0 Privilege Attribution Target module: RSPI0 and the MSTPCRB.MSTPB19 bit 0: Privileged 1: UnPrivileged	R/W
21:20	—	These bits are read as 1. The write value should be 1.	R/W
22	PPARB22	Serial Communication Interface 9 Privilege Attribution Target module: SCI9 and the MSTPCRB.MSTPB22 bit 0: Privileged 1: UnPrivileged	R/W
26:23	—	These bits are read as 1. The write value should be 1.	R/W
27	PPARB27	Serial Communication Interface 4 Privilege Attribution Target module: SCI4 and the MSTPCRB.MSTPB27 bit 0: Privileged 1: UnPrivileged	R/W
28	PPARB28	Serial Communication Interface 3 Privilege Attribution Target module: SCI3 and the MSTPCRB.MSTPB28 bit 0: Privileged 1: UnPrivileged	R/W
29	PPARB29	Serial Communication Interface 2 Privilege Attribution Target module: SCI2 and the MSTPCRB.MSTPB29 bit 0: Privileged 1: UnPrivileged	R/W
30	PPARB30	Serial Communication Interface 1 Privilege Attribution Target module: SCI1 and the MSTPCRB.MSTPB30 bit 0: Privileged 1: UnPrivileged	R/W
31	PPARB31	Serial Communication Interface 0 Privilege Attribution Target module: SCI0 and the MSTPCRB.MSTPB31 bit 0: Privileged 1: UnPrivileged	R/W

Note: S-TYPE-2, P-TYPE-1.

The PPARB register specifies the privilege attribution for each module and the corresponding bit in Module Stop Control Register.

43.9.7 PPARC : Peripheral Privilege Attribution Register C

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PPARC31	—	—	—	PPARC27	PPARC26	—	—	—	—	—	—	—	—	—	PPARC16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PPARC15	—	PPARC13	PPARC12	PPARC11	—	—	PPARC8	PPARC7	—	—	—	—	—	PPARC1	PPARC0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	PPARC0	Clock Frequency Accuracy Measurement Circuit Privilege Attribution Target module: CAC and the MSTPCRC.MSTPC0 bit 0: Privileged 1: UnPrivileged	R/W

Bit	Symbol	Function	R/W
1	PPARC1	Cyclic Redundancy Check Calculator Privilege Attribution Target module: CRC and the MSTPCRC.MSTPC1 bit 0: Privileged 1: UnPrivileged	R/W
6:2	—	These bits are read as 1. The write value should be 1.	R/W
7	PPARC7	Serial Sound Interface Enhanced (Channel 1) Privilege Attribution Target module: SSIE1 and the MSTPCRC.MSTPC7 bit 0: Privileged 1: UnPrivileged	R/W
8	PPARC8	Serial Sound Interface Enhanced (Channel 0) Privilege Attribution Target module: SSIE0 and the MSTPCRC.MSTPC8 bit 0: Privileged 1: UnPrivileged	R/W
10:9	—	These bits are read as 1. The write value should be 1.	R/W
11	PPARC11	Secure Digital Host IF 1 Privilege Attribution Target module: SDHI1 and the MSTPCRC.MSTPC11 bit 0: Privileged 1: UnPrivileged	R/W
12	PPARC12	Secure Digital Host IF 0 Privilege Attribution Target module: SDHI0 and the MSTPCRC.MSTPC12 bit 0: Privileged 1: UnPrivileged	R/W
13	PPARC13	Data Operation Circuit Privilege Attribution Target module: DOC and the MSTPCRC.MSTPC13 bit 0: Privileged 1: UnPrivileged	R/W
14	—	This bit is read as 1. The write value should be 1.	R/W
15	PPARC15	Graphic (GLCDC, MIPI-DSI, DRW) Privilege Attribution Target module: GLCDC, MIPI-DSI, DRW and the MSTPCRC.MSTPC4, MSTPCRC.MSTPC6, MSTPCRC.MSTPC10 bits 0: Privileged 1: UnPrivileged	R/W
16	PPARC16	CEU Privilege Attribution Target module: CEU and the MSTPCRC.MSTPC16 bit 0: Privileged 1: UnPrivileged	R/W
25:17	—	These bits are read as 1. The write value should be 1.	R/W
26	PPARC26	Controller Area Network with Flexible Data-Rate 1 Privilege Attribution Target module: CANFD1 and the MSTPCRC.MSTPC26 bit 0: Privileged 1: UnPrivileged	R/W
27	PPARC27	Controller Area Network with Flexible Data-Rate 0 Privilege Attribution Target module: CANFD0 and the MSTPCRC.MSTPC27 bit 0: Privileged 1: UnPrivileged	R/W
30:28	—	These bits are read as 1. The write value should be 1.	R/W
31	PPARC31	RSIP-E51A Privilege Attribution Target module: RSIP-E51A and the MSTPCRC.MSTPC31 bit 0: Privileged 1: UnPrivileged	R/W

Note: S-TYPE-2, P-TYPE-1.

The PPARC register specifies the privilege attribution for each module and the corresponding bit in Module Stop Control Register.

43.9.8 PPARD : Peripheral Privilege Attribution Register D

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	PPARD28	PPARD27	—	—	—	—	PPARD22	—	PPARD20	—	—	—	PPARD16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PPARD15	PPARD14	PPARD13	PPARD12	PPARD11	—	—	—	—	—	PPARD5	PPARD4	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 1. The write value should be 1.	R/W
4	PPARD4	Low Power Asynchronous General Purpose Timer 1 Privilege Attribution Target module: AGT1 and the MSTPCRD.MSTPD4 bit 0: Privileged 1: UnPrivileged	R/W
5	PPARD5	Low Power Asynchronous General Purpose Timer 0 Privilege Attribution Target module: AGT0 and the MSTPCRD.MSTPD5 bit 0: Privileged 1: UnPrivileged	R/W
10:6	—	These bits are read as 1. The write value should be 1.	R/W
11	PPARD11	Port Output Enable for GPT Group 3 Privilege Attribution Target module: POEG Group D and the MSTPCRD.MSTPD11 bit 0: Privileged 1: UnPrivileged	R/W
12	PPARD12	Port Output Enable for GPT Group 2 Privilege Attribution Target module: POEG Group C and the MSTPCRD.MSTPD12 bit 0: Privileged 1: UnPrivileged	R/W
13	PPARD13	Port Output Enable for GPT Group 1 Privilege Attribution Target module: POEG Group B and the MSTPCRD.MSTPD13 bit 0: Privileged 1: UnPrivileged	R/W
14	PPARD14	Port Output Enable for GPT Group 0 Privilege Attribution Target module: POEG Group A and the MSTPCRD.MSTPD14 bit 0: Privileged 1: UnPrivileged	R/W
15	PPARD15	12-Bit A/D 1 Converter Privilege Attribution Target module: ADC12_1 and the MSTPCRD.MSTPD15 bit 0: Privileged 1: UnPrivileged	R/W
16	PPARD16	12-Bit A/D 0 Converter Privilege Attribution Target module: ADC12_0 and the MSTPCRD.MSTPD16 bit 0: Privileged 1: UnPrivileged	R/W
19:17	—	These bits are read as 1. The write value should be 1.	R/W
20	PPARD20	12-Bit D/A Converter Privilege Attribution Target module: DAC12 and the MSTPCRD.MSTPD20 bit 0: Privileged 1: UnPrivileged	R/W
21	—	This bit is read as 1. The write value should be 1.	R/W

Bit	Symbol	Function	R/W
22	PPARD22	Temperature Sensor Privilege Attribution Target module: TSN and the MSTPCRD.MSTPD22 bit 0: Privileged 1: UnPrivileged	R/W
26:23	—	These bits are read as 1. The write value should be 1.	R/W
27	PPARD27	High speed analog Comparator 1 Privilege Attribution Target module: ACMPHS1 and the MSTPCRD.MSTPD27 bit 0: Privileged 1: UnPrivileged	R/W
28	PPARD28	High speed analog Comparator 0 Privilege Attribution Target module: ACMPHS0 and the MSTPCRD.MSTPD28 bit 0: Privileged 1: UnPrivileged	R/W
31:29	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-2, P-TYPE-1.

The PPARD register specifies the privilege attribution for each module and the corresponding bit in Module Stop Control Register.

43.9.9 PPARE : Peripheral Privilege Attribution Register E

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PPARE31	PPARE30	PPARE29	PPARE28	PPARE27	PPARE26	PPARE25	PPARE24	PPARE23	PPARE22	PPARE21	PPARE20	PPARE19	PPARE18	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PPARE9	PPARE8	—	—	—	—	PPARE3	PPARE2	PPARE1	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	—	This bit is read as 1. The write value should be 1.	R/W
1	PPARE1	WDT Privilege Attribution Target module: WDT 0: Privileged 1: UnPrivileged	R/W
2	PPARE2	IWDT Privilege Attribution Target module: IWDT 0: Privileged 1: UnPrivileged	R/W
3	PPARE3	Real Time Clock Privilege Attribution Target module: RTC 0: Privileged 1: UnPrivileged	R/W
7:4	—	These bits are read as 1. The write value should be 1.	R/W
8	PPARE8	ULPT1 Privilege Attribution Target module: ULPT1 and the MSTPCRE.MSTPE8 bit 0: Privileged 1: UnPrivileged	R/W

Bit	Symbol	Function	R/W
9	PPARE9	ULPT0 Privilege Attribution Target module: ULPT0 and the MSTPCRE.MSTPE9 bit 0: Privileged 1: UnPrivileged	R/W
17:10	—	These bits are read as 1. The write value should be 1.	R/W
18	PPARE18	General PWM Timer Channel 13 Privilege Attribution Target module: : GPT13 and the MSTPCRE.MSTPE18 bit 0: Privileged 1: UnPrivileged	R/W
19	PPARE19	General PWM Timer Channel 12 Privilege Attribution Target module: : GPT12 and the MSTPCRE.MSTPE19 bit 0: Privileged 1: UnPrivileged	R/W
20	PPARE20	General PWM Timer Channel 11 Privilege Attribution Target module: : GPT11 and the MSTPCRE.MSTPE20 bit 0: Privileged 1: UnPrivileged	R/W
21	PPARE21	General PWM Timer Channel 10 Privilege Attribution Target module: : GPT10 and the MSTPCRE.MSTPE21 bit 0: Privileged 1: UnPrivileged	R/W
22	PPARE22	General PWM Timer Channel 9 Privilege Attribution Target module: : GPT9 and the MSTPCRE.MSTPE22 bit 0: Privileged 1: UnPrivileged	R/W
23	PPARE23	General PWM Timer Channel 8 Privilege Attribution Target module: : GPT8 and the MSTPCRE.MSTPE23 bit 0: Privileged 1: UnPrivileged	R/W
24	PPARE24	General PWM Timer Channel 7 Privilege Attribution Target module: : GPT7 and the MSTPCRE.MSTPE24 bit 0: Privileged 1: UnPrivileged	R/W
25	PPARE25	General PWM Timer Channel 6 Privilege Attribution Target module: : GPT6 and the MSTPCRE.MSTPE25 bit 0: Privileged 1: UnPrivileged	R/W
26	PPARE26	General PWM Timer Channel 5 Privilege Attribution Target module: : GPT5 and the MSTPCRE.MSTPE26 bit 0: Privileged 1: UnPrivileged	R/W
27	PPARE27	General PWM Timer Channel 4 Privilege Attribution Target module: : GPT4 and the MSTPCRE.MSTPE27 bit 0: Privileged 1: UnPrivileged	R/W
28	PPARE28	General PWM Timer Channel 3 Privilege Attribution Target module: : GPT3 and the MSTPCRE.MSTPE28 bit 0: Privileged 1: UnPrivileged	R/W
29	PPARE29	General PWM Timer Channel 2 Privilege Attribution Target module: : GPT2 and the MSTPCRE.MSTPE29 bit 0: Privileged 1: UnPrivileged	R/W
30	PPARE30	General PWM Timer Channel 1 Privilege Attribution Target module: : GPT1 and the MSTPCRE.MSTPE30 bit 0: Privileged 1: UnPrivileged	R/W

Bit	Symbol	Function	R/W
31	PPARE31	General PWM Timer Channel 0 Privilege Attribution Target module: : GPT0 and the MSTPCRE.MSTPE31 bit 0: Privileged 1: UnPrivileged	R/W

Note: S-TYPE-2, P-TYPE-1.

The PPARE register specifies the privilege attribution for each module and the corresponding bit in Module Stop Control Register.

43.9.10 MSPAR : Module Stop Privilege Attribution Register

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSPAR31	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
30:0	—	These bits are read as 1. The write value should be 1.	R/W
31	MSPAR31	ELC Clock Stop Privilege Attribution Target module: the MSTPCRC.MSTPC14 bit 0: Privileged 1: UnPrivileged	R/W

Note: S-TYPE-2, P-TYPE-1.

The MSPAR register specifies the privilege attribution for the corresponding bit in the Module Stop Control Register.

43.9.11 CFSAMONA : Code Flash Security Attribution Monitor Register A

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x30

Bit position:	31								23								15								0
Bit field:	—	—	—	—	—	—	—	—	CFS2[8:0]								—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	*1	*1	*1	*1	*1	*1	*1	*1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note 1. The value in a blank product is 0x1FF. It is set to the value written by your application.

Bit	Symbol	Function	R/W
14:0	—	These bits are read as 0.	R
23:15	CFS2[8:0]	Code Flash Secure Area Indicate the area of secure region for code flash.	R
31:24	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-5.

43.9.12 DFSAMON : Data Flash Security Attribution Monitor Register

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x34

Bit position:	31																15						10									0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DFS[5:0]						—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1	*1	*1	0	0	0	0	0	0	0	0	0	0

Note 1. The value in a blank product is 0x3F. It is set to the value written by your application.

Bit	Symbol	Function	R/W
9:0	—	These bits are read as 0.	R
15:10	DFS[5:0]	Data Flash Secure Area Indicate the area of secure region for data flash.	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-5.

43.9.13 DLMMON : Device Lifecycle Management State Monitor Register

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	DLMMON[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1

Note 1. The value in a blank product is 0001b. These bits depend on DLM status.

Bit	Symbol	Function	R/W
3:0	DLMMON[3:0]	Device Lifecycle Management State Monitor Indicate DLM status value. 0x0: Reserved 0x1: CM 0x2: Reserved 0x3: Reserved 0x4: OEM 0x5: Reserved 0x6: LCK_BOOT 0x7: RMA_REQ 0x8: RMA_ACK 0x9: RMA_RET Others: Reserved	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-5.

43.9.14 MSAOAD : Master Security Attribution Operation After Detection Register

Base address: BUS = 0x4000_3000

Offset address: 0x1010

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OAD	Operation after detection 0: NMI 1: Reset	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writing of the OAD bit.	W

Note: S-TYPE-6, P-TYPE-2.

OAD bit (Operation after detection)

The OAD bit specifies operation when the access violation is detected. When OAD = 0, error response is returned and NMI is generated. When OAD = 1, reset request is generated. When writing to the OAD bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the OAD bit. When writing to the OAD bit, write 0xA5 simultaneously to the KEY[7:0] bits. If this value is not written to the KEY[7:0] bits, the OAD bit is not updated.

43.9.15 MSAPT : Master Security Attribution Protect Register

Base address: BUS = 0x4000_3000

Offset address: 0x1014

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MSAOAD register writing is possible. 1: MSAOAD register writing is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writing of the PROTECT bit.	W

Note: S-TYPE-6, P-TYPE-2.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writes to the associated registers to be protected.

MSAPT.PROTECT controls the MSAOAD register.

When the PROTECT bit is set simultaneously, write 0xA5 to the KEY[7:0] bits using half word access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. If this value is not written to the KEY[7:0] bits, the PROTECT bit is not updated.

43.10 Usage Note

43.10.1 Security or Privilege Bit Write Timing

When writing a security or privilege attribution bit, confirm that the write is complete by reading the security or privilege register until it matches the value written. The protection is not effective until the write to the security or privilege register is complete.

44. Renesas Secure IP (RSIP-E51A)

44.1 Overview

This security engine consists of an isolated subsystem comprising an access management circuit, storage area, encryption/decryption circuit, and random number generation circuit. In combination with the RSIP library, the security engine can prevent eavesdropping (to maintain confidentiality), falsification of information (to ensure integrity), and impersonation (to verify authenticity).

Because key information required for encryption and decryption is stored only in the security engine and all accesses from the outside can be blocked, the security engine enables building a more robust security system.

Table 44.1 lists the security engine specifications. Figure 44.1 shows the security engine block diagram.

Table 44.1 Security engine specifications (1 of 2)

Parameter	Specifications
Access control	Access management circuit ● In case of irregular access to the security engine due to a tampered program or CPU runaway, this circuit blocks all subsequent accesses and stops data output from the security engine
Symmetric cryptography	AES: Compliant with NIST FIPS PUB 197 ● Key length: 128, 192, or 256 bits ● Data block size: 128 bits ● AES supports following block cipher mode ECB, CBC, CTR: Compliant with NIST SP 800-38A CCM: Compliant with NIST SP 800-38C GCM: Compliant with NIST SP 800-38D XTS: Compliant with IEEE 1619-2007 ● AES supports following authentication algorithm CMAC: Compliant with NIST SP 800-38B GMAC: Compliant with NIST SP 800-38D ● AES does not support the KEY size of 192-bit in CCM, GCM, XTS, CMAC and GMAC.
Random number generation	128-bit true random number generation circuit
Asymmetric cryptography	RSA ● Maximum number of operable bits: 4224 bits ● Supported key sizes: 1024-bit, 2048-bit, 3072-bit, and 4096-bit ● Signature generation, signature verification, public-key encryption, private-key decryption ECC ● Maximum number of operable bits: 576 bits ● Support for curve – NIST P-192, P-224, P-256, P-384 and P-521 – Brainpool P256r1, P384r1, and P512r1 – Ed25519 – secp256k1 ● Signature generation, signature verification, key generation
Message digest computation	HASH ● Block size: 512-bit(SHA-224, SHA-256) 1024-bit(SHA-512/224, SHA-512/256, SHA-384, SHA-512) ● Key size: 512-bit or less ● HASH supports following secure hash algorithms SHA-224, SHA-256, SHA-512/224, SHA-512/256, SHA-384 and SHA-512: Compliant with FIPS PUB 180-4 ● HASH supports following message authentication algorithm HMAC: Compliant with FIPS PUB 198

Table 44.1 Security engine specifications (2 of 2)

Parameter	Specifications
Hardware Unique Key	• A read-only, 256-bit Hardware Unique Key (HUK) is exclusively accessible by the security engine access management circuit through a dedicated bus. • Key derivation functions (KDFs) combine the Hardware Unique Key with the key generation information. The derived keys implement the key wrapping for user key secure storage. • The HUK uniqueness prevents the illicit cloning and copying of keys to another MCU of the MCU group. • The HUK itself is stored in wrapped (encrypted, non-plain) format in an isolated memory area. Therefore, it is protected from illicit access and copy.
Application Key Management	• Wrapped keys are only valid within the security engine.
Unique ID	• A read-only, 128-bit ID unique to an MCU (Unique ID) is accessible from the access management circuit. • Key derivation functions (KDFs) combine the Unique ID with the key generation information. Such derived keys are used to unwrap the HUK within the security engine boundary.
OEM boot loader version	• The security engine outputs OEM boot loader version (OEM BL Ver.). • Flash sequencer supports Anti-Rollback Counter Protection using the OEM boot loader version, which is protected by the security engine.
Decryption-on-the-fly	The security engine outputs the key data of the decryption-on-the-fly IP (DOTF) through a dedicated bus
Tamper Resistance	Countermeasures available for side-channel attacks, including SPA/DPA and timing attacks.
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security and Privilege attribution can be set

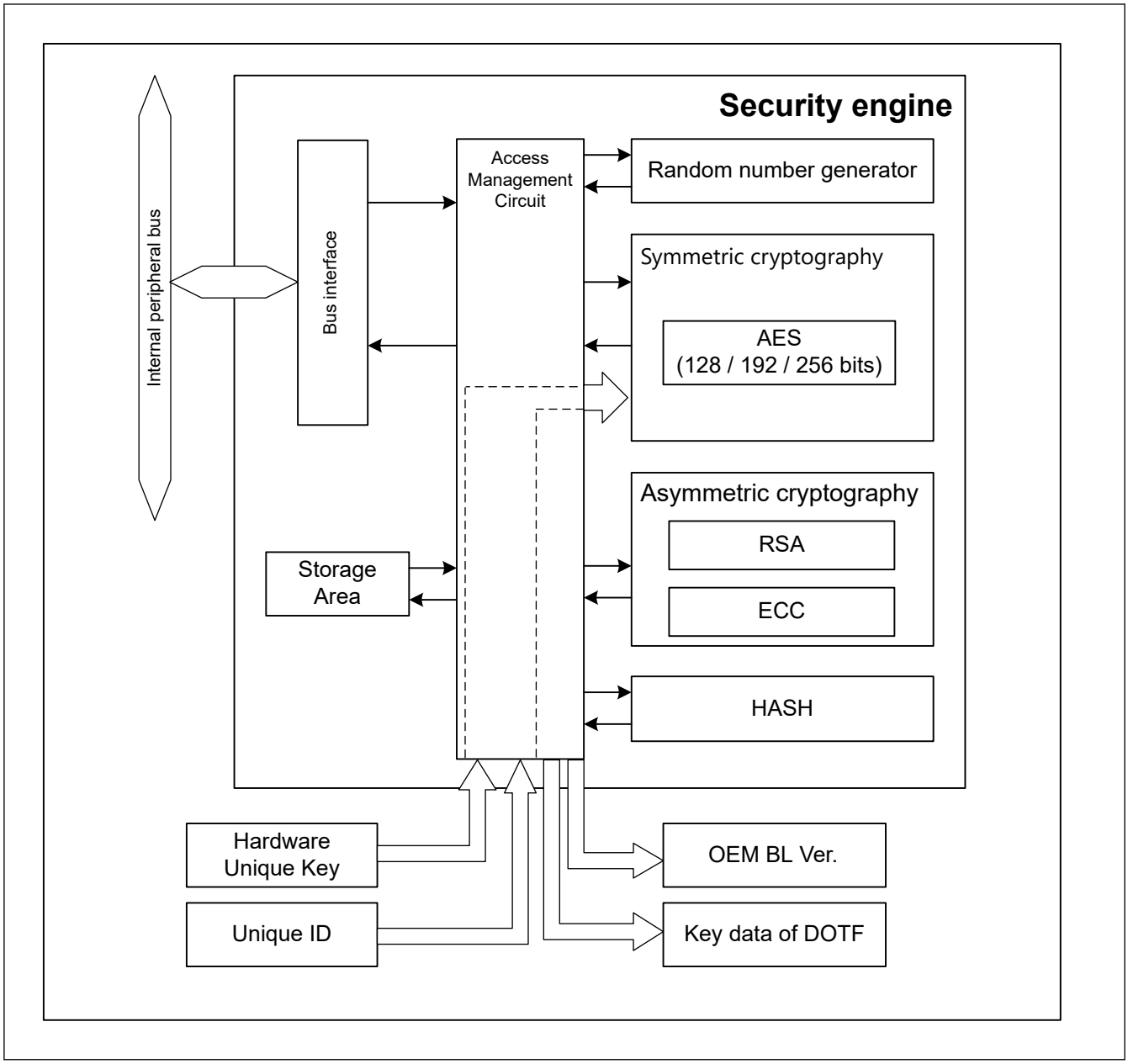


Figure 44.1 Security engine block diagram

44.2 Operation

44.2.1 Symmetric cryptography

Figure 44.2 shows conceptual block diagram of the symmetric cryptography installed in the security engine.

The symmetric cryptography uses the input key information and converts the plaintext data to ciphertext or ciphertext data to plaintext through the hardware.

The encryption/decryption process can be completed without exposing the key data and the process's intermediate data outside of the security engine. This process is performed by the encryption/decryption circuit and storage area of the security engine internally.

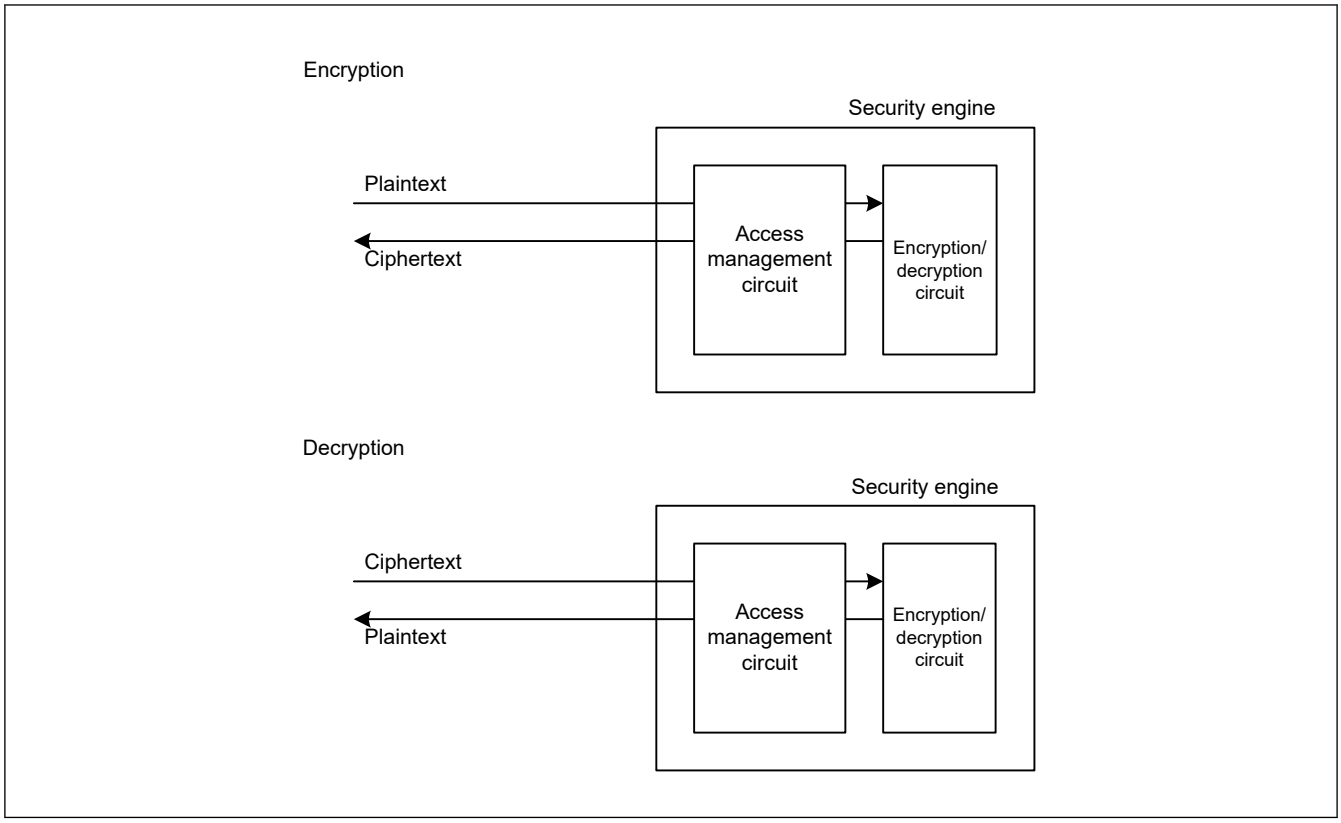


Figure 44.2 Conceptual diagram of the symmetric cryptography

44.2.2 Asymmetric Cryptography

Figure 44.3 shows conceptual block diagram of the asymmetric cryptography installed in the security engine.

The asymmetric cryptography uses the input key information and converts the plaintext data to ciphertext or ciphertext data to plaintext or generates/verifies the signature.

The encryption/decryption or signature generation/verification process can be completed without exposing the key data and the process's intermediate data outside of the security engine. This process is performed by the cryptographic circuit and storage area of the security engine internally.

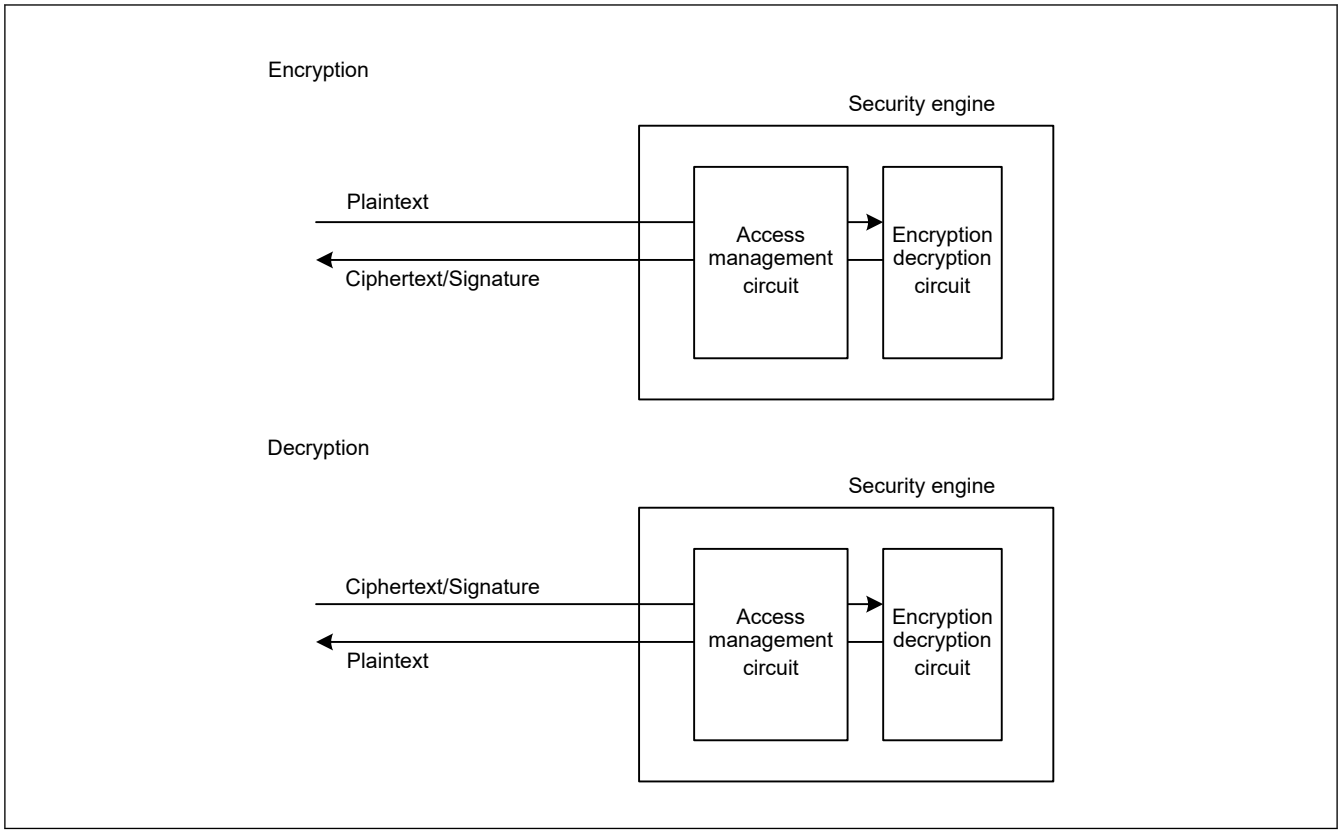


Figure 44.3 Conceptual diagram of the asymmetric cryptography

44.2.3 Hash generator

Figure 44.4 shows conceptual block diagram of the hash generator installed in the security engine.

The hash generator generates uses the input key information (if required for the hash calculation) and generates the message digest or tag.

The calculation process can be completed without exposing any key data and the process's intermediate data outside of the security engine. This process is performed by the hash circuit and storage area of the security engine internally.

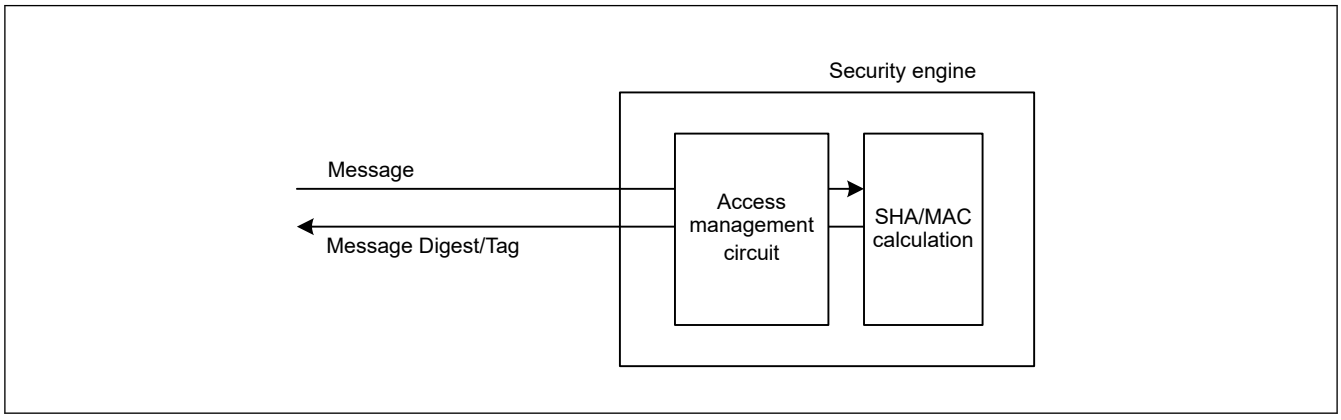


Figure 44.4 Conceptual diagram of the hash generator

44.2.4 Encryption and Decryption

Follow the procedure below to encrypt and decrypt the data:

1. Input the key information to the security engine.
2. Input the target data to the security engine. Plaintext data is converted to ciphertext and ciphertext data to plaintext.
3. Read the converted data.

The encryption/decryption circuit has input and output buffers, and can perform encryption/decryption in parallel with data input/output. Figure 44.5 shows the encryption engine timing.

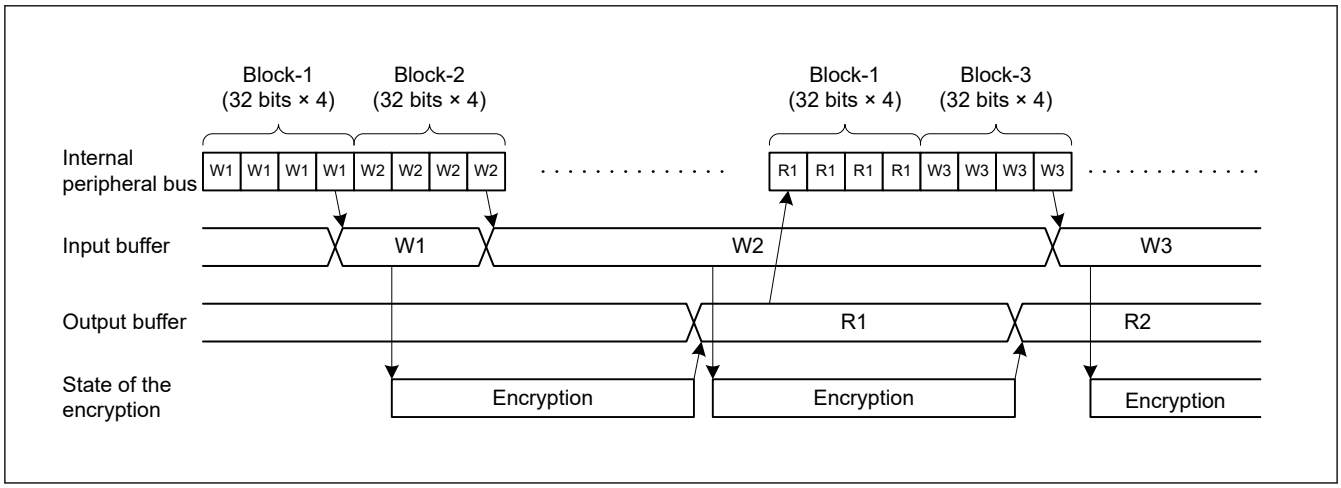


Figure 44.5 Encryption and decryption timing (AES)

44.3 Usage Notes

44.3.1 Software Standby Mode

When Software Standby mode is entered while the security engine is in process, proper processing cannot be resumed after cancelling Software Standby mode. Software Standby mode should therefore be entered while the security engine is not running.

44.3.2 Module-Stop Function Setting

The security engine operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The security engine module is initially stopped after reset. Releasing the module-stop state enables access to the registers.

44.3.3 SPA/DPA Protections

SPA/DPA protections can optionally be enabled for encryption/decryption processing.

44.3.4 Restrictions on Open process

While open process of security engine's API (R_RSIP_Open) is running, the following restrictions is applied.

- Available CPU frequency is up to 240MHz
- Do not change frequency of CPUCLK, ICLK and PCLKn (n = A to E)
- Do not change MSTPCRn registers (n = A to E)
- Do not transition to CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode and Deep Software Standby mode 1/2/3."

45. 12-Bit A/D Converter (ADC12)

45.1 Overview

The MCU includes 12-bit successive approximation A/D converter (ADC12) units. In unit 0, up to 12 analog input channels are selectable. In unit 1, up to 13 analog input channels, temperature sensor output, internal reference voltage, VBATT 1/3 voltage monitor output can be selected for conversion in respective units.

The A/D conversion accuracy is selectable from 12-bit, 10-bit, 8-bit conversion, making it possible to optimize the trade-off between speed and resolution in generating a digital value.

The ADC12 supports the following operating modes:

- Single scan mode to convert analog inputs of selected channels in ascending order of channel number
- Continuous scan mode to convert analog inputs of selected channels continuously in ascending order of channel number
- Group scan mode to divide analog inputs of channels into two groups (group A and B) and convert the analog inputs of selected channels for each group in ascending order of channel number.

In group scan mode, select two groups (group A and B). You can individually select the scan start conditions for each group (group A, B) and start scanning of each group at different times. In addition, when group A priority control operation is set, the ADC12 accepts group A scan start during group B A/D conversion, suspending group B conversion. This allows you to assign higher priority to A/D conversion start for group A.

In double trigger mode, the analog input of a selected channel is converted in single scan mode or group scan mode (group A), and data converted by the first and second A/D conversion start triggers are stored in different registers, providing duplexing of A/D converted data.

Self-diagnosis is performed once at the beginning of each scan, and one of the three reference voltage values generated in ADC12 is A/D converted.

The temperature sensor output and the internal reference voltage is selectable at the same time as the analog input of the channel. First A/D conversion is performed for the analog input of the channel, next the temperature sensor output and then for the internal reference voltage.

The ADC12 also provides a compare function (window A and window B). The compare function specifies the upper reference value for window A and lower reference value for window B, and outputs an interrupt when the A/D converted value of the selected channel meets the comparison conditions.

The A/D data storage buffer is a ring buffer consisting of 16 buffers to sequentially store A/D converted data.

Table 45.1 lists the ADC12 specifications and Table 45.2 lists the functions. Figure 45.1 shows a block diagram of ADC12 and Table 45.3 lists the I/O pins.

Table 45.1 ADC12 specifications (1 of 3)

Parameter	Specifications
Number of units	Two units
Input channels	Up to 25 channels (AN000 to AN002, AN004 to AN008, AN016 to AN019, AN100 to AN102, AN104 to AN106, AN116 to AN122)
Analog function	Temperature sensor output, internal reference voltage, VBATT 1/3 voltage monitor output
Conversion method	Successive approximation method
Resolution	12-bit, 10-bit, 8-bit
Conversion time	0.4 µs/channel (when 12-bit A/D conversion clock PCLKC (ADCLK) is operating at 60 MHz)
A/D conversion clock	Peripheral module clock PCLKA and A/D conversion clock PCLKC (ADCLK) can be set with the following division ratios: PCLKA to PCLKC (ADCLK) frequency ratio = 1:1, 2:1, 4:1, 8:1, 1:2, 1:4

Table 45.1 ADC12 specifications (2 of 3)

Parameter	Specifications
Data registers[*1]	• 25 registers for analog input • One register for A/D-converted data duplication in double trigger mode • Two registers for A/D-converted data duplication during extended operation in double trigger mode • One register for temperature sensor output • One register for internal reference voltage • One register for VBATT 1/3 voltage monitor output • One register for self-diagnosis • A/D conversion results are stored in A/D data registers • 12-bit, 10-bit, 8-bit accuracy for A/D conversion results • A/D-converted value addition mode, in which the sum of all A/D-converted results is stored in the A/D data registers as a value with the conversion accuracy bit count + extended bits • Double-trigger mode (selectable in single scan and group scan modes): – The first unit of A/D-converted analog input data on one selected channel is stored in the data register for the channel, and the second unit is stored in the duplication register. • Extended operation in double trigger mode (available for specific triggers): – A/D-converted analog input data on one selected channel is stored in the duplication register provided for the associated trigger.
Operating modes[*2]	• Single scan mode: – A/D conversion is performed only once on the analog inputs of arbitrarily selected channels, on the temperature sensor output, on the internal reference voltage. • Continuous scan mode: A/D conversion is performed repeatedly on the analog inputs of the selected channels on the temperature sensor output, and on the internal reference voltage. • Group scan mode: – Analog inputs of selected channels, the temperature sensor output, and the internal reference voltage are divided into groups A and B. Then A/D conversion of the analog inputs selected on a group basis is performed once. – The scan start conditions can be independently selected for group A, B, allowing A/D conversion of group A, B to be started independently. • Group scan mode (when group priority operation is selected): – If a priority group trigger is input during scanning of a lower-priority group, the scanning of the lower-priority group is stopped and then scanning of the priority group is started. The order of priority is group A > group B. – It is possible to select whether to restart scanning (rescan) of the lower-priority group upon completion of the priority group scan. It is also possible to specify rescanning to be started from the first channel of the selected channels or from the channel for which A/D conversion has not been completed.
Conditions for A/D conversion start	• Software trigger • Synchronous triggers from the Event Link Controller (ELC) and GPT • Asynchronous triggering by the external trigger pins, ADTRG0 (unit 0) and ADTRG1 (unit 1)
Functions	• Dedicated sample-and-hold function with optional constant sampling and 3 channels in unit 0 • Variable sampling state count • Self-diagnosis of A/D converter • Selectable A/D-converted value addition mode or average mode • Analog input disconnection detection function (discharge and precharge functions) • Double-trigger mode (duplication of A/D conversion data) • Automatic clear function for A/D data registers • Digital comparison of values in the comparison register and data register, and comparison between values in the data registers • Ring buffer

Table 45.1 ADC12 specifications (3 of 3)

Parameter	Specifications
Interrupt sources	• In single scan mode (double trigger deselected), an A/D scan end interrupt request (ADC12i_ADI (i = 0, 1)) and ELC event signal (ADC12i_ADI (i = 0, 1)) can be generated on completion of single scan. – A compare interrupt request (ADC12i_CMPAI (i = 0, 1)/ADC12i_CMPBI (i = 0, 1)) can be generated in response to a match with a digital comparison condition. – A window compare ELC event signal (ADC12i_WCMPM (i = 0, 1)) can be generated in response to a match with a digital comparison condition. – A window compare ELC event signal (ADC12i_WCMPUM (i = 0, 1)) can be generated in response to a mismatch with a digital comparison condition. • In single scan mode (double trigger selected), an A/D scan end interrupt request (ADC12i_ADI (i = 0, 1)) and ELC event signal (ADC12i_ADI (i = 0, 1)) is generated on completion of two scans. • In continuous scan mode, an A/D scan end interrupt request (ADC12i_ADI (i = 0, 1)) and ELC event signal (ADC12i_ADI (i = 0, 1)) is generated on completion of all the selected channel scans. • In group scan mode (double trigger deselected), an A/D scan end interrupt request (ADC12i_ADI (i = 0,1)) and ELC event signal (ADC12i_ADI (i = 0, 1)) is generated on completion of group A scan, and an A/D scan end interrupt request for group B (ADC12i_GBADI (i = 0, 1)) can be generated on completion of group B scan. • In group scan mode (double trigger selected), an A/D scan end interrupt request (ADC12i_ADI (i = 0, 1)) and ELC event signal (ADC12i_ADI (i = 0, 1)) is generated on completion of two group A scans, and an A/D scan end interrupt request for group B (ADC12i_GBADI (i = 0, 1)) can be generated on completion of group B scan. • ADC12i_ADI (i = 0, 1), ADC12i_GBADI (i = 0, 1), ADC12i_WCMPM (i = 0, 1), and ADC12i_WCMPUM (i = 0, 1) can activate the Data Transfer Controller (DTC).
ELC interface	• An event is generated upon completion of group A scan in group-scan mode. • An event is generated upon completion of group B scan in group-scan mode. • An event is generated when all scans complete. • Scan can be started by a trigger from the ELC. • An event is generated according to conditions of the compare function window in single-scan mode.
Reference voltage	• Unit 0: VREFH0 is the analog reference voltage. VREFL0 is the analog reference ground. • Unit 1: VREFH is the analog reference voltage. VREFL is the analog reference ground.
Analog channel input SA judgement function	Input channels of analog can be valid only when the security attribution of ADC12n (n = 0, 1) match the security attribution of PORT (PmSAR). When the security attribution of ADC12n (n = 0, 1) does not match the PORT security attribution, the conversion result of ADC input channel will be unexpected. Internal reference voltage, Temperature sensor output and VBATT 1/3 voltage monitor output have no Security Attribution judgement function.
Module-stop function	Module-stop state can be set to for each units reduce power consumption.*3
TrustZone Filter	Security and Privilege attribution can be set for each unit

Note 1. Changing the A/D conversion accuracy also changes the A/D conversion time. For details, see section 45.3.6. Analog Input Sampling and Scan Conversion Time.
Note 2. When selecting the temperature sensor output, the internal reference voltage, do not use continuous scan mode or group scan mode.
Note 3. For details, see section 10, Low Power Modes.

Table 45.2 ADC12 functions (1 of 2)

Parameter	Function
Analog input channel	AN000 to AN002, AN004 to AN008, AN016 to AN019 (unit 0), AN100 to AN102, AN104 to AN106, AN116 to AN122 (unit 1) Internal reference voltage Temperature sensor output VBATT 1/3 voltage monitor output

Table 45.2 ADC12 functions (2 of 2)

Parameter			Function
Conditions for A/D conversion starts	Software	Software trigger	Enabled
	Asynchronous trigger (external trigger)	Trigger input pin	ADTRGn (n = 0, 1)
	Synchronous trigger (trigger from ELC)	ELC trigger	ELC_AD00 (unit 0) and ELC_AD10 (unit 1), ELC_AD01 (unit 0) and ELC_AD11 (unit 1)
		GPT trigger	GTCIADAm, GTCIADBm (unit 0) (m = 0 to 3) GTCIADAn, GTCIADBn (unit 1) (n = 4 to 7)
Channel-dedicated sample-and-hold function Target channel			AN000 to AN002 (unit 0)
Interrupt			ADC12i_ADI (i = 0, 1) ADC12i_GBADI (i = 0, 1) ADC12i_CMPAI (i = 0, 1) ADC12i_CMPBI (i = 0, 1)
Output to ELC			ADC12i_ADI (i = 0, 1) ADC12i_WCMPM (i = 0, 1) ADC12i_WCMPUM (i = 0, 1)
Module-stop function settings*1 *2			MSTPCRD.MSTPD16 bit (unit 0) MSTPCRD.MSTPD15 bit (unit 1)

Note 1. For details, see section 10, Low Power Modes.
Note 2. Wait 1 µs or longer to start A/D conversion after release from the module-stop state.

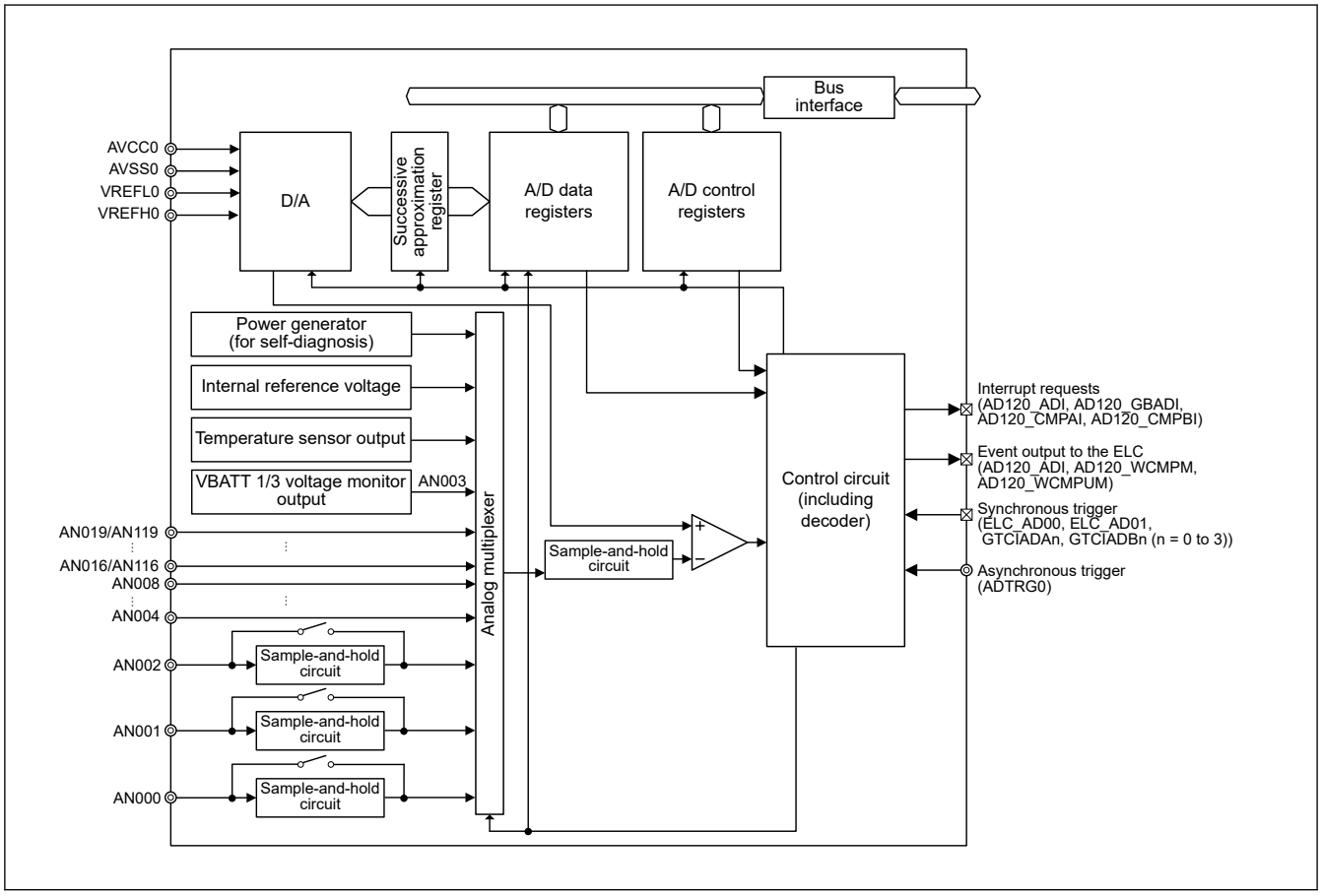


Figure 45.1 ADC12 block diagram (unit 0)

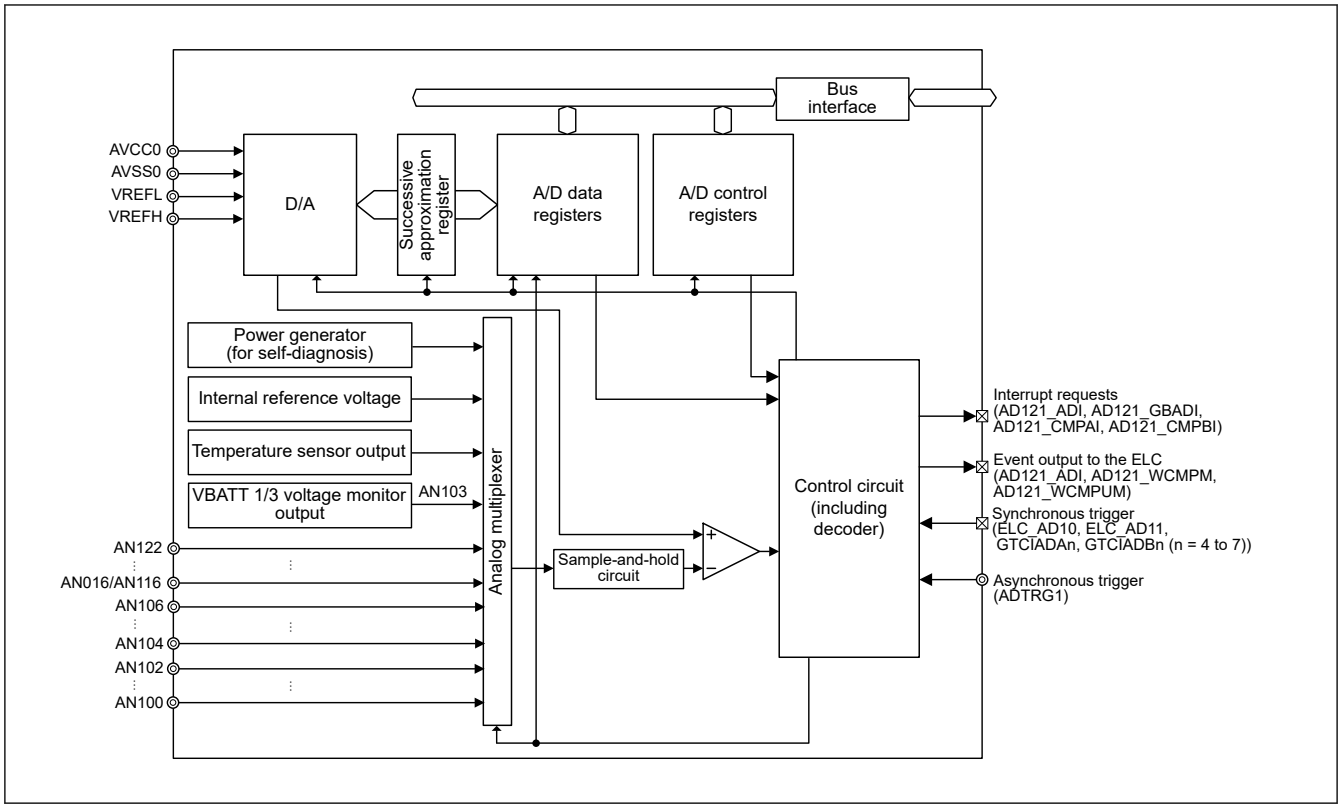


Figure 45.2 ADC12 block diagram (unit 1)

Table 45.3 lists the ADC12 I/O pins.

Table 45.3 ADC12 I/O pins (unit 0)

Pin name	I/O	Function
AVCC0	Input	Analog block power supply pin (Connect to VCC when ADC12/DAC12 is not used.)
AVSS0	Input	Analog block power supply ground pin (Connect to VSS when ADC12/DAC12 is not used.)
VREFH0	Input	Analog reference voltage supply pin
VREFL0	Input	Analog reference ground pin
AN000 to AN002, AN004 to AN008, AN016 to AN019	Input	Analog input pins 0 to 2, 4 to 8, 16 to 19
ADTRG0	Input	External trigger input pin for starting A/D conversion

Table 45.4 ADC12 I/O pins (unit 1)

Pin name	I/O	Function
AVCC0	Input	Analog block power supply pin (Connect to VCC when ADC12/DAC12 is not used.)
AVSS0	Input	Analog block power supply ground pin (Connect to VSS when ADC12/DAC12 is not used.)
VREFH	Input	Analog reference voltage supply pin
VREFL	Input	Analog reference ground pin
AN100 to AN102, AN104 to AN106, AN116 to AN122	Input	Analog input pins 0 to 2, 4 to 6, 16 to 22
ADTRG1	Input	External trigger input pin for starting A/D conversion

45.2 Register Descriptions

45.2.1 ADDRn : A/D Data Registers n

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x020 + 0x2 × n (n = 0 to 8, 16 to 19, m = 0)
0x020 + 0x2 × n (n = 0 to 6, 16 to 22, m = 1)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADDR [15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ADDR [15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See Table 45.5 and Table 45.6.	R

Note: S-TYPE-3, P-TYPE-3

ADDRn registers are 16-bit read-only registers to store A/D conversion results.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- The setting in the A/D Conversion Accuracy Select bits (ADCER.ADPRC[1:0]) (12-bit, 10-bit, 8-bit is selectable.)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 3, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

Table 45.5 shows the example of bit assignment for 12-bit accuracy.

Table 45.5 Example of bit assignment for 12-bit accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, these registers indicate the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 12-bit, 10-bit, 8-bit accuracy, 1, 2, 3, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, these registers indicate the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

Table 45.6 shows example of the bit assignment for 12-bit accuracy.

Table 45.6 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	These bits are read as 0.		Added Value 13 to 0: 14-bit sum of A/D conversion results													
Left-justified data with 12-bit accuracy	When 1, 2, 3, or 4 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 16 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

45.2.2 ADDBLDR : A/D Data Duplexing Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x018

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADDBLDR [15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ADDBLDR [15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See Table 45.7 and Table 45.8.	R

Note: S-TYPE-3, P-TYPE-3

ADDBLDR register is a 16-bit read-only register to store A/D conversion results in response to a second trigger in double-trigger mode.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- The setting in the A/D Conversion Accuracy Select bits (ADCER.ADPRC[1:0]) (12-bit, 10-bit, 8-bit is selectable.)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 3, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

Table 45.7 shows the example of bit assignment for 12-bit accuracy.

Table 45.7 Example of bit assignment for 12-bit accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, this register indicates the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 12-bit, 10-bit, 8-bit accuracy, 1, 2, 3, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, this register indicates the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

Table 45.8 shows example of the bit assignment for 12-bit accuracy.

Table 45.8 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	These bits are read as 0.		Added Value 13 to 0: 14-bit sum of A/D conversion results													
Left-justified data with 12-bit accuracy	When 1, 2, 3, or 4 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 16 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

45.2.3 ADDBLDRn : A/D Data Duplexing Register n (n = A, B)

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x084 (n = A)
0x086 (n = B)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADDBLDR [15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ADDBLDR [15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See Table 45.9 and Table 45.10.	R

Note: S-TYPE-3, P-TYPE-3

ADDBLDRn registers are 16-bit read-only registers to store A/D conversion results in response to respective triggers during extended operation in double-trigger mode.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- The setting in the A/D Conversion Accuracy Select bits (ADCER.ADPRC[1:0]) (12-bit, 10-bit, 8-bit is selectable.)

- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 3, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

Table 45.9 shows the example of bit assignment for 12-bit accuracy.

Table 45.9 Example of bit assignment for 12-bit accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, these registers indicate the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 12-bit, 10-bit, 8-bit accuracy, 1, 2, 3, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, these registers indicate the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

Table 45.10 shows example of the bit assignment for 12-bit accuracy.

Table 45.10 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	These bits are read as 0.		Added Value 13 to 0: 14-bit sum of A/D conversion results													
Left-justified data with 12-bit accuracy	When 1, 2, 3, or 4 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 16 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

45.2.4 ADTSDR : A/D Temperature Sensor Data Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x01A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADTSDR [15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ADTSDR [15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See Table 45.11 and Table 45.12.	R

Note: S-TYPE-3, P-TYPE-3

ADTSDR register is a 16-bit read-only register to store A/D conversion result of the temperature sensor output.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- The setting in the A/D Conversion Accuracy Select bits (ADCER.ADPRC[1:0]) (12-bit, 10-bit, 8-bit is selectable.)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 3, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

Table 45.11 shows the example of bit assignment for 12-bit accuracy.

Table 45.11 Example of bit assignment for 12-bit accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, this register indicates the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 12-bit, 10-bit, 8-bit accuracy, 1, 2, 3, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, this register indicates the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

Table 45.12 shows example of the bit assignment for 12-bit accuracy.

Table 45.12 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	These bits are read as 0.	Added Value 13 to 0: 14-bit sum of A/D conversion results														
Left-justified data with 12-bit accuracy	When 1, 2, 3, or 4 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 16 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

45.2.5 ADOCDR : A/D Internal Reference Voltage Data Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x01C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADOCDR [15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ADOCDR [15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See Table 45.13 and Table 45.14.	R

Note: S-TYPE-3, P-TYPE-3

ADOCDR register is a 16-bit read-only register to store A/D conversion result of the internal reference voltage.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- The setting in the A/D Conversion Accuracy Select bits (ADCER.ADPRC[1:0]) (12-bit, 10-bit, 8-bit is selectable.)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 3, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

Table 45.13 shows the example of bit assignment for 12-bit accuracy.

Table 45.13 Example of bit assignment for 12-bit accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, this register indicates the mean of A/D-converted values on a

specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 12-bit, 10-bit, 8-bit accuracy, 1, 2, 3, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, this register indicates the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

Table 45.14 shows example of the bit assignment for 12-bit accuracy.

Table 45.14 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	These bits are read as 0.		Added Value 13 to 0: 14-bit sum of A/D conversion results													
Left-justified data with 12-bit accuracy	When 1, 2, 3, or 4 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 16 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

45.2.6 ADVMDR : A/D VBATT Monitor Data Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x026

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADDR [15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ADDR[15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See Table 45.15 and Table 45.16	R

Note: S-TYPE-3, P-TYPE-3

ADVMDR register is a 16-bit read-only register to store A/D conversion result of the VBATT 1/3 voltage. For VBATT 1/3 voltage measurement, VBATTMNSELR.VBTMNSEL need to be set. For more information, seesection 11, Battery Backup Function. ADVMDR register is an alias for the ADDR3 register.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- The setting in the A/D Conversion Accuracy Select bits (ADCER.ADPRC[1:0]) (12-bit, 10-bit, 8-bit is selectable.)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 3, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

Table 45.15 shows the example of bit assignment for 12-bit accuracy.

Table 45.15 Example of bit assignment for 12-bit accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, this register indicates the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 12-bit, 10-bit, 8-bit accuracy, 1, 2, 3, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, this register indicates the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

Table 45.16 shows example of the bit assignment for 12-bit accuracy.

Table 45.16 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	These bits are read as 0.	Added Value 13 to 0: 14-bit sum of A/D conversion results														
Left-justified data with 12-bit accuracy	When 1, 2, 3, or 4 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 16 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

45.2.7 ADRD : A/D Self-Diagnosis Data Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x01E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DIAGST[1:0]		—	—	AD[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	AD[11:0]	Converted Value 11 to 0 12-bit A/D-converted value	R
13:12	—	These bits are read as 0.	R
15:14	DIAGST[1:0]	Self-Diagnosis Status For details on self-diagnosis, see section 45.2.16. ADCER : A/D Control Extended Register. 0 0: Self-diagnosis not executed after power-on. 0 1: Self-diagnosis was executed using the 0 V voltage. 1 0: Self-diagnosis was executed using the reference voltage*1 × 1/2. 1 1: Self-diagnosis was executed using the reference voltage *1.	R

Note: S-TYPE-3, P-TYPE-3
Note: The example of the bit assignment for the right-justified data with 12-bit accuracy is indicated.
Note 1. The reference voltage refers to VREFH0 for unit 0 and to VREFH for unit 1.

ADRD is a 16-bit read-only register that holds the A/D conversion results based on the self-diagnosis of the ADC12. In addition to the AD[11:0] bits indicating the A/D-converted value, it includes the Self-Diagnosis Status bit (DIAGST[1:0]).

The settings of the A/D data register format and the A/D conversion accuracy determines the formats for data in this register.

The A/D-converted value addition and average modes cannot be applied to the A/D self-diagnosis function. For details on self-diagnosis, see section 45.2.16. ADCER : A/D Control Extended Register.

This section describes the data formats for each condition. The register diagram and the register bit table shown in this section indicate example of the bit assignment for the left and right-justified data with 12-bit accuracy.

Table 45.17 Bit assignment for each right-justified accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	DIAGST[1:0]		—		AD[11:0]											

Table 45.18 Bit assignment for each left-justified accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Left-justified data with 12-bit accuracy	AD[11:0]												—		DIAGST[1:0]	

45.2.8 ADCSR : A/D Control Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x000

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADST	ADCS[1:0]		—	—	—	TRGE	EXTRG	DBLE	GBADIE	—	DBLANS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	DBLANS[4:0]	Double Trigger Channel Select These bits select one analog input channel for double-trigger operation. The setting is only valid in double-trigger mode.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	GBADIE	Group B Scan End Interrupt and ELC Event Enable Group B scan only works in group scan mode. 0: Disable ADC12i_GBADI (i = 0, 1) interrupt generation on group B scan completion. 1: Enable ADC12i_GBADI (i = 0, 1) interrupt generation on group B scan completion.	R/W

Bit	Symbol	Function	R/W
7	DBLE	Double Trigger Mode Select 0: Deselect double-trigger mode. 1: Select double-trigger mode.	R/W
8	EXTRG	Trigger Select*1 0: Start A/D conversion by the synchronous trigger (GPT, ELC). 1: Start A/D conversion by the asynchronous trigger (ADTRGn*2).	R/W
9	TRGE	Trigger Start Enable 0: Disable A/D conversion to be started by the synchronous or asynchronous trigger 1: Enable A/D conversion to be started by the synchronous or asynchronous trigger	R/W
12:10	—	These bits are read as 0. The write value should be 0.	R/W
14:13	ADCS[1:0]	Scan Mode Select 0 0: Single scan mode 0 1: Group scan mode 1 0: Continuous scan mode 1 1: Setting prohibited	R/W
15	ADST	A/D Conversion Start 0: Stop A/D conversion process. 1: Start A/D conversion process.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. To start A/D conversion using an external pin (asynchronous trigger):
After a high-level signal is input to the external pin (ADTRGn), write 1 to both the TRGE and EXTRG bits in the ADCSR register and drive the ADTRGn pin low. With these settings, the scan conversion process starts on detection of the falling edge of ADTRGn. The pulse width of the low-level input must be at least PCLKA 1.5 clock cycles.
Note 2. n = 0 (unit 0), 1 (unit 1)

The ADCSR register sets double-trigger mode and A/D conversion start trigger, enables or disables scan end interrupt, selects the scan mode, and starts or stops A/D conversion.

DBLANS[4:0] bits (Double Trigger Channel Select)

The DBLANS[4:0] bits select one channel for A/D conversion data duplication in double-trigger mode. This can be selected by setting the binary value of the channel number to be duplicated. The A/D conversion results of the analog input of the channel selected in the DBLANS[4:0] bits are stored in A/D Data Register y when conversion is started by the first trigger, and stored in the A/D Data Duplexing Register when conversion is started by the second trigger.

In double-trigger mode, the channels selected in the ADANSA0 and ADANSA1 registers, are invalid, and the channel selected in the DBLANS[4:0] bits is A/D converted instead.

When double-trigger mode is used in group scan mode, double-trigger control is only applied to group A and not to group B.

Also, in double trigger mode, the analog inputs of multiple channels, temperature sensor outputs, or internal reference voltage cannot be selected for group A, but can be selected for groups B.

Only set the DBLANS[4:0] bits when the ADST bit is 0. Do not set the DBLANS[4:0] bits at the same time that you write 1 to the ADST bit.

To enter A/D-converted value addition/average mode when in double-trigger mode, set the same channel in the ADADS0 and ADADS1 registers as the channel selected in the DBLANS[4:0] bits.

A/D-converted data from the self-diagnosis function temperature sensor output and internal reference voltage cannot be used in double-trigger mode.

GBADIE bit (Group B Scan End Interrupt and ELC Event Enable)

The GBADIE bit enables or disables group B scan end interrupt (ADC12i_GBADI (i = 0, 1)) in group scan mode.

DBLE bit (Double Trigger Mode Select)

The DBLE bit selects or deselects double-trigger mode. Double-trigger mode can only be operated by the synchronous trigger (GPT, ELC) selected in the ADSTRGR.TRSA[5:0] bits.

Double-trigger operation is as follows:

1. The ADC12i_ADI (i = 0, 1) interrupt is not output on completion of the first conversion but on completion of the second conversion.
2. The A/D conversion results from the duplication channel (selected in DBLANS[4:0]) started by the first trigger are stored in A/D Data Register y and those started by the second trigger are stored in the A/D Data Duplexing Register.

When the DBLE bit is set (double-trigger mode is selected), the channels specified in the ADANSA0 and ADANSA1 registers are invalid. Double-trigger mode is deselected by setting DBLE to 0. Setting DBLE to 1 again enables the same double-trigger operation described in 1. and 2. for first time scanning with the first trigger.

Do not select double-trigger mode in continuous scan mode. When using double trigger mode in group scan mode, A/D conversion of the temperature sensor output and internal reference voltage should not be selected for group A. Software triggering cannot be used in double-trigger mode. Always set the ADST bit to 0 before setting the DBLE bit. Do not set the DBLE bit at that same time as writing 1 to the ADST bit.

EXTRG bit (Trigger Select)

The EXTRG bit selects the synchronous or asynchronous trigger as the trigger for starting A/D conversion.

In group scan mode, the setting of this bit takes effect on the trigger selected for group A. For group B, A/D conversion is started by the selected synchronous trigger regardless of this bit setting.

TRGE bit (Trigger Start Enable)

The TRGE bit enables or disables A/D conversion by the synchronous and asynchronous triggers. In group scan mode, set this bit to 1.

ADCS[1:0] bits (Scan Mode Select)

The ADCS[1:0] bits select the scan mode.

In single scan mode, A/D conversion is performed for the analog inputs of the channels selected in the ADANSA0 and ADANSA1 registers, in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, the scan conversion stops.

In continuous scan mode, when the ADCSR.ADST bit is 1, A/D conversion is performed for the analog inputs of the channels selected with the ADANSA0 and ADANSA1 registers , in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, A/D conversion repeats from the first channel. If the ADCSR.ADST bit is set to 0 during continuous scan, A/D conversion stops even if scanning is in progress.

In group scan mode:

- Group A scanning is started by the synchronous trigger (GPT, ELC) selected in the TRSA[5:0] bits in the ADSTRGR register. A/D conversion is performed on group A analog inputs of the channels selected in the ADANSA0 and ADANSA1 registers, in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, A/D conversion stops.
- Group B scanning is started by the synchronous trigger (GPT, ELC) selected in the ADSTRGR.TRSB[5:0] bits. A/D conversion is performed on group B analog inputs of the channels selected in the ADANSB0 and ADANSB1 registers, in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, A/D conversion stops.

If the conversion processes in group A and group B occur at the same time, those conversions cannot be controlled separately. In this case, set group A Priority Control Setting bit (ADGSPCR.PGS) in the A/D Group Scan Priority Control Register (ADGSPCR) to 1 to assign a priority to group A conversion.

In group scan mode, select different channels and triggers for group A and group B.

Only set the ADCS[1:0] bits when the ADST bit is 0. Do not set the ADCS[1:0] bits at the same time that you write 1 to the ADST bit.

Table 45.19 Selectable targets for A/D conversion depending on scan and double-trigger mode settings

Scan mode setting	Double-trigger mode setting	Targets for A/D conversion				
		Self-diagnosis	Analog input (group A)	Analog input (group B)	Temperature sensor output	Internal reference voltage
Single scan	DBLE = 0	✓	✓	—	✓	✓
	DBLE = 1	—	✓ (1 ch only)	—	—	—
Continuous scan	DBLE = 0	✓	✓	—	✓	✓
	DBLE = 1	—	—	—	—	—
Group scan	DBLE = 0	✓	✓	✓	✓	✓
	DBLE = 1	—	✓ (1 ch only)	✓	✓	✓

Note: ✓: Selectable, —: Not selectable

ADST bit (A/D Conversion Start)

The ADST bit starts or stops the A/D conversion process. Before the ADST bit is set to 1, set the A/D conversion clock, the conversion mode, and the conversion target analog input.

[Setting conditions]

- 1 is written.
- The synchronous trigger (GPT, ELC) selected in the ADSTRGR.TRSA[5:0] bits is detected when ADCSR.EXTRG is 0 and ADCSR.TRGE is 1.
- The synchronous trigger (GPT, ELC) selected in the ADSTRGR.TRSB[5:0] bits is detected when ADCSR.TRGE is set to 1 in group scan mode.
- The asynchronous trigger is detected when the ADCSR.TRGE and ADCSR.EXTRG bits are set to 1 and the ADSTRGR.TRSA[5:0] bits are set to 0x00.
- When group priority operation mode is enabled (ADCSR.ADCS[1:0] = 01b and ADGSPCR.PGS = 1), the ADGSPCR.GBRP bit is set to 1, and each time A/D conversion on the group with the lowest priority is started.

[Clearing conditions]

- 0 is written.
- The A/D conversion of all the selected channels, the temperature sensor output the internal reference voltage completes in single scan mode.
- Group A scan completes in group scan mode.
- Group B scan completes in group scan mode.
- When group priority operation mode is enabled (ADCSR.ADCS[1:0] = 01b and ADGSPCR.PGS = 1), the ADGSPCR.GBRSCN bit is set to 1, and A/D conversion on the group with the lowest priority started by trigger completes.

Note: When group priority operation mode is enabled (ADCSR.ADCS[1:0] = 01b and ADGSPCR.PGS = 1), do not set the ADST bit to 1.

Note: When group priority operation mode is enabled (ADCSR.ADCS[1:0] = 01b and ADGSPCR.PGS = 1), do not set the ADST bit to 0. When forcing A/D conversion to terminate, follow the procedure for clearing the ADST bit.

45.2.9 ADANSA0 : A/D Channel Select Register A0

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x004

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ANSA 15	ANSA 14	ANSA 13	ANSA 12	ANSA 11	ANSA 10	ANSA 09	ANSA 08	ANSA 07	ANSA 06	ANSA 05	ANSA 04	ANSA 03	ANSA 02	ANSA 01	ANSA 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ANSA15 to ANSA00*1	A/D Conversion Channels Select Bit 15 (ANSA15) is associated with ANm15 and bit 0 (ANSA00) is associated with ANm00. 0: Do not select associated input channel. 1: Select associated input channel.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.
Note 1. 00 to 08 (unit 0), 00 to 06 (unit 1) is available.

ADANSA0 register selects analog input channels for A/D conversion. In group scan mode, this register selects group A channels.

Only set the ADANSA0 register when the ADCSR.ADST bit is 0.

ANSAn bits (A/D Conversion Channels Select)

The ADANSA0 register selects any combination of analog input channels for A/D conversion. The channels and the number of channels can be arbitrarily set.

In double trigger mode, the channels selected in the ADANSA0 register are invalid, and the channel selected in the ADCSR.DBLANS[4:0] bits is selected in group A instead.

When group scan mode is selected, do not select the channels specified in A/D Channel Select Register B0 (ADANSB0) and A/D Channel Select Register B1 (ADANSB1).

VBATT 1/3 voltage monitor output correspond to ADC120.ADANSA0.ANSA03 or ADC121.ADANSA0.ANSA03 bit.

Security attribution function between ADC12n (n = 0, 1) and PORT. Detail conversion result please see section 45.3.12. Security Attribution between ADCn (n = 0, 1) and PORT (PmSAR (m = 0 to 9, A, B)).

45.2.10 ADANSA1 : A/D Channel Select Register A1

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x006

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ANSA 31	ANSA 30	ANSA 29	ANSA 28	ANSA 27	ANSA 26	ANSA 25	ANSA 24	ANSA 23	ANSA 22	ANSA 21	ANSA 20	ANSA 19	ANSA 18	ANSA 17	ANSA 16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ANSA31 to ANSA16*1	A/D Conversion Channels Select Bit 15 (ANSA31) is associated with ANm31 and bit 0 (ANSA16) is associated with ANm16. 0: Do not select associated input channel. 1: Select associated input channel.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.
Note 1. 16 to 19 (unit 0), 16 to 22 (unit 1) is available.

ADANSA1 register selects analog input channels for A/D conversion. In group scan mode, this register selects group A channels.

Only set the ADANSA1 register when the ADCSR.ADST bit is 0.

ANSAn bits (A/D Conversion Channels Select)

The ADANSA1 register selects any combination of analog input channels for A/D conversion. The channels and the number of channels can be arbitrarily set.

In double trigger mode, the channels selected in the ADANSA1 register are invalid, and the channel selected in the ADCSR.DBLANS[4:0] bits is selected in group A instead.

When group scan mode is selected, do not select the channels specified in A/D Channel Select Register B0 (ADANSB0) and A/D Channel Select Register B1 (ADANSB1).

Security attribution function between ADC12n (n = 0, 1) and PORT.

Detail conversion result please see section 45.3.12. Security Attribution between ADCn (n = 0, 1) and PORT (PmSAR (m = 0 to 9, A, B)).

45.2.11 ADANSB0 : A/D Channel Select Register B0

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x014

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ANSB 15	ANSB 14	ANSB 13	ANSB 12	ANSB 11	ANSB 10	ANSB 09	ANSB 08	ANSB 07	ANSB 06	ANSB 05	ANSB 04	ANSB 03	ANSB 02	ANSB 01	ANSB 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ANSB15 to ANSB00*1	A/D Conversion Channels Select Bit 15 (ANSB15) is associated with ANm15 and bit 0 (ANSB00) is associated with ANm00. 0: Do not select associated input channel. 1: Select associated input channel.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.
Note 1. 00 to 08 (unit 0), 00 to 06 (unit 1) is available.

ADANSB0 selects analog input channels for A/D conversion in group B when group scan mode is selected. The ADANSB0 register is not used in any scan mode other than group scan mode.

Only set the ADANSB0 register when the ADCSR.ADST bit is 0.

ANSBn bits (A/D Conversion Channels Select)

The ADANSB0 register selects any combination of analog input channels in group B for A/D conversion when group scan mode is selected. The ADANSB0 register is used for group scan mode only and not for any other modes.

Do not select channels specified in group A as selected in the ADANSA0 and ADANSA1 registers or the ADCSR.DBLANS[4:0] bits in double-trigger mode.

VBATT 1/3 voltage monitor output correspond to ADC120.ADANSB0.ANSB03 or ADC121.ADANSB0.ANSB03.

Security attribution function between ADC12n (n = 0, 1) and PORT (PmSAR). Detail conversion result please see section 45.3.12. Security Attribution between ADCn (n = 0, 1) and PORT (PmSAR (m = 0 to 9, A, B)).

45.2.12 ADANSB1 : A/D Channel Select Register B1

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x016

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ANSB31	ANSB30	ANSB29	ANSB28	ANSB27	ANSB26	ANSB25	ANSB24	ANSB23	ANSB22	ANSB21	ANSB20	ANSB19	ANSB18	ANSB17	ANSB16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ANSB31 to ANSB16[*1]	A/D Conversion Channels Select Bit 15 (ANSB31) is associated with ANm31 and bit 0 (ANSB16) is associated with ANm16. 0: Do not select associated input channel. 1: Select associated input channel.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.
Note 1. 16 to 19 (unit 0), 16 to 22 (unit 1) is available.

ADANSB1 selects analog input channels for A/D conversion in group B when group scan mode is selected. The ADANSB1 register is not used in any scan mode other than group scan mode.

Only set the ADANSB1 register when the ADCSR.ADST bit is 0.

ANSBn bits (A/D Conversion Channels Select)

The ADANSB1 register selects any combination of analog input channels in group B for A/D conversion when group scan mode is selected. The ADANSB1 register is used for group scan mode only and not for any other modes.

Do not select channels specified in group A as selected in the ADANSA0 and ADANSA1 registers or the ADCSR.DBLANS[4:0] bits in double-trigger mode.

Security attribution function between ADC12n (n = 0, 1) and PORT (PmSAR). Detail conversion result please see section 45.3.12. Security Attribution between ADCn (n = 0, 1) and PORT (PmSAR (m = 0 to 9, A, B)).

45.2.13 ADADS0 : A/D-Converted Value Addition/Average Channel Select Register 0

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x008

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADS15	ADS14	ADS13	ADS12	ADS11	ADS10	ADS09	ADS08	ADS07	ADS06	ADS05	ADS04	ADS03	ADS02	ADS01	ADS00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ADS15 to ADS00[*1]	A/D-Converted Value Addition/Average Channel Select Bit 15 (ADS15) is associated with ANm15 and bit 0 (ADS00) is associated with ANm00. 0: Do not select associated input channel. 1: Select associated input channel.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.
Note 1. 00 to 08 (unit 0), 00 to 06 (unit 1) is available.

ADSn bits (A/D-Converted Value Addition/Average Channel Select)

The ADSn bits determine which A/D-converted channels are subject to A/D-converted value addition/averaging. When an ADSn bit associated with a channel selected for A/D conversion is set to 1, A/D conversion of the analog input of the respective channel is performed successively 1, 2, 3, 4, or 16 times, as specified in the ADC[2:0] bits in the ADADC register.

When the ADADC.AVEE bit is 0, the value obtained by addition is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition is stored in the A/D data register.

The ADSn bits apply only to channels that are selected for A/D conversion in:

- The ANSAn bits in the ADANSA0 register or the DBLANS[4:0] bits in the ADCSR register
- The ANSBn bits in the ADANSB0 register

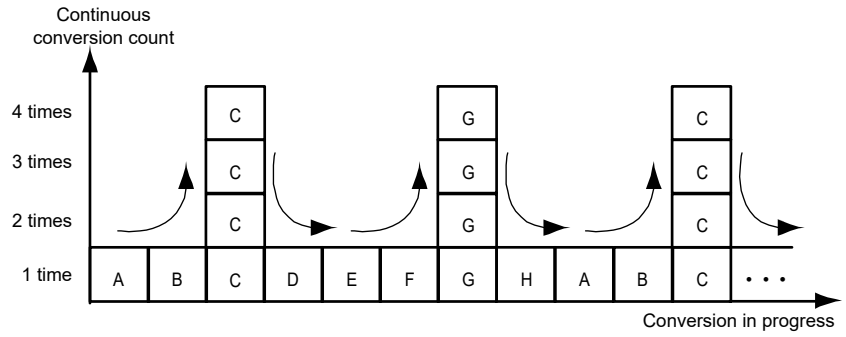
For channels on which the A/D conversion is performed and for which addition/average mode is not selected, a normal 1-time conversion is executed, and the conversion result is stored in the A/D data register.

Only set ADADS0 register bits when the ADCSR.ADST bit is 0.

Figure 45.3 shows a scanning operation sequence in which the ADADS0 register bits (channel c and g) are set to 1. In this figure:

- Addition mode is selected (ADADC.AVEE = 0)
- The number of conversions is set to 4 (ADADC.ADC[1:0] = 11b)
- The analog input channels (a to h) are selected by ADANSA0 register in continuous scan mode (ADCSR.ADCS[1:0] = 10b).

The conversion process begins with analog input A (channel a). The analog input C (channel c) conversion is performed successively 4 times and the added value is returned to A/D Data Register c (ADDRc). Next, the analog input D (channel d) conversion process is started. The analog input G (channel g) is performed successively 4 times and the added value is returned to A/D Data Register g (ADDRg). After conversion of analog input H (channel h), the conversion operation repeats in the same sequence starting with analog input A (channel a).



Note: A to H are analog inputs (channel a to h)

Figure 45.3 Scan conversion sequence with ADADC.ADC[2:0] = 011b, set 1 for analog inputs C and G by ADADS0/1

45.2.14 ADADS1 : A/D-Converted Value Addition/Average Channel Select Register 1

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x00A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADS31	ADS30	ADS29	ADS28	ADS27	ADS26	ADS25	ADS24	ADS23	ADS22	ADS21	ADS20	ADS19	ADS18	ADS17	ADS16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ADS31 to ADS16*1	A/D-Converted Value Addition/Average Channel Select Bit 15 (ADS31) is associated with ANm31 and bit 0 (ADS16) is associated with ANm16. 0: Do not select associated input channel. 1: Select associated input channel.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.
Note 1. 16 to 19 (unit 0), 16 to 22 (unit 1) is available.

ADSn bits (A/D-Converted Value Addition/Average Channel Select)

The ADSn bits determine which A/D-converted channels are subject to A/D-converted value addition/averaging. When an ADSn bit associated with a channel selected for A/D conversion is set to 1, A/D conversion of the analog input of the respective channel is performed successively 1, 2, 3, 4, or 16 times, as specified in the ADC[2:0] bits in the ADADC register.

When the ADADC.AVEE bit is 0, the value obtained by addition is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition is stored in the A/D data register.

The ADSn bits apply only to channels that are selected for A/D conversion in:

- The ANSAn bits in the ADANSA1 register or the DBLANS[4:0] bits in the ADCSR register
- The ANSBn bits in the ADANSB1 register.

For channels on which the A/D conversion is performed and for which addition/average mode is not selected, a normal 1-time conversion is executed, and the conversion result is stored in the A/D data register.

Only set ADADS1 register when the ADCSR.ADST bit is 0.

45.2.15 ADADC : A/D-Converted Value Addition/Average Count Select Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x00C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	AVEE	—	—	—	—	ADC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	ADC[2:0]	Addition/Average Count Select 0 0 0: 1-time conversion (no addition, same as normal conversion) 0 0 1: 2-time conversion (1 addition) 0 1 0: 3-time conversion (2 additions) 0 1 1: 4-time conversion (3 additions) 1 0 1: 16-time conversion (15 additions) Others: Setting prohibited	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	AVEE	Average Mode Select 0: Enable addition mode 1: Enable average mode	R/W

Note: S-TYPE-3, P-TYPE-3

ADADC sets the addition or average mode and addition count for A/D conversion. Table 45.20 lists the settable combinations of ADADC register.

Table 45.20 Settable combinations of ADADC register

Mode select	Resolution	Conversion time				
		1-time	2-time	3-time	4-time	16-time
Addition mode (AVEE = 0)	8-bit	✓	✓	✓	✓	—
	10-bit	✓	✓	✓	✓	—
	12-bit	✓	✓	✓	✓	✓
Average mode (AVEE = 1)	8, 10, 12 bits	—	✓	—	✓	—

Note: ✓: Selectable, —: Not selectable

ADC[2:0] bits (Addition/Average Count Select)

The ADC[2:0] bits set the addition count in all channels for which A/D conversion and addition/average mode are selected, including the channel selected in double trigger mode with the ADCSR.DBLANS[4:0] bits. The count also applies to A/D conversion of the temperature sensor output and the internal reference voltage.
When self-diagnosis is executed (ADCER.DIAGM = 1), do not set the ADC[2:0] bits to any value other than 000b.

AVEE bit (Average Mode Select)

The AVEE bit selects addition or average mode in all channels for which A/D conversion and addition/average mode are selected, including the channels selected in double-trigger mode in the ADCSR.DBLANS[4:0] bits, temperature sensor output, internal reference voltage.

45.2.16 ADCER : A/D Control Extended Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x00E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADRFMT	—	—	—	DIAGM	DIAGLD	DIAGVAL[1:0]		—	—	ACE	—	—	ADPRC[1:0]		—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
2:1	ADPRC[1:0]	A/D Conversion Accuracy Specify 0 0: 12-bit accuracy 0 1: 10-bit accuracy 1 0: 8-bit accuracy 1 1: Setting prohibited	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	ACE	A/D Data Register Automatic Clearing Enable 0: Disable automatic clearing 1: Enable automatic clearing	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
9:8	DIAGVAL[1:0]	Self-Diagnosis Conversion Voltage Select 0 0: Setting prohibited when self-diagnosis is enabled 0 1: 0 volts 1 0: Reference voltage*1 × 1/2 1 1: Reference voltage*1	R/W
10	DIAGLD	Self-Diagnosis Mode Select 0: Select rotation mode for self-diagnosis voltage 1: Select mixed mode for self-diagnosis voltage	R/W

Bit	Symbol	Function	R/W
11	DIAGM	Self-Diagnosis Enable 0: Disable ADC12 self-diagnosis 1: Enable ADC12 self-diagnosis	R/W
14:12	—	These bits are read as 0. The write value should be 0.	R/W
15	ADRFMT	A/D Data Register Format Select 0: Select right-justified for the A/D data register format 1: Select left-justified for the A/D data register format	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The reference voltage refers to VREFH0 for unit 0 and to VREFH for unit 1.

ADPRC[1:0] bit (A/D Conversion Accuracy Specify)

The ADPRC[1:0] bits set the A/D conversion accuracy. Changing the A/D conversion accuracy also changes the bit width of valid data stored in the result register and the A/D conversion time. For details, see section 45.3.6. Analog Input Sampling and Scan Conversion Time. Only set the ADPRC[1:0] bits while the ADCSR.ADST bit is 0.

ACE bit (A/D Data Register Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (all 0) of the ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR or ADVMDR register after any of these registers is read by the CPU or DTC. Automatic clearing of the A/D data registers enables detection of failures that are not updated in the A/D data registers. For details, see section 45.3.7. Usage Example of A/D Data Register Automatic Clearing Function.

DIAGVAL[1:0] bits (Self-Diagnosis Conversion Voltage Select)

The DIAGVAL[1:0] bits select the voltage value used in self-diagnosis fixed voltage mode. For details, see the DIAGLD bit description.

Do not execute self-diagnosis by setting the DIAGLD bit to 1 when the DIAGVAL[1:0] bits are set to 00b.

DIAGLD bit (Self-Diagnosis Mode Select)

The DIAGLD bit selects whether the three voltage values are rotated, or the fixed voltage is used in self-diagnosis.

Setting the DIAGLD bit to 0 selects conversion of the voltages in rotation mode, where 0 V, the reference voltage × 1/2, and the reference voltage are converted, in that order. After reset and when self-diagnosis voltage rotation mode is selected, self-diagnosis is executed from 0 V. The self-diagnosis voltage value does not return to 0 V when scan conversion completes. When scan conversion is restarted, rotation starts at the voltage value following the previous value.

Setting the DIAGLD bit to 1 selects fixed voltage, in which the fixed voltage specified in the ADCER.DIAGVAL[1:0] bits is converted. If fixed mode is switched to rotation mode, rotation starts at the fixed voltage value.

Only set the DIAGLD bit when the ADCSR.ADST bit is 0.

DIAGM bit (Self-Diagnosis Enable)

The DIAGM bit enables or disables self-diagnosis.

Self-diagnosis is used to detect a failure of the ADC12. In self-diagnosis mode, one of the three voltage values (0 V, the reference voltage × 1/2, or the reference voltage) is converted. When conversion completes, information on the converted voltage and the conversion result is stored into the A/D Self-Diagnosis Data Register (ADRD). The ADRD register can be read to determine whether the conversion result falls within the normal or abnormal range.

Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages is converted. In double trigger mode (ADCSR.DBLE = 1), self-diagnosis (DIAGM = 0) is deselected. When self-diagnosis is selected in group scan mode, self-diagnosis is executed separately for group A and group B.

Only set the DIAGM bit when the ADCSR.ADST bit is 0.

ADRFMT bit (A/D Data Register Format Select)

The ADRFMT bit specifies flush-right or flush-left for data to be stored in the ADDRy, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR, ADVMDR, ADCMPDR0/1, ADWINLLB, ADWINULB, or ADRD register.

Only set the ADRFMT bit when the ADCSR.ADST bit is 0.

45.2.17 ADSTRGR : A/D Conversion Start Trigger Select Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x010

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TRSA[5:0]						—	—	TRSB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	TRSB[5:0]	A/D Conversion Start Trigger Select for Group B Select the A/D conversion start trigger for group B in group scan mode.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
13:8	TRSA[5:0]	A/D Conversion Start Trigger Select Select the A/D conversion start trigger in single scan mode and continuous scan mode. In group scan mode, the A/D conversion start trigger for group A is selected.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

TRSB[5:0] bits (A/D Conversion Start Trigger Select for Group B)

The TRSB[5:0] bits select the trigger to start scanning of the analog input selected in group B. The TRSB[5:0] bits must only be set in group scan mode and are not used in any other scan mode. For the scan conversion start trigger for group B, setting a software trigger or an asynchronous trigger is prohibited. In group scan mode, set the TRSB[5:0] bits to a value other than 0x00 and set the ADCSR.TRGE bit to 1.

When group A is given priority in group scan mode, setting the ADGSPCR.GBRP bit to 1 allows group B to continuously operate in single scan mode. When setting the ADGSPCR.GBRP bit to 1, set the TRSB[5:0] bits to 0x3F. The issuance period for a conversion trigger must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than t_{SCAN}, A/D conversion by the trigger might have no effect.

When the GPT module is selected as an A/D conversion start trigger, a delay for synchronization processing occurs. For details, see section 45.3.6. Analog Input Sampling and Scan Conversion Time.

Table 45.21 lists the A/D conversion startup sources selected in the TRSB[5:0] bits.

Table 45.21 Selection of A/D conversion start sources in the TRSB[5:0] bits (1 of 2)

Source	Remarks	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
Trigger source deselected state	—	1	1	1	1	1	1
ELC_AD00 (unit 0) and ELC_AD10 (unit 1)	ELC	0	0	1	0	0	1
ELC_AD01 (unit 0) and ELC_AD11 (unit 1)	ELC	0	0	1	0	1	0
ELC_AD00 (unit 0) and ELC_AD10 (unit 1), ELC_AD01 (unit 0) and ELC_AD11 (unit 1)	ELC	0	0	1	0	1	1
GTCIADA0 (unit 0) and GTCIADA4 (unit 1)	GPT	0	1	0	0	0	1
GTCIADB0 (unit 0) and GTCIADB4 (unit 1)	GPT	0	1	0	0	1	0
GTCIADA1 (unit 0) and GTCIADA5 (unit 1)	GPT	0	1	0	0	1	1
GTCIADB1 (unit 0) and GTCIADB5 (unit 1)	GPT	0	1	0	1	0	0

Table 45.21 Selection of A/D conversion start sources in the TRSB[5:0] bits (2 of 2)

Source	Remarks	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
GTCIADA2 (unit 0) and GTCIADA6 (unit 1)	GPT	0	1	0	1	0	1
GTCIADB2 (unit 0) and GTCIADB6 (unit 1)	GPT	0	1	0	1	1	0
GTCIADA3 (unit 0) and GTCIADA7 (unit 1)	GPT	0	1	0	1	1	1
GTCIADB3 (unit 0) and GTCIADB7 (unit 1)	GPT	0	1	1	0	0	0
GTCIADA0 (unit 0) and GTCIADA4 (unit 1), GTCIADB0 (unit 0) and GTCIADB4 (unit 1)	GPT	0	1	1	0	0	1
GTCIADA1 (unit 0) and GTCIADA5 (unit 1), GTCIADB1 (unit 0) and GTCIADB5 (unit 1)	GPT	0	1	1	0	1	0
GTCIADA2 (unit 0) and GTCIADA6 (unit 1), GTCIADB2 (unit 0) and GTCIADB6 (unit 1)	GPT	0	1	1	0	1	1
GTCIADA3 (unit 0) and GTCIADA7 (unit 1), GTCIADB3 (unit 0) and GTCIADB7 (unit 1)	GPT	0	1	1	1	0	0

TRSA[5:0] bits (A/D Conversion Start Trigger Select)

The TRSA[5:0] bits select the trigger to start A/D conversion in single scan mode and continuous scan mode, or the trigger to start scanning of group A analog inputs in group scan mode. When scanning is executed in group scan mode or double trigger mode, software trigger or asynchronous trigger is prohibited.

- When using a synchronous trigger (GPT, ELC), set the TRGE bit in the ADCSR register to 1 and set the EXTRG bit in the ADCSR register to 0.
- When using the asynchronous trigger (ADTRGn (n = 0, 1), set the TRGE bit in the ADCSR register to 1 and set the EXTRG bit in the ADCSR register to 1.
- Software trigger (ADCSR.ADST) is enabled regardless of the settings of the ADCSR.TRGE bit, the ADCSR.EXTRG bit, or the TRSA[5:0] bits.

The issuance period for a conversion trigger must be more than or equal to the actual scan conversion time (tSCAN). If the issuance period is less than tSCAN, A/D conversion by a trigger might have no effect.

When the GPT module is selected as an A/D conversion start trigger, a delay for synchronization processing occurs. For details, see section 45.3.6. Analog Input Sampling and Scan Conversion Time.

Table 45.22 lists the A/D conversion start sources selected in the TRSA[5:0] bits.

Table 45.22 Selection of A/D activation sources in the TRSA[5:0] bits (1 of 2)

Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
Trigger source deselected state	—	1	1	1	1	1	1
ADTRGn (n = 0, 1)	Input pin for the trigger	0	0	0	0	0	0
ELC_AD00 (unit 0) and ELC_AD10 (unit 1)	ELC	0	0	1	0	0	1
ELC_AD01 (unit 0) and ELC_AD11 (unit 1)	ELC	0	0	1	0	1	0

Table 45.22 Selection of A/D activation sources in the TRSA[5:0] bits (2 of 2)

Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
ELC_AD00 (unit 0) and ELC_AD10 (unit 1), ELC_AD01 (unit 0) and ELC_AD11 (unit 1)	ELC	0	0	1	0	1	1
GTCIADA0 (unit 0) and GTCIADA4 (unit 1)	GPT	0	1	0	0	0	1
GTCIADB0 (unit 0) and GTCIADB4 (unit 1)	GPT	0	1	0	0	1	0
GTCIADA1 (unit 0) and GTCIADA5 (unit 1)	GPT	0	1	0	0	1	1
GTCIADB1 (unit 0) and GTCIADB5 (unit 1)	GPT	0	1	0	1	0	0
GTCIADA2 (unit 0) and GTCIADA6 (unit 1)	GPT	0	1	0	1	0	1
GTCIADB2 (unit 0) and GTCIADB6 (unit 1)	GPT	0	1	0	1	1	0
GTCIADA3 (unit 0) and GTCIADA7 (unit 1)	GPT	0	1	0	1	1	1
GTCIADB3 (unit 0) and GTCIADB7 (unit 1)	GPT	0	1	1	0	0	0
GTCIADA0 (unit 0) and GTCIADA4 (unit 1), GTCIADB0 (unit 0) and GTCIADB4 (unit 1)	GPT	0	1	1	0	0	1
GTCIADA1 (unit 0) and GTCIADA5 (unit 1), GTCIADB1 (unit 0) and GTCIADB5 (unit 1)	GPT	0	1	1	0	1	0
GTCIADA2 (unit 0) and GTCIADA6 (unit 1), GTCIADB2 (unit 0) and GTCIADB6 (unit 1)	GPT	0	1	1	0	1	1
GTCIADA3 (unit 0) and GTCIADA7 (unit 1), GTCIADB3 (unit 0) and GTCIADB7 (unit 1)	GPT	0	1	1	1	0	0

45.2.18 ADEXICR : A/D Conversion Extended Input Control Registers

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x012

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	OCSB	TSSB	OCSA	TSSA	—	—	—	—	—	—	OCSAD	TSSAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSSAD	Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select 0: Do not select addition/average mode for temperature sensor output. 1: Select addition/average mode for temperature sensor output.	R/W
1	OCSAD	Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select 0: Do not select addition/average mode for internal reference voltage. 1: Select addition/average mode for internal reference voltage.	R/W

Bit	Symbol	Function	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	TSSA	Temperature Sensor Output A/D Conversion Select 0: Disable A/D conversion of temperature sensor output 1: Enable A/D conversion of temperature sensor output	R/W
9	OCSA	Internal Reference Voltage A/D Conversion Select 0: Disable A/D conversion of internal reference voltage 1: Enable A/D conversion of internal reference voltage	R/W
10	TSSB	Temperature Sensor Output A/D Conversion Select for Group B 0: Disable A/D conversion of temperature sensor output 1: Enable A/D conversion of temperature sensor output	R/W
11	OCSB	Internal Reference Voltage A/D Conversion Select for Group B 0: Disable A/D conversion of internal reference voltage 1: Enable A/D conversion of internal reference voltage	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

TSSAD bit (Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select)

When the TSSAD bit is set to 1, A/D conversion of the temperature sensor output is selected and performed successively the number of times specified in the ADC[2:0] bits in ADADC. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is returned to the A/D Temperature Sensor Data Register (ADTSDR). When the ADADC.AVEE bit is 1, the mean value is returned to ADTSDR.

Only set the TSSAD bit while the ADCSR.ADST bit is 0.

OCSAD bit (Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select)

When the OCSAD bit is set to 1, A/D conversion of the internal reference voltage is selected and performed successively the number of times specified in the ADC[2:0] bits in ADADC. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is returned to the A/D Internal Reference Voltage Data Register (ADOCDR). When the ADADC.AVEE bit is 1, the mean value is returned to ADOCDR.

Only set the OCSAD bit while the ADCSR.ADST bit is 0.

TSSA bit (Temperature Sensor Output A/D Conversion Select)

The TSSA bit selects A/D conversion of the temperature sensor output for group A in single scan mode, continuous scan mode, or group scan mode. When A/D conversion of the temperature sensor output is selected and performed, set the ADCSR.DBLE bit to 0.

Only set the TSSA bit while the ADCSR.ADST bit is 0.

OCSA bit (Internal Reference Voltage A/D Conversion Select)

The OCSA bit selects A/D conversion of the internal reference voltage for group A in single scan mode, continuous scan mode, or group scan mode. When A/D conversion of the internal reference voltage is selected and performed, set the ADCSR.DBLE bit to 0.

Only set the OCSA bit while the ADCSR.ADST bit is 0. In addition, wait for 400 ns or more after the OCSA bit is set to 1 before starting A/D conversion.

TSSB bit (Temperature Sensor Output A/D Conversion Select for Group B)

The TSSB bit selects A/D conversion of the temperature sensor output for group B in group scan mode. Only set the TSSB bit while the ADCSR.ADST bit is 0. Do not set the TSSB bit to 1 while the TSSA bit is 1.

OCSB bit (Internal Reference Voltage A/D Conversion Select for Group B)

The OCSB bit selects A/D conversion of the internal reference voltage for group B in group scan mode. Only set the OCSB bit while the ADCSR.ADST bit is 0. Do not set the OCSB bit to 1 while the OCSA bit is 1. Moreover, start the A/D conversion after waiting for 400 ns or more after the OCSB bit is set to 1.

45.2.19 ADSSTRn/ADSSTRL/ADSSTRT/ADSSTRO/ADSSTRV: A/D Sampling State Register (n = 0 to 8)

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x0E0 + 0x1 × n (n = 0 to 8, m = 0)
0x0E0 + 0x1 × n (n = 0 to 6, m = 1)
0x0DD (ADSSTRL)
0x0DE (ADSSTRT)
0x0DF (ADSSTRO)
0x0E3 (ADSSTRV)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SST[7:0]							
Value after reset:	0	0	0	0	1	0	1	1

Bit	Symbol	Function	R/W
7:0	SST[7:0]	Sampling Time Setting These bits set the sampling time in the range from 5 to 255 states.	R/W

Note: S-TYPE-3, P-TYPE-3

The ADSSTRn register sets the sampling time for analog input.

The sampling time can be adjusted if the impedance of the analog input signal source is too high to secure sufficient sampling time, or if the ADCLK clock is slow. The set value indicates the time for one ADCLK cycle, and the required sampling time is specified by the voltage conditions. For details, see section 60.6. ADC12 Characteristics.

ADSSTRV register is an alias for the ADSSTR3 register.

The lower limit of the sampling time setting depends on the frequency ratio:

- If the frequency ratio of PCLKA to PCLKC (ADCLK) = 1:1, 2:1, 4:1, or 8:1 the sampling time must be set to a value of more than 5 states
- If the frequency ratio of PCLKA to PCLKC (ADCLK) = 1:2 or 1:4, the sampling time must be set to a value of more than 6 states.

Table 45.23 shows the relationship between the A/D Sampling State Register and the associated channels. For details, see section 45.3.6. Analog Input Sampling and Scan Conversion Time.

Only set the SST[7:0] bits when the ADCSR.ADST bit is 0.

Table 45.23 Relationship between A/D sampling state register and associated channels

Bit name	Associated channels
ADSSTRn.SST[7:0] bits (n = 0 to 8 (unit 0), 0 to 6 (unit 1))*1	AN000 to AN008, AN100 to AN106
ADSSTRL.SST[7:0] bits	AN016 to AN019, AN116 to AN122
ADSSTRT.SST[7:0] bits	Temperature sensor output
ADSSTRO.SST[7:0] bits	Internal reference voltage
ADSSTRV.SST[7:0] bits	VBATT 1/3 voltage monitor output*2

Note 1. When the self-diagnosis function is selected, the sampling time set in the ADSSTR0.SST[7:0] bits is applied.
Note 2. For the stabilization time, see section 11, Battery Backup Function.

45.2.20 ADSHCR : A/D Sample and Hold Circuit Control Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0)

Offset address: 0x066

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SHANS[2:0]			SSTSH[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

Bit	Symbol	Function	R/W
7:0	SSTSH[7:0]	Channel-Dedicated Sample-and-Hold Circuit Sampling Time Setting Sampling time (4 to 255 states).	R/W
10:8	SHANS[2:0]	Channel-Dedicated Sample-and-Hold Circuit Bypass Select Select whether to use or bypass channel-dedicated sample-and-hold circuits for AN000 to AN002 (unit 0). 0: Bypass the circuits 1: Use the circuits	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

SSTSH[7:0] bits (Channel-Dedicated Sample-and-Hold Circuit Sampling Time Setting)

The SSTSH[7:0] bits set the sampling time for the channel-dedicated sample-and-hold circuits. If one state is 1 ADCLK (A/D conversion clock) cycle and the ADCLK clock is 60 MHz, one state is 16.7 ns. The initial value is 24 states. The sampling time can be adjusted if the impedance of the analog input signal source is too high to secure sufficient sampling time, or if the ADCLK clock is slow.

Only set the SSTSH[7:0] bits while the ADCSR.ADST bit is 0. The sampling time must be set to a value that is 4 states or more and 255 or less.

SHANS[2:0] bits (Channel-Dedicated Sample-and-Hold Circuit Bypass Select)

The SHANS[2:0] bits select whether to use or bypass the channel-dedicated sample-and-hold circuits for AN000 to AN002 (unit 0). In unit 0, the SHANS[0] bit is associated with AN000, the SHANS[1] bit with AN001, and the SHANS[2] bit with AN002.

If any channel from among AN000 to AN002 (unit 0) is selected for group B while operation is in group scan mode under group A priority control, use this setting to bypass the dedicated sample-and-hold circuit of the channel.

Only set the SHANS[2:0] bits while the ADCSR.ADST bit is 0 and the ADSHMSR.SHMD bit is 0.

45.2.21 ADSHMSR : A/D Sample and Hold Operation Mode Selection Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0)

Offset address: 0x07C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SHMD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SHMD	Sampling Operation Selection 0: Sampling Operation Selection 1: Enable continuous sampling function	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

SHMD bit (Sampling Operation Selection)

Setting SHMD to 1 enables the constant sampling function of the channel-dedicated sample-and-hold selected in the ADSHCR.SHANS[2:0] bits. Only set the SHMD bit while the ADCSR.ADST bit is 0.

When the sampling function is enabled, the sample-and-hold circuit operates sampling while the ADC12 is not operating, and it operates holding while the ADC12 is operating.

Note: The ADCSR.ADST bit must become 1 after a time of 400 ns or more elapses after the SHMD bit is set to 1 (when the permissible signal source impedance is 1 kΩ). The sampling period of the sample-and-hold circuit must be 400 ns or more (when the permissible signal source impedance is 1 kΩ).

45.2.22 ADDISCR : A/D Disconnection Detection Control Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x07A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PCHG	ADNDIS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	ADNDIS[3:0]	Disconnection Detection Assist Setting 0x0: The disconnection detection assist function is disabled 0x1: Setting prohibited Others: The number of states for the discharge or precharge period.	R/W
4	PCHG	Precharge/discharge select 0: Discharge 1: Precharge	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The ADDISCR register selects either precharge or discharge, and the period of precharge or discharge for the A/D disconnection detection assist function. Only set the ADDISCR register when the ADCSR.ADST bit is 0. When the temperature sensor output or internal reference voltage is converted, the A/D converter executes discharge automatically.

Disable the disconnection detection assist function if any of the following functions are used:

- The temperature sensor
- The internal reference voltage
- A/D self-diagnosis
- VBATT 1/3 voltage monitor

ADNDIS[3:0] bits (Disconnection Detection Assist Setting)

The ADNDIS[3:0] bits specify the period of precharge or discharge. When ADNDIS[3:0] = 0000b, the disconnection detection assist function is disabled. Setting the ADNDIS[3:0] bits to 0001b is prohibited. Except when ADNDIS[3:0] = 0000b or 0001b, the specified value indicates the number of states for the period of precharge or discharge. When the ADNDIS[3:0] bits are set to any values other than 0000b or 0001b, the disconnection detection assistance function is enabled.The ADNDIS[4:0] should be set while the ADCSR.ADST bit is 0. When the ADNDIS[3:0] bits are set to any values other than 0000b, and the disconnection detection assistance is enabled, the disconnection detection assistance for the channel-dedicated sample-and-hold circuit used for analog inputs are also enabled. When the temperature sensor output or internal reference voltage is converted or self-diagnosis is used, the disconnection detection assistance cannot be used. In that case, the ADNDIS[3:0] bits should be set to 0000b.

PCHG bit (Precharge/discharge select)

The PCHG bit selects either precharge or discharge.

45.2.23 ADGSPCR : A/D Group Scan Priority Control Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x080

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GBRP	LGRRS	—	—	—	—	—	—	—	—	—	—	—	—	GBRSCN	PGS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PGS	Group Priority Operation Setting*1 0: Operate without group priority control. 1: Operate with group priority control.	R/W
1	GBRSCN	Lower-Priority Group Restart Setting (enabled only when PGS = 1 and reserved when PGS = 0.) 0: Disable rescanning of the group that was stopped in group priority operation 1: Enable rescanning of the group that was stopped in group priority operation.	R/W
13:2	—	These bits are read as 0. The write value should be 0.	R/W
14	LGRRS	Restart Channel Select Enabled only when PGS = 1 and GBRSCN = 1. 0: Start rescanning from the first channel for scanning 1: Start rescanning from the channel for which A/D conversion is not completed.	R/W
15	GBRP	Single Scan Continuous Start*2 (enabled only when PGS = 1 and reserved when PGS = 0.) 0: Single scan is not continuously activated. 1: Single scan for the group with the lower-priority is continuously activated.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode) before setting PGS to 1. Operation is not guaranteed if these bits are set to any other value.
Note 2. When the GBRP bit is set to 1, single scan is performed continuously for the group with the lower-priority regardless of the setting in the GBRSCN bit.

PGS bit (Group Priority Operation Setting)

The PGS bit controls group priority operation in group scan mode. Set the PGS bit to 1 to enable group priority operation.

The ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode) before setting the PGS bit to 1. Operation is not guaranteed if the bits are set to any other value.

When the PGS bit is set to 0, a clear operation must be performed by software as described in section 45.6.3. Constraints on Stopping A/D Conversion. When the PGS bit is set to 1, use the settings described in section 45.3.4.3. Group Priority Operation.

GBRSCN bit (Lower-Priority Group Restart Setting)

The GBRSCN bit controls the restarting of scan operation in group priority operation.

When the GBRSCN bit is set to 1, if the scan operation of a lower-priority group is stopped by a trigger input of a priority group, the lower-priority group scanning is restarted on completion of the priority group scanning. If a trigger of a lower-priority group is input during scanning of the priority group, the lower-priority group scanning is started on completion of the priority group scanning.

When the GBRSCN bit is set to 0, triggers input during scanning are ignored. Set the GBRSCN bit while the ADCSR.ADST bit is 0.

LGRRS bit (Restart Channel Select)

This bit sets the channel from which rescanning is to be started in group priority operation. The setting of the LGRRS bit is valid when the PGS and GBRSCN bits are 1.

If the LGRRS bit is 0, scanning of a lower-priority group that was stopped in group priority operation is restarted from the first channel after scanning of the priority group completes.

If the LGRRS bit is 1, scanning of a lower-priority group that was stopped in group priority operation is restarted (upon completion of scanning of the priority group) from the channel for which A/D conversion is not complete. If A/D conversion of the addition setting channel was not completed the specified number of times when scanning stopped, A/D conversion of the addition setting channel is performed again the specified number of times when scanning restarts.

Set the LGRRS bit while the ADCSR.ADST bit is 0.

GBRP bit (Single Scan Continuous Start)

The GBRP bit is set when a single scan operation is to be performed continuously on the group with the lower-priority.

Setting the GBRP bit to 1 starts a single scan of the group with the lower-priority. On completion of the scan, another single scan of the group with the lower-priority is started automatically. If scanning has been stopped during group priority operation, single scan of the group with the lower-priority is automatically restarted on completion of the A/D conversion of the priority group.

Before setting the GBRP bit to 1, disable input of a trigger for the lower-priority group. If the GBRP bit is set to 1, rescanning is performed only on the group with the lower-priority even if the GBRSCN bit is set to 0.

45.2.24 ADCMPCR : A/D Compare Function Control Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x090

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPAIE	WCMPE	CMPBIE	—	CMPAE	—	CMPBE	—	—	—	—	—	—	—	CMPAB[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CMPAB[1:0]	Window A/B Composite Conditions Setting These bits are valid when both window A and window B are enabled (CMPAE = 1 and CMPBE = 1). 0 0: Output ADC12i_WCMPM (i = 0, 1) when window A OR window B comparison conditions are met. Otherwise, output ADC12i_WCMPUM (i = 0, 1). 0 1: Output ADC12i_WCMPM (i = 0, 1) when window A EXOR window B comparison conditions are met. Otherwise, output ADC12i_WCMPUM (i = 0, 1). 1 0: Output ADC12i_WCMPM (i = 0, 1) when window A AND window B comparison conditions are met. Otherwise, output ADC12i_WCMPUM (i = 0, 1). 1 1: Setting prohibited.	R/W
8:2	—	These bits are read as 0. The write value should be 0.	R/W
9	CMPBE	Compare Window B Operation Enable 0: Disable compare window B operation. Disable ADC12i_WCMPM (i = 0, 1) and ADC12i_WCMPUM (i = 0, 1) outputs. 1: Enable compare window B operation.	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	CMPAE	Compare Window A Operation Enable 0: Disable compare window A operation. Disable ADC12i_WCMPM (i = 0, 1) and ADC12i_WCMPUM (i = 0, 1) outputs. 1: Enable compare window A operation.	R/W
12	—	This bit is read as 0. The write value should be 0.	R/W
13	CMPBIE	Compare B Interrupt Enable 0: Disable ADC12i_CMPBI (i = 0, 1) interrupt when comparison conditions (window B) are met. 1: Enable ADC12i_CMPBI (i = 0, 1) interrupt when comparison conditions (window B) are met.	R/W

Bit	Symbol	Function	R/W
14	WCMPE	Window Function Setting 0: Disable window function Window A and window B operate as a comparator to compare the single value on the lower side with the A/D conversion result. 1: Enable window function Window A and window B operate as a comparator to compare the two values on the upper and lower sides with the A/D conversion result.	R/W
15	CMPAIE	Compare A Interrupt Enable 0: Disable ADC12i_CMPAI (i = 0, 1) interrupt when comparison conditions (window A) are met. 1: Enable ADC12i_CMPAI (i = 0, 1) interrupt when comparison conditions (window A) are met.	R/W

Note: S-TYPE-3, P-TYPE-3

CMPAB[1:0] bits (Window A/B Composite Conditions Setting)

The CMPAB[1:0] bits are valid when both window A and window B are enabled (CMPAE = 1 and CMPBE = 1) in single scan mode. These bits specify the compare function match/mismatch event output conditions and monitoring conditions of ADWINMON.MONCOMB. Only set the CMPAB[1:0] bits while the ADCSR.ADST bit is 0.

CMPBE bit (Compare Window B Operation Enable)

The CMPBE bit enables or disables the compare window B operation. Set the CMPBE bit while the ADCSR.ADST bit is 0.

Set this bit to 0 before setting the following registers:

- A/D Channel Select Registers A0, A1, B0, B1 (ADANSA0, ADANSA1, ADANSB0, ADANSB1)
- OCSB, TSSB, OCSA, or TSSA bits in the A/D Conversion Extended Input Control Register (ADEXICR)
- CMPCHB[5:0] bits in the Window B Channel Select Register (ADCMPBNSR)

CMPAE bit (Compare Window A Operation Enable)

The CMPAE bit enables or disables the compare window A operation. Set the CMPAE bit while the ADCSR.ADST bit is 0.

Set this bit to 0 before setting the following registers:

- A/D Channel Select Registers A0, A1, B0, B1 (ADANSA0, ADANSA1, ADANSB0, ADANSB1)
- OCSB, TSSB, OCSA, or TSSA bits in the A/D Conversion Extended Input Control Register (ADEXICR)
- Window A Channel Select Registers 0 and 1 (ADCMPANSR0 and ADCMPANSR1)
- Window A Extended Input Select Register (ADCMPANSER)

CMPBIE bit (Compare B Interrupt Enable)

The CMPBIE bit enables or disables the ADC12i_CMPBI (i = 0, 1) interrupt output when the comparison conditions (window B) are met.

WCMPE bit (Window Function Setting)

The WCMPE bit enables or disables the window function. Set the WCMPE bit while the ADCSR.ADST bit is 0.

CMPAIE bit (Compare A Interrupt Enable)

The CMPAIE bit enables or disables the ADC12i_CMPAI (i = 0, 1) interrupt output when the comparison conditions (window A) are met.

45.2.25 ADCMPANSR0 : A/D Compare Function Window A Channel Select Register 0

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x094

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPCHA15	CMPCHA14	CMPCHA13	CMPCHA12	CMPCHA11	CMPCHA10	CMPCHA09	CMPCHA08	CMPCHA07	CMPCHA06	CMPCHA05	CMPCHA04	CMPCHA03	CMPCHA02	CMPCHA01	CMPCHA00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPCHA15 to CMPCHA00*1	Compare Window A Channel Select Bit 15 (CMPCHA15) is associated with ANm15 and bit 0 (CMPCHA00) is associated with ANm00. 0: Disable compare function for associated input channel 1: Enable compare function for associated input channel	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.
Note 1. 00 to 08 (unit 0), 00 to 06 (unit 1) is available.

CMPCHAn bits (Compare Window A Channel Select)

The compare function is enabled by writing 1 to the CMPCHAn bits with the same number as the A/D conversion channel selected in the ADANSA0.ANSAn bits and the ADANSB0.ANSBn bits.

Set the CMPCHAn bits while the ADCSR.ADST bit is 0.

45.2.26 ADCMPANSR1 : A/D Compare Function Window A Channel Select Register 1

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x096

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPCHA31	CMPCHA30	CMPCHA29	CMPCHA28	CMPCHA27	CMPCHA26	CMPCHA25	CMPCHA24	CMPCHA23	CMPCHA22	CMPCHA21	CMPCHA20	CMPCHA19	CMPCHA18	CMPCHA17	CMPCHA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPCHA31 to CMPCHA16*1	Compare Window A Channel Select Bit 15 (CMPCHA31) is associated with ANm31 and bit 0 (CMPCHA16) is associated with ANm16. 0: Disable compare function for associated input channel 1: Enable compare function for associated input channel	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.
Note 1. 16 to 19 (unit 0), 16 to 22 (unit 1) is available.

CMPCHAn bits (Compare Window A Channel Select)

The compare function is enabled by writing 1 to the CMPCHAn bits with the same number as the A/D conversion channel selected in the ADANSA1.ANSA bits and the ADANSB1.ANSB bits.

Set the CMPCHAn bits while the ADCSR.ADST bit is 0.

45.2.27 ADCMPANSER : A/D Compare Function Window A Extended Input Select Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x092

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CMPOCA	CMPTSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMPTSA	Temperature Sensor Output Compare Select 0: Exclude the temperature sensor output from the compare Window A target range. 1: Include the temperature sensor output in the compare Window A target range.	R/W
1	CMPOCA	Internal Reference Voltage Compare Select 0: Exclude the internal reference voltage from the compare Window A target range. 1: Include the internal reference voltage in the compare Window A target range.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CMPTSA bit (Temperature Sensor Output Compare Select)

The compare Window A function is enabled by setting the CMPTSA bit to 1 while the ADEXICR.TSSA bit or the ADEXICR.TSSB bit is 1. Set the CMPTSA bit while the ADCSR.ADST bit is 0.

CMPOCA bit (Internal Reference Voltage Compare Select)

The compare window A function is enabled by setting the CMPOCA bit to 1 when the ADEXICR.OCSA and ADEXICR.OCSB bit is 1. Set the CMPOCA bit when the ADCSR.ADST bit is 0.

45.2.28 ADCMPLR0 : A/D Compare Function Window A Comparison Condition Setting Register 0

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x098

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPLCHA15	CMPLCHA14	CMPLCHA13	CMPLCHA12	CMPLCHA11	CMPLCHA10	CMPLCHA09	CMPLCHA08	CMPLCHA07	CMPLCHA06	CMPLCHA05	CMPLCHA04	CMPLCHA03	CMPLCHA02	CMPLCHA01	CMPLCHA00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPLCHA15 to CMPLCHA00*1	Compare Window A Comparison Condition Select These bits set comparison conditions for channels to which Window A comparison conditions are applied. Bit 15 (CMPLCHA15) is associated with ANm15 and bit 0 (CMPLCHA00) is associated with ANm00. Comparison conditions are shown in Figure 45.4. 0: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 value > A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): A/D-converted value < ADCMPDR0 value, or ADCMPDR1 value < A/D-converted value 1: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 value < A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): ADCMPDR0 value < A/D-converted value < ADCMPDR1 value	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.
Note 1. 00 to 08 (unit 0), 00 to 06 (unit 1) is available.

CMPLCHAn bits (Compare Window A Comparison Condition Select)

The CMPLCHAn bits specify the comparison conditions for channels to which Window A comparison conditions are applied. These bits can be set for each analog input to be compared. When the comparison result of each analog input meets the set condition, the ADCMPSR0.CMPSTCHAn flag sets to 1 and a compare interrupt (ADC12i_CMPAI (i = 0, 1)) is generated.

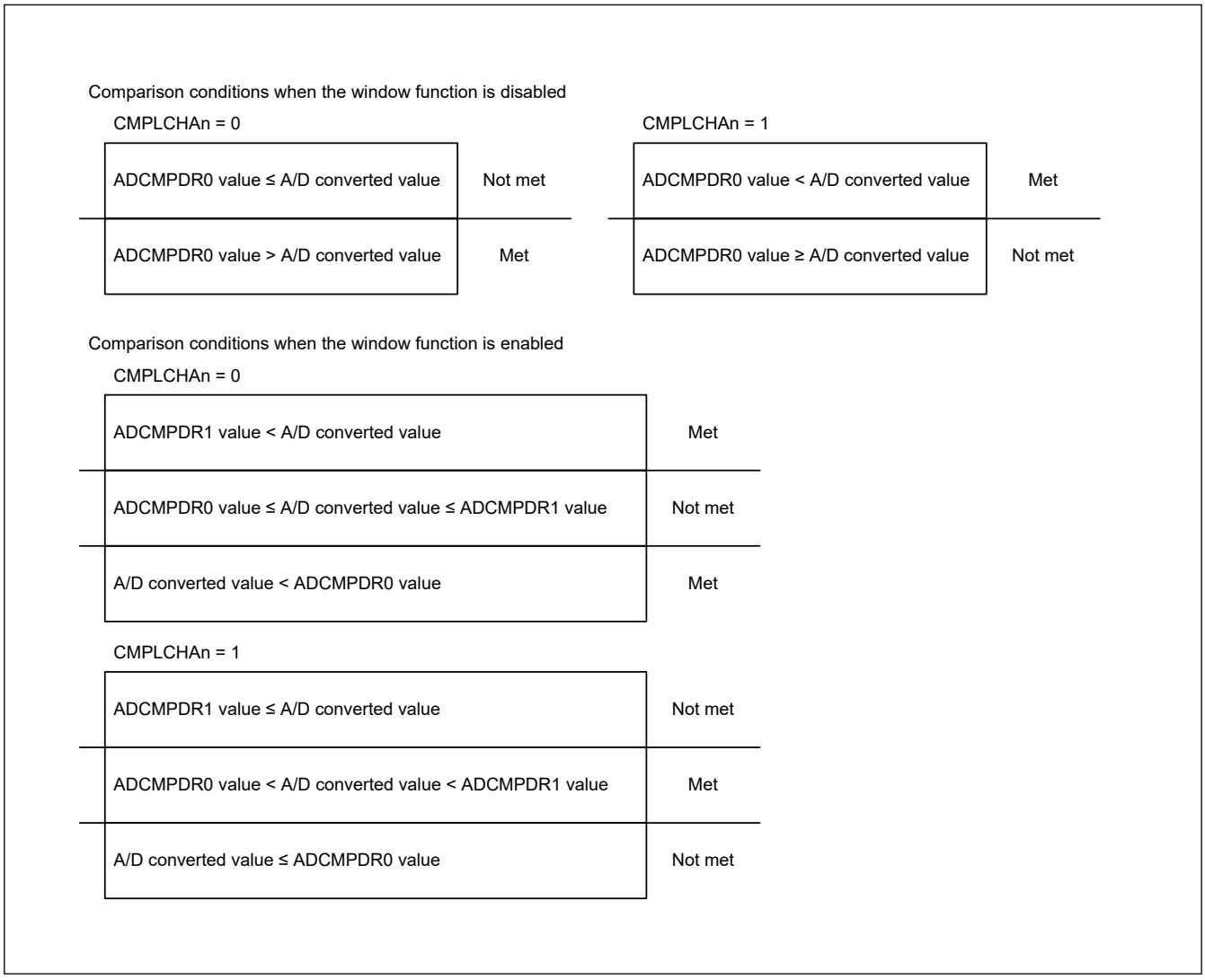


Figure 45.4 Explanation of comparison conditions for compare function Window A

45.2.29 ADCMPLR1 : A/D Compare Function Window A Comparison Condition Setting Register 1

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x09A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPLCHA31	CMPLCHA30	CMPLCHA29	CMPLCHA28	CMPLCHA27	CMPLCHA26	CMPLCHA25	CMPLCHA24	CMPLCHA23	CMPLCHA22	CMPLCHA21	CMPLCHA20	CMPLCHA19	CMPLCHA18	CMPLCHA17	CMPLCHA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPLCHA31 to CMPLCHA16*1	Compare Window A Comparison Condition Select These bits set comparison conditions for channels to which Window A comparison conditions are applied. Bit 15 (CMPLCHA31) is associated with ANm31 and bit 0 (CMPLCHA16) is associated with ANm16. Comparison conditions are shown in Figure 45.4. 0: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 value > A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): A/D-converted value < ADCMPDR0 value, or ADCMPDR1 value < A/D-converted value 1: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 value < A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): ADCMPDR0 value < A/D-converted value < ADCMPDR1 value	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.
Note 1. 16 to 19 (unit 0), 16 to 22 (unit 1) is available.

CMPLCHAn bits (Compare Window A Comparison Condition Select)

The CMPLCHAn bits specify the comparison conditions for analog channels to which window A comparison conditions are applied. These bits can be set for each analog input to be compared. When the comparison result of each analog input meets the set condition, the ADCMPSR1.CMPSTCHAn bit is set to 1 and a compare interrupt (ADC12i_CMPAI (i = 0, 1)) is generated.

45.2.30 ADCMPLER : A/D Compare Function Window A Extended Input Comparison Condition Setting Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x093

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CMPLOCA	CMPLTSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMPLTSA	Compare Window A Temperature Sensor Output Comparison Condition Select Comparison conditions are shown in Figure 45.4. 0: When window function is disabled (ADCMPCR.WCMPE = 0) : ADCMPDR0 value > A/D-converted valueCompare Window A Temperature Sensor Output Comparison Condition Select When window function is enabled (ADCMPCR.WCMPE = 1) : Compare Window A Temperature Sensor Output Comparison ConditionA/D-converted value < ADCMPDR0 value, or A/D-converted value > ADCMPDR1 value 1: When window function is disabled (ADCMPCR.WCMPE = 0) : ADCMPDR0 value < A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1) : ADCMPDR0 value < A/D-converted value < ADCMPDR1 value	R/W
1	CMPLOCA	Compare Window A Internal Reference Voltage Comparison Condition Select Comparison conditions are shown in Figure 45.4. 0: When window function is disabled (ADCMPCR.WCMPE = 0) : ADCMPDR0 value > A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): A/D-converted value < ADCMPDR0 value, or A/D-converted value > ADCMPDR1 value 1: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 value < A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): ADCMPDR0 value < A/D-converted value < ADCMPDR1 value	R/W

Bit	Symbol	Function	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CMPLTSA bit (Compare Window A Temperature Sensor Output Comparison Condition Select)

The CMPLTSA bit specifies comparison conditions when the temperature sensor output is the target for the Window A comparison condition. When the temperature sensor output comparison result meets the set condition, the ADCMPSER.CMPSTTSA flag sets to 1 and a compare interrupt (ADC12i_CMPAI (i = 0, 1)) is generated.

CMPLOCA bit (Compare Window A Internal Reference Voltage Comparison Condition Select)

The CMPLOCA bit specifies comparison conditions when the internal reference voltage is the target for the Window A comparison condition. When the internal reference voltage comparison result meets the set condition, the ADCMPSER.CMPSTOCA flag sets to 1 and a compare interrupt (ADC12i_CMPAI (i = 0, 1)) is generated.

45.2.31 ADCMPDRn : A/D Compare Function Window A Lower-Side/Upper-Side Level Setting Register (n = 0, 1)

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x09C + (0x2 × n)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: S-TYPE-3, P-TYPE-3

The ADCMPDRy (y = 0, 1) register specifies the reference data when the compare window A function is used. ADCMPDR0 sets the lower reference for window A, and ADCMPDR1 sets the upper reference for window A.

ADCMPDRy are read/write registers.

ADCMPDRy are writable even during A/D conversion. The reference data can be dynamically changed by rewriting register values during A/D conversion*1.

Set these registers so that the upper reference is not less than the lower reference (ADCMPDR1 ≥ ADCMPDR0). ADCMPDR1 are not used when the window function is disabled.

Note 1. The lower and the upper references are changed when each register is written. For example, when the upper reference value is changed and the lower reference value is being changed, the MCU compares the upper reference (after rewrite), and the lower reference (before rewrite) with the A/D conversion result. See Figure 45.5. If the comparison during the rewriting of these two references is erroneous, then rewrite these reference values when both ADCSR.ADST and the target Compare Window Operation Enable bit (ADCMPCR.CMPAE or ADCMPCR.CMPBE) are 0.

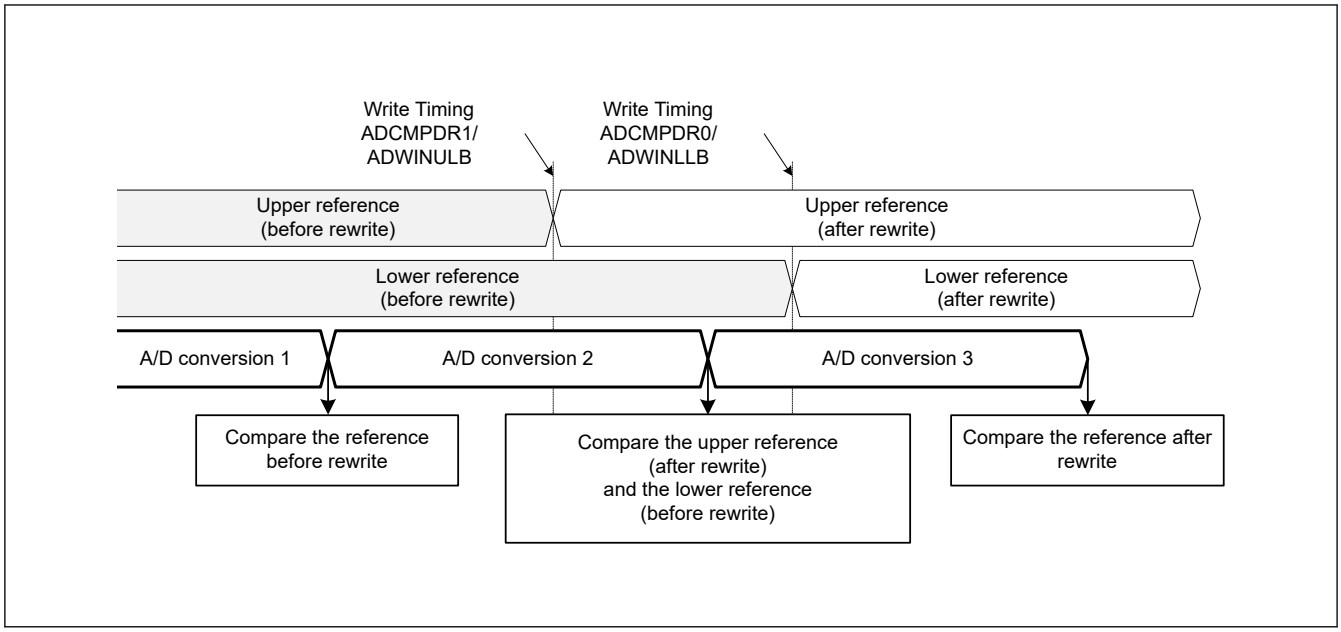


Figure 45.5 Comparison between upper and lower references before and after a rewrite

The ADCMPDRy registers use different formats depending on the following conditions:

- The value of A/D Data Register Format Select bit (flush-right or flush-left)
- The value of the A/D Conversion Accuracy Select bit (12-bit, 10-bit, 8-bit)
- The value of A/D-Converted Value Addition/Average Channel Select bits (A/D-converted value addition mode selected or not selected).

The data formats for each condition are shown as follows:

1. When A/D-converted value addition mode is not selected
   - Flush-right data with 12-bit accuracy — Lower 12 bits ([11:0]) are valid
   - Flush-right data with 10-bit accuracy — Lower 10 bits ([9:0]) are valid
   - Flush-right data with 8-bit accuracy — Lower 8 bits ([7:0]) are valid
   - Flush-left data with 12-bit accuracy — Upper 12 bits ([15:4]) are valid
   - Flush-left data with 10-bit accuracy — Upper 10 bits ([15:6]) are valid
   - Flush-left data with 8-bit accuracy — Upper 8 bits ([15:8]) are valid
2. When A/D-converted value addition mode is selected
   - Flush-right data with 12-bit accuracy — Lower 14 bits ([13:0]) are valid
   - Flush-right data with 10-bit accuracy — Lower 12 bits ([11:0]) are valid
   - Flush-right data with 8-bit accuracy — Lower 10 bits ([9:0]) are valid
   - Flush-left data with 12-bit accuracy — Upper 14 bits ([15:2]) are valid
   - Flush-left data with 10-bit accuracy — Upper 12 bits ([15:4]) are valid
   - Flush-left data with 8-bit accuracy — Upper 10 bits ([15:6]) are valid

45.2.32 ADWINnLB : A/D Compare Function Window B Lower-Side/Upper-Side Level Setting Register (n = L, U)

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x0A8 (n = L)
0x0AA (n = U)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: S-TYPE-3, P-TYPE-3

The ADWINULB and ADWINLLB registers specify the reference data when the compare window B function is used. ADWINLLB sets the lower reference for window B, and ADWINULB sets the upper reference for window B.

ADWINnLB are read/write registers.

ADWINnLB are writable even during A/D conversion. The reference data can be dynamically changed by rewriting register values during A/D conversion*1.

Set these registers so that the upper reference is not less than the lower reference (ADWINULB ≥ ADWINLLB). ADWINULB are not used when the window function is disabled.

Note 1. The lower and the upper references are changed when each register is written. For example, when the upper reference value is changed and the lower reference value is being changed, the MCU compares the upper reference (after rewrite), and the lower reference (before rewrite) with the A/D conversion result. See Figure 45.6. If the comparison during the rewriting of these two references is erroneous, then rewrite these reference values when both ADCSR.ADST and the target Compare Window Operation Enable bit (ADCMPCR.CMPAE or ADCMPCR.CMPBE) are 0.

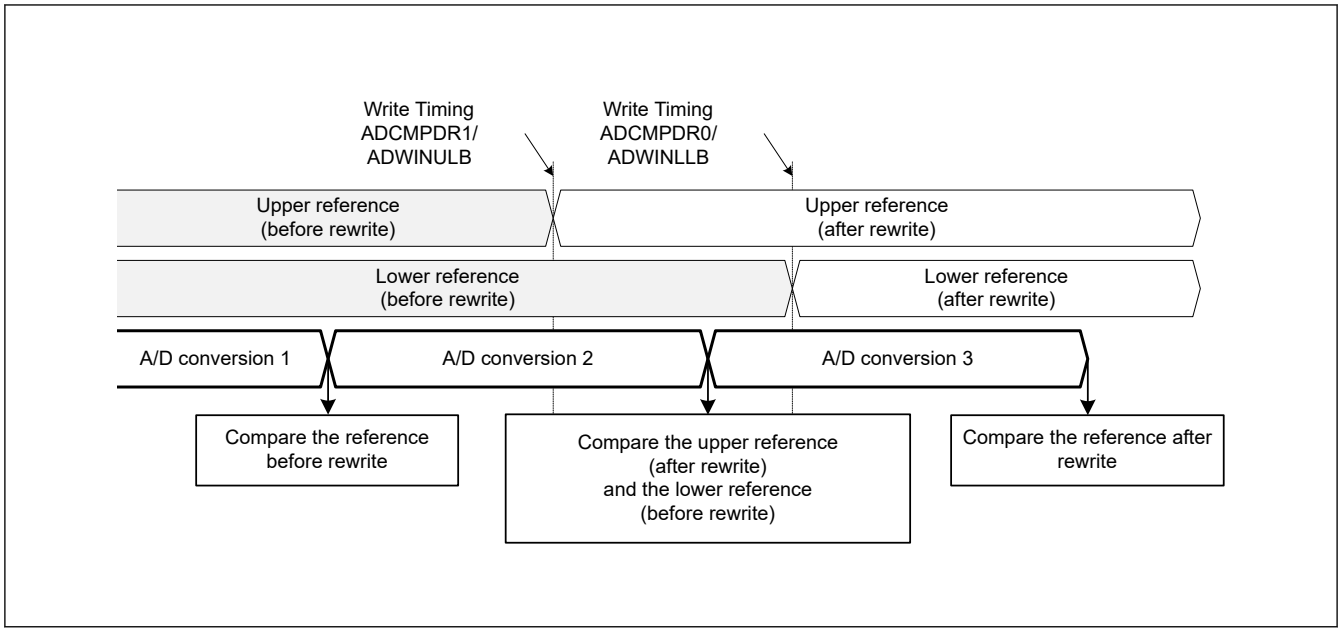


Figure 45.6 Comparison between upper and lower references before and after a rewrite

The ADWINnLB registers use different formats depending on the following conditions:

- The value of A/D Data Register Format Select bit (flush-right or flush-left)
- The value of the A/D Conversion Accuracy Select bit (12-bit, 10-bit, 8-bit)
- The value of A/D-Converted Value Addition/Average Channel Select bits (A/D-converted value addition mode selected or not selected).

The data formats for each condition are shown as follows:

1. When A/D-converted value addition mode is not selected
   - Flush-right data with 12-bit accuracy — Lower 12 bits ([11:0]) are valid
   - Flush-right data with 10-bit accuracy — Lower 10 bits ([9:0]) are valid
   - Flush-right data with 8-bit accuracy — Lower 8 bits ([7:0]) are valid
   - Flush-left data with 12-bit accuracy — Upper 12 bits ([15:4]) are valid
   - Flush-left data with 10-bit accuracy — Upper 10 bits ([15:6]) are valid
   - Flush-left data with 8-bit accuracy — Upper 8 bits ([15:8]) are valid
2. When A/D-converted value addition mode is selected
   - Flush-right data with 12-bit accuracy — Lower 14 bits ([13:0]) are valid
   - Flush-right data with 10-bit accuracy — Lower 12 bits ([11:0]) are valid
   - Flush-right data with 8-bit accuracy — Lower 10 bits ([9:0]) are valid
   - Flush-left data with 12-bit accuracy — Upper 14 bits ([15:2]) are valid
   - Flush-left data with 10-bit accuracy — Upper 12 bits ([15:4]) are valid
   - Flush-left data with 8-bit accuracy — Upper 10 bits ([15:6]) are valid

45.2.33 ADCMPSR0 : A/D Compare Function Window A Channel Status Register 0

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x0A0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPSTCHA15	CMPSTCHA14	CMPSTCHA13	CMPSTCHA12	CMPSTCHA11	CMPSTCHA10	CMPSTCHA09	CMPSTCHA08	CMPSTCHA07	CMPSTCHA06	CMPSTCHA05	CMPSTCHA04	CMPSTCHA03	CMPSTCHA02	CMPSTCHA01	CMPSTCHA00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPSTCHA15 to CMPSTCHA00*1	Compare Window A Flag When Window A operation is enabled (ADCMPCR.CMPAE = 1b), these bits indicate the comparison result of channels to which Window A comparison conditions are applied. Bit 15 (CMPSTCHA15) is associated with ANm15 and bit 0 (CMPSTCHA00) is associated with ANm00. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.
Note 1. 00 to 08 (unit 0), 00 to 06 (unit 1) is available.

CMPSTCHAn flags (Compare Window A Flag)

The CMPSTCHAn flags indicate the comparison results for channels to which Window A comparison conditions are applied. When a comparison condition set in ADCMPLR0.CMPLCHA is met at the end of A/D conversion, the associated CMPSTCHAn flag sets to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt request (ADC12i_CMPAI (i = 0, 1)) is generated when this flag sets to 1.

Writing 1 to the CMPSTCHAn flags is invalid.

[Setting condition]

- The condition set in ADCMPLR0.CMPLCHA is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

45.2.34 ADCMPSR1 : A/D Compare Function Window A Channel Status Register1

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x0A2

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPSTCHA31	CMPSTCHA30	CMPSTCHA29	CMPSTCHA28	CMPSTCHA27	CMPSTCHA26	CMPSTCHA25	CMPSTCHA24	CMPSTCHA23	CMPSTCHA22	CMPSTCHA21	CMPSTCHA20	CMPSTCHA19	CMPSTCHA18	CMPSTCHA17	CMPSTCHA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPSTCHA31 to CMPSTCHA16*1	Compare Window A Flag When Window A operation is enabled (ADCMPCR.CMPAE = 1), these bits indicate the comparison result of channels to which Window A comparison conditions are applied. Bit 15 (CMPSTCHA31) is associated with ANm31 and bit 0 (CMPSTCHA16) is associated with ANm16. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.
Note 1. 16 to 19 (unit 0), 16 to 22 (unit 1) is available.

CMPSTCHAn flags (Compare Window A Flag)

The CMPSTCHAn flags indicate the comparison results for channels to which Window A comparison conditions are applied. When the comparison condition set in ADCMPLR1.CMPLCHA is met at the end of A/D conversion, the associated CMPSTCHAn flag sets to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt request (ADC12i_CMPAI (i = 0, 1)) is generated when this flag sets to 1.

Writing 1 to the CMPSTCHAn flags is invalid.

[Setting condition]

- The condition set in ADCMPLR1.CMPLCHA is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

45.2.35 ADCMPSER : A/D Compare Function Window A Extended Input Channel Status Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x0A4

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CMPSTOCA	CMPSTTSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMPSTTSA	Compare Window A Temperature Sensor Output Compare Flag When Window A operation is enabled (ADCMPCR.CMPAE = 1), this bit indicates the temperature sensor output comparison result. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W

Bit	Symbol	Function	R/W
1	CMPSTOCA	Compare Window A Internal Reference Voltage Compare Flag When Window A operation is enabled (ADCMPCR.CMPAE = 1), this bit indicates the internal reference voltage comparison result. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The ADCMPSER register stores compare results of compare function window A.

CMPSTTSA flag (Compare Window A Temperature Sensor Output Compare Flag)

The CMPSTTSA flag indicates the temperature sensor output comparison result. When the comparison condition set in ADCMPLER.CMPLTSA is met at the end of A/D conversion, this flag sets to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt request (ADC12i_CMPAI (i = 0, 1)) is generated when this flag sets to 1.

Writing 1 to the CMPSTTSA flag is invalid.

[Setting condition]

- The condition set in ADCMPLER.CMPLTSA is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

CMPSTOCA flag (Compare Window A Internal Reference Voltage Compare Flag)

The CMPSTOCA flag indicates the internal reference voltage comparison result. When the comparison condition set in ADCMPLER.CMPLOCA is met at the end of A/D conversion, this flag sets to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt request (ADC12i_CMPAI (i = 0, 1)) is generated when this flag sets to 1.

Writing 1 to the CMPSTOCA flag is invalid.

[Setting condition]

- The condition set in ADCMPLER.CMPLOCA is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

45.2.36 ADCMPBNSR : A/D Compare Function Window B Channel Select Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x0A6

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CMPLB	—	CMPCHB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>5:0</td><td>CMPCHB[5:0]</td><td>Compare Window B Channel Select
These bits select channels to be compared with the compare Window B conditions.
<table>
<tr><th>CMPCHB[5:0]</th><th>Unit 0</th><th>Unit 1</th></tr>
<tr><td>0x00</td><td>AN000</td><td>AN100</td></tr>
<tr><td>0x01</td><td>AN001</td><td>AN101</td></tr>
<tr><td>0x02</td><td>AN002</td><td>AN102</td></tr>
<tr><td>0x03</td><td>AN003 / VBATT 1/3 voltage monitor output</td><td>AN103 / VBATT 1/3 voltage monitor output</td></tr>
<tr><td>0x04</td><td>AN004</td><td>AN104</td></tr>
<tr><td>0x05</td><td>AN005</td><td>AN105</td></tr>
<tr><td>0x06</td><td>AN006</td><td>AN106</td></tr>
<tr><td>0x07</td><td>AN007</td><td>Not selected</td></tr>
<tr><td>0x08</td><td>AN008</td><td>Not selected</td></tr>
<tr><td>0x09</td><td>Not selected</td><td>Not selected</td></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td></tr>
<tr><td>0x0F</td><td>Not selected</td><td>Not selected</td></tr>
<tr><td>0x10</td><td>AN016</td><td>AN116</td></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td></tr>
<tr><td>0x13</td><td>AN019</td><td>AN119</td></tr>
<tr><td>0x14</td><td>Not selected</td><td>AN120</td></tr>
<tr><td>0x15</td><td>Not selected</td><td>AN121</td></tr>
<tr><td>0x16</td><td>Not selected</td><td>AN122</td></tr>
<tr><td>0x20</td><td colspan="2">Temperature sensor</td></tr>
<tr><td>0x21</td><td colspan="2">Internal reference voltage</td></tr>
<tr><td>0x3F</td><td colspan="2">Not selected</td></tr>
<tr><td>Others</td><td colspan="2">Setting prohibited</td></tr>
</table></td><td>R/W</td></tr>
<tr><td>6</td><td>—</td><td>This bit is read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>7</td><td>CMPLB</td><td>Compare Window B Comparison Condition Setting
This bit sets comparison conditions for channels for Window B. The comparison conditions are shown in Figure 45.7.
0: When window function is disabled (ADCMPCR.WCMPE = 0):
ADWINLLB value > A/D-converted value
When window function is enabled (ADCMPCR.WCMPE = 1):
A/D-converted value < ADWINLLB value, or ADWINULB value < A/D-converted value
1: When window function is disabled (ADCMPCR.WCMPE = 0):
ADWINLLB value < A/D-converted value
When window function is enabled (ADCMPCR.WCMPE = 1):
ADWINLLB value < A/D-converted value < ADWINULB value</td><td>R/W</td></tr>
</table>

Note: S-TYPE-3, P-TYPE-3

CMPCHB[5:0] bits (Compare Window B Channel Select)

The CMPCHB[5:0] bits specify the channels to be compared with the compare Window B conditions from AN000 to AN002, AN004 to AN008, AN016 to AN019, AN100 to AN102, AN104 to AN106, AN116 to AN122, the temperature sensor, the internal reference voltage and the VBATT 1/3 voltage monitor output. The compare Window B function is enabled by specifying the hexadecimal number of the A/D conversion channel selected in the ADANSA0, ADANSA1, ADANSB0, ADANSB1 registers.

Set the CMPCHB[5:0] bits while the ADCSR.ADST bit is 0.

CMPLB bit (Compare Window B Comparison Condition Setting)

The CMPLB bit specifies the comparison conditions for channels for Window B. When the comparison result of an analog input meets the set condition, the associated ADCMPBSR.CMPSTB flag sets to 1 and a compare interrupt request (ADC12i_CMPBI (i = 0, 1)) is generated.

Compare conditions when the window function is disabled

CMPLB = 0

ADWINLLB value ≤ A/D converted value	Not met
ADWINLLB value > A/D converted value	Met

CMPLB = 1

ADWINLLB value < A/D converted value	Met
ADWINLLB value ≥ A/D converted value	Not met

Compare conditions when the window function is enabled

CMPLB = 0

A/D converted value > ADWINULB value	Met
ADWINLLB value ≤ A/D converted value ≤ ADWINULB value	Not met
A/D converted value < ADWINLLB value	Met

CMPLB = 1

A/D converted value ≥ ADWINULB value	Not met
ADWINLLB value <A/D converted value < ADWINULB value	Met
A/D converted value ≤ ADWINLLB value	Not met

Figure 45.7 Explanation of compare conditions for compare function Window B

45.2.37 ADCMPBSR : A/D Compare Function Window B Status Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x0AC

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CMPSTB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMPSTB	Compare Window B Flag When Window B operation is enabled (ADCMPCR.CMPBE = 1), this bit indicates the comparison result of channels to which Window B comparison conditions are applied, temperature sensor output, internal reference voltage. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CMPSTB flag (Compare Window B Flag)

The CMPSTB flag indicates the comparison result of channels to which Window B comparison conditions are applied, the temperature sensor output, internal reference voltage. When the comparison condition set in ADCMPBNSR.CMPLB is met at the end of A/D conversion, this flag sets to 1. When the ADCMPCR.CMPBIE bit is 1, a compare interrupt request (ADC12i_CMPBI (i = 0, 1)) is generated when this flag sets to 1.

Writing 1 to the CMPSTB flag is invalid.

[Setting condition]

- The condition set in ADCMPBNSR.CMPLB is met when ADCMPCR.CMPBE = 1.

[Clearing condition]

- Writing 0 after reading 1.

45.2.38 ADWINMON : A/D Compare Function Window A/B Status Monitor Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x08C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	MONCMPB	MONCMPA	—	—	—	MONCOMB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MONCOMB	Combination Result Monitor This bit indicates the combination result. This bit is valid when both Window A and Window B operations are enabled. 0: Window A/B composite conditions are not met. 1: Window A/B composite conditions are met.	R
3:1	—	These bits are read as 0.	R
4	MONCMPA	Comparison Result Monitor A 0: Window A comparison conditions are not met. 1: Window A comparison conditions are met.	R
5	MONCMPB	Comparison Result Monitor B 0: Window B comparison conditions are not met. 1: Window B comparison conditions are met.	R
7:6	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

MONCOMB bit (Combination Result Monitor)

The read-only MONCOMB bit indicates the combined result of comparison condition results A and B based on the combination condition set in the ADCMPCR.CMPAB[1:0] bits.

[Setting condition]

- The combined result meets the combination condition set in the ADCMPCR.CMPAB[1:0] bits when ADCMPCR.CMPAE = 1 and ADCMPCR.CMPBE = 1.

[Clearing conditions]

- The combined result does not meet the combination condition set in the ADCMPCR.CMPAB[1:0] bits.
- ADCMPCR.CMPAE = 0 or ADCMPCR.CMPBE = 0.

MONCMPA bit (Comparison Result Monitor A)

The read-only MONCMPA bit is read as 1 when the A/D-converted value of the Window A target channel meets the condition set in ADCMPLR0/ADCMPLR1 and ADCMPLER. Otherwise, it is read as 0.

[Setting condition]

- The A/D-converted value meets the condition set in the ADCMPLR0/ADCMPLR1 and ADCMPLER registers when ADCMPCR.CMPAE = 1.

[Clearing conditions]

- The A/D-converted value does not meet the condition set in the ADCMPLR0/ADCMPLR1 and ADCMPLER registers when ADCMPCR.CMPAE = 1.
- ADCMPCR.CMPAE = 0 (automatically cleared when the ADCMPCR.CMPAE value changes from 1 to 0).

MONCMPB bit (Comparison Result Monitor B)

The read-only MONCMPB bit is read as 1 when the A/D-converted value of the Window B target channel meets the condition set in the ADCMPBNSR.CMPLB bit. Otherwise, it is read as 0.

[Setting condition]

- The A/D-converted value meets the condition set in ADCMPBNSR.CMPLB when ADCMPCR.CMPBE = 1.

[Clearing conditions]

- The A/D-converted value does not meet the condition set in ADCMPBNSR.CMPLB when ADCMPCR.CMPBE = 1.
- ADCMPCR.CMPBE = 0 (automatically cleared when the ADCMPCR.CMPBE value changes from 1 to 0).

45.2.39 ADBUFEN : A/D Data Buffer Enable Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x0D0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	BUFEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BUFEN	Data Buffer Enable 0: The data buffer is not used. 1: The data buffer is used.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The ADBUFEN register sets whether to enable the data buffer.

BUFEN bit (Data Buffer Enable)

This bit enables the use of the data buffer.

When BUFEN = 1b, A/D conversion result (addition result) other than self-diagnosis result is stored in ADBUFn.

Disable the data storage operation (BUFEN = 0b) before reading ADBUFPTR.

Do not use the data buffer for data duplexing, or group scan.

45.2.40 ADBUFPTR : A/D Data Buffer Pointer Register

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x0D2

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PTROVF	BUFPTR[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	BUFPTR[3:0]	Data Buffer Pointer These bits indicate the number of data buffer to which the next A/D converted data is transferred.	R/W
4	PTROVF	Pointer Overflow Flag 0: The data buffer pointer has not overflowed. 1: The data buffer pointer has overflowed.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

ADBUFPTR is a register that indicates the data buffer pointer and overflow status.

BUFPTR[3:0] bit (Data Buffer Pointer)

These bits indicate the number of data buffer to which the next A/D converted data is transferred.

When data has been transferred to data buffer 15, the pointer value becomes 0000b and the PTROVF bit is set to 1.

When the next data has been transferred, the data in data buffer 0 is overwritten.

Writing 0x00 to this register clears the value of these bits. Writing a value other than 0x00 is disabled.

PTROVF bit (Pointer Overflow Flag)

This bit indicates whether the data buffer pointer has overflowed. This bit is set to 1 when the pointer value becomes 0000b (overflow).

Writing 0x00 to this register clears this bit value. Writing a value other than 0x00 is disabled.

45.2.41 ADBUFn : A/D Data Buffer Registers n (n = 0 to 15)

Base address: ADC12m = 0x4033_2000 + 0x0200 × m (m = 0, 1)
ADC12m_NS = 0x5033_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x0B0 + 0x2 × n (n = 0 to 15)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADBUF[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ADBUF[15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See Table 45.24 and Table 45.25.	R

Note: S-TYPE-3, P-TYPE-3

ADBUFn registers are 16-bit read-only registers that sequentially store all A/D conversion results. The automatic clear function is not applied to these registers.

ADBUFn settings are the same as the A/D data register format settings.

The following conditions determine the formats for data in the ADBUFn registers:

- Setting of the Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 3, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

Table 45.24 shows the bit assignment for each accuracy.

Table 45.24 Bit assignment for each accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, This register indicatesThese registers indicate the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 12-bit, 10-bit, 8-bit accuracy, 1, 2, 3, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, this register indicatesthese registers indicate the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

Table 45.25 shows the bit assignment for each accuracy.

Table 45.25 Bit assignment for each accuracy when A/D-converted value addition mode is selected

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	These bits are read as 0.	Added Value 13 to 0: 14-bit sum of A/D conversion results														
Left-justified data with 12-bit accuracy	When 1, 2, 3, or 4 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 16 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

45.3 Operation

45.3.1 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.

Scan conversion is performed in any of the three operating modes:

- Single scan mode
- Continuous scan mode
- Group scan mode

In single scan mode, one or more specified channels are scanned once. In continuous scan mode, one or more specified channels are scanned repeatedly until software sets the ADCSR.ADST bit to 0. In group scan mode, the selected channels in group A, B are scanned once after scan starts in response to the respective synchronous trigger.

In single scan mode and continuous scan mode, A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n. In group scan mode, A/D conversion is performed for the ANn channels in group A selected in the ADANSA0 and ADANSA1 registers, and for the ANn channels in group B selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.

When self-diagnosis is selected, it is executed once at the beginning of each scan and one of the three reference voltages is converted.

The temperature sensor output and internal reference voltage can be selected at the same time as the analog input of the channels, and A/D conversion is performed on the analog input of channels, temperature sensor output, and internal reference voltage, in that order.

Double trigger mode can be used with single scan mode or group scan mode. With double trigger mode enabled (ADCSR.DBLE = 1), A/D conversion data of a channel selected in the ADCSR.DBLANS[4:0] bits is duplicated only if the conversion is started by the synchronous trigger (GPT, ELC) selected in the ADSTRGR.TRSA[5:0] bits. In group scan mode, only group A can use double trigger mode.

In the extended operation of double trigger mode, the A/D conversion operation is generated from the synchronous trigger combination selected in the ADSTRGR.TRSA[5:0] bits. In addition to normal double trigger mode operation, A/D conversion data with odd number trigger (ELC_AD00 (unit 0), ELC_AD10 (unit 1), GTCIADAn (Unit0) and GTCIADAm (Unit1) (n = 0 to 3, m = 4 to 7)) is stored in A/D Data Duplexing Register A (ADDBLDRA), and A/D conversion data with even number trigger (ELC_AD01 (unit 0), ELC_AD11 (unit1), GTCIADBn (Unit0) and GTCIADBm (Unit1) (n = 0 to 3, m = 4 to 7)) is stored in A/D Data Duplexing Register B (ADDBLDRB). In the extended operation of double trigger mode, when one of the trigger combinations occurs at the same time, the data duplexing register settings for the specified triggers do not work, and A/D conversion data is stored in A/D Data Duplexing Register B (ADDBLDRB).

The ADC12 ignores a synchronous trigger that occurs during the A/D conversion started by another synchronous trigger.

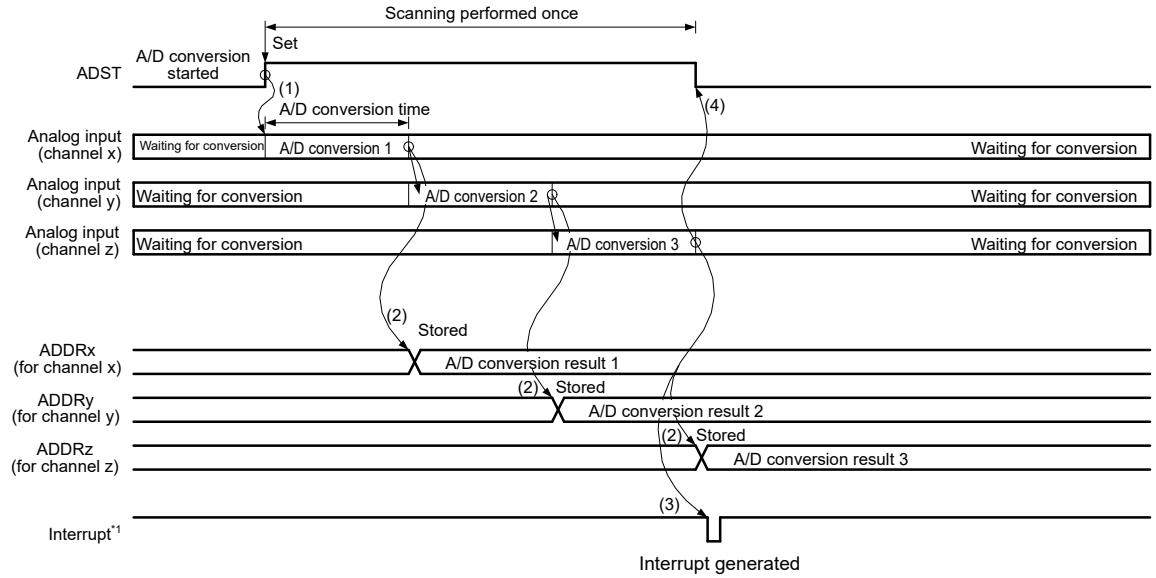
When any of the channels AN000 to AN002 (unit 0) are set as a channel-dedicated sample-and-hold circuit in the SHANS[2:0] bits in ADSHCR, the target analog input specified is sampled and held before the first A/D conversion of each scan.

45.3.2 Single Scan Mode

45.3.2.1 Basic Operation (Without Channel-Dedicated Sample-and-Hold Circuits)

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, a synchronous trigger input (GPT, ELC), or an asynchronous trigger input, A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
2. Each time A/D conversion of a single channel is completed, the A/D conversion result is stored in the associated A/D data register (ADDRy).
3. When A/D conversion of all the selected channels is completed, an ADC12i_ADI (i = 0, 1) interrupt request is generated.
4. The ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically set to 0 when A/D conversion of all the selected channels is completed. The ADC12 then enters a wait state.



Note: x, y, and z (x < y < z) are the selected channel numbers by ADANSA0/1.
Note 1. ADC12i_ADI (i = 0, 1)

Figure 45.8 Example basic operation in single scan mode when the analog inputs (channel x to z) are selected

45.3.2.2 Basic Operation (With Channel-Dedicated Sample-and-Hold Circuits and Continuous Sampling Disabled)

When the channel-dedicated sample-and-hold circuit is used, sample-and-hold operation is first performed, and then A/D conversion is performed on the analog input of all the specified channels once. The channels whose dedicated sample-and-hold circuit is to be used can be selected in the SHANS[2:0] bits in ADSHCR.

The operation is as follows:

1. Analog input sampling of all channels whose dedicated sample-and-hold circuit is to be used starts when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (GPT, ELC), or asynchronous trigger input.
2. After the sample-and-hold operation, A/D conversion is performed on the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
3. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D data register (ADDRy).
4. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0) interrupt request is generated (no register setting).
5. The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels completes. Then, the ADC12 enters a wait state.

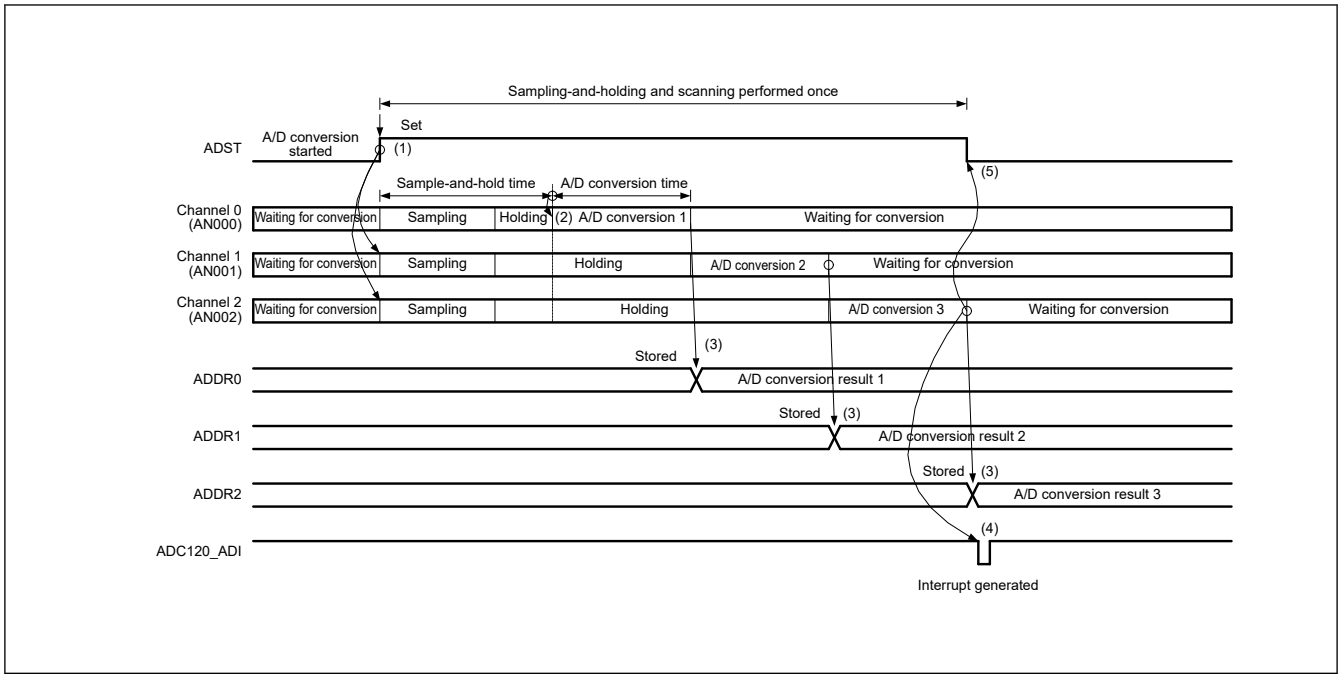


Figure 45.9 Example operation in single scan mode when the channel-dedicated sample-and-hold circuits are used and AN000 to AN002 are selected

45.3.2.3 Basic Operation (With Channel-Dedicated Sample-and-Hold Circuits and Continuous Sampling Enabled)

When a channel-dedicated sample-and-hold circuit is used while continuous sampling is enabled, sample-and-hold operations are performed first, followed by A/D conversion on the analog inputs on all selected channels, once. The ADSHCR.SHANS[2:0] bits specify the channels for which the channel-dedicated sample-and-hold circuits are to be used.

The operation is as follows:

1. When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuits selected in the ADSHCR.SHANS[2:0] bits start continuous sampling.
2. Analog input holding starts for all the channels for which the channel-dedicated sample-and-hold circuits are to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (GPT, ELC), or input of an asynchronous trigger.
3. After the stabilization time of the sample-and-hold circuits elapses, A/D conversion is performed on the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
4. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D data register (ADDRy), and the sample-and-hold circuit restarts continuous sampling.
5. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0) interrupt request is generated (no register setting).
6. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels completes. Then, the ADC12 enters a wait state. If this is followed by single scanning, set the continuous sampling time for the sample-and-hold circuits to at least 400 ns (when the permissible signal source impedance is 1 kΩ).
7. When the ADSHMSR.SHMD bit is set to 0, the sample-and-hold circuits stop.

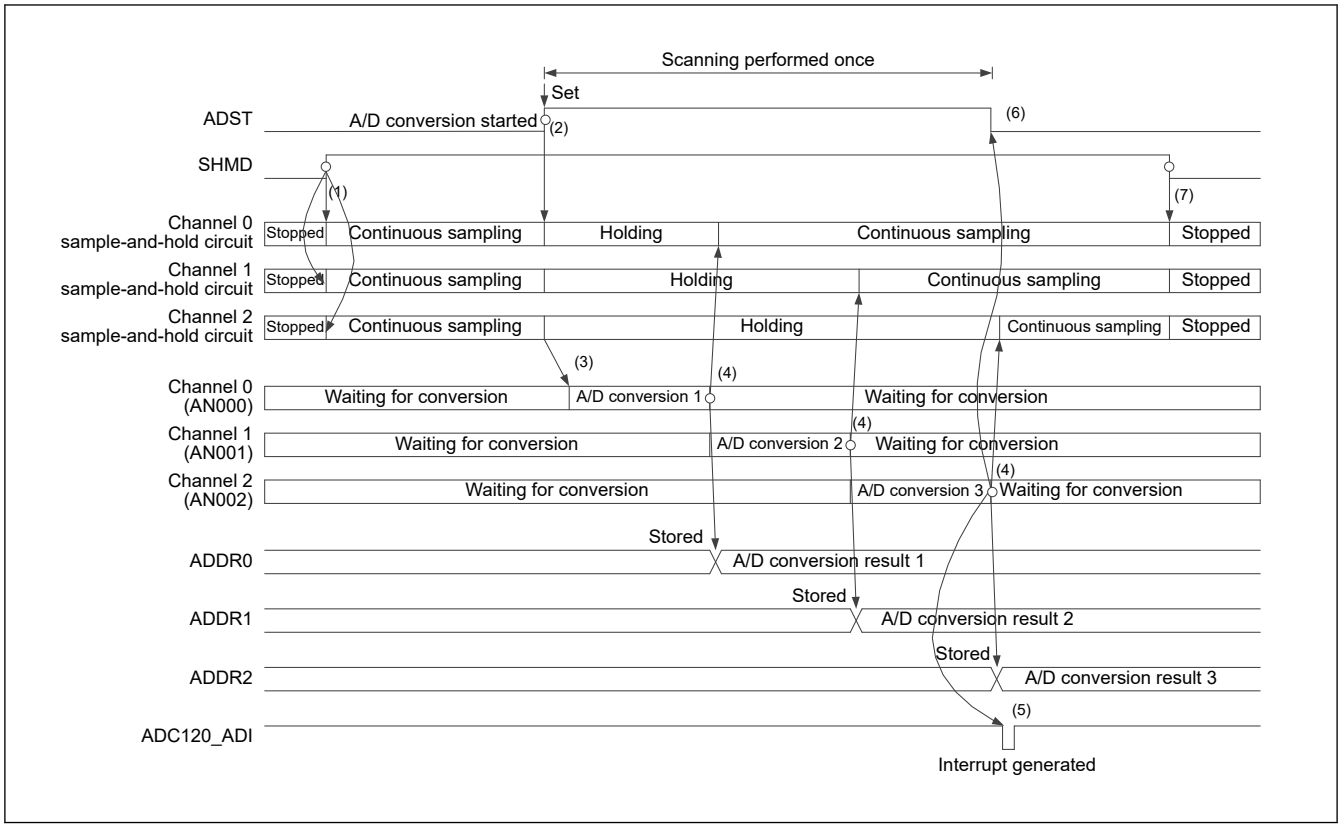
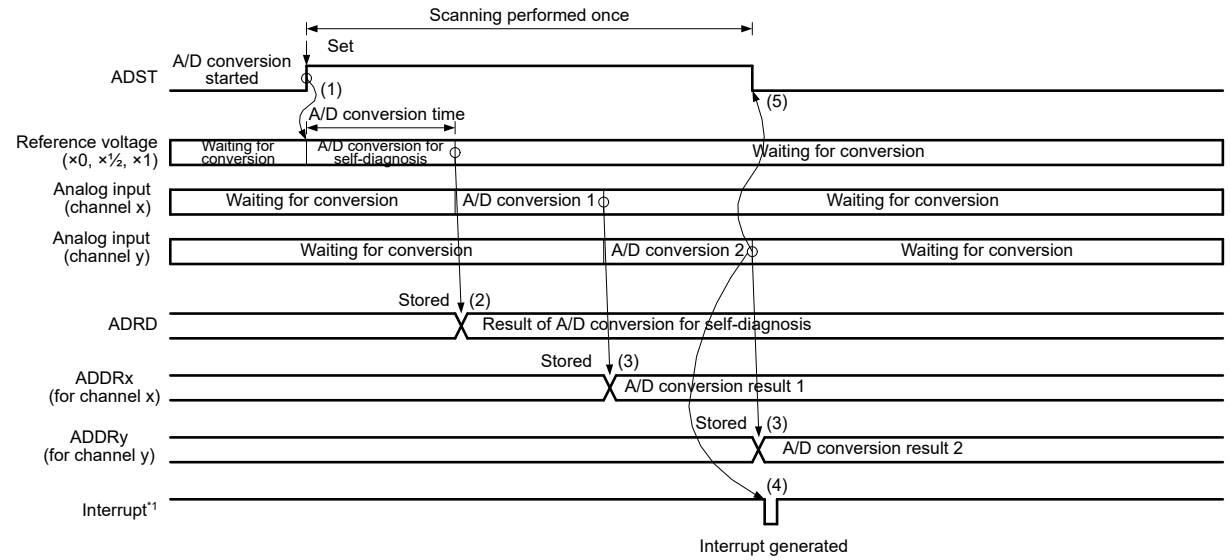


Figure 45.10 Example operation in single scan mode when channel-dedicated sample-and-hold circuits are used, AN000 to AN002 are selected, and continuous sampling is enabled

45.3.2.4 Channel Selection and Self-Diagnosis (Without Channel-Dedicated Sample-and-Hold Circuits)

When channels and self-diagnosis are selected, A/D conversion is first performed for the reference voltage VREFH0, VREFH (×0, ×1/2, or ×1), then A/D conversion is performed once on the analog input of the selected channels as follows:

1. A/D conversion for self-diagnosis is first started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger input, a synchronous trigger input (GPT, ELC), or an asynchronous trigger input.
2. When A/D conversion for self-diagnosis is completed, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
3. Each time A/D conversion of a single channel is completed, the A/D conversion result is stored in the associated A/D data register (ADDRy).
4. When A/D conversion of all the selected channels is completed, an ADC12i_ADI (i = 0, 1) interrupt request is generated.
5. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically set to 0 when A/D conversion of all the selected channels is completed. The ADC12 then enters a wait state.



Note: x and y (x < y) are the selected channel numbers by ADANSA0/1.
Note 1. ADC12i_ADI (i = 0, 1)

Figure 45.11 Example basic operation in single scan mode when the analog inputs (channel x and y) are selected with self-diagnosis

45.3.2.5 Channel Selection and Self-Diagnosis (With Channel-Dedicated Sample-and-Hold Circuits and Continuous Sampling Disabled)

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used while continuous sampling is disabled, the sample-and-hold operation is performed first, and then A/D conversion is performed once for the reference voltage VREFH0 (unit 0) (×0, ×1/2, or ×1) supplied to the ADC12. After that, A/D conversion is performed only once on the analog input of the selected channels.

The operation is as follows:

1. Analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (GPT, ELC), or asynchronous trigger input.
2. After the sample-and-hold operation, A/D conversion for self-diagnosis starts.
3. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
4. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
5. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0) interrupt request is generated (no register setting).
6. The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels completes. Then, the ADC12 enters a wait state.

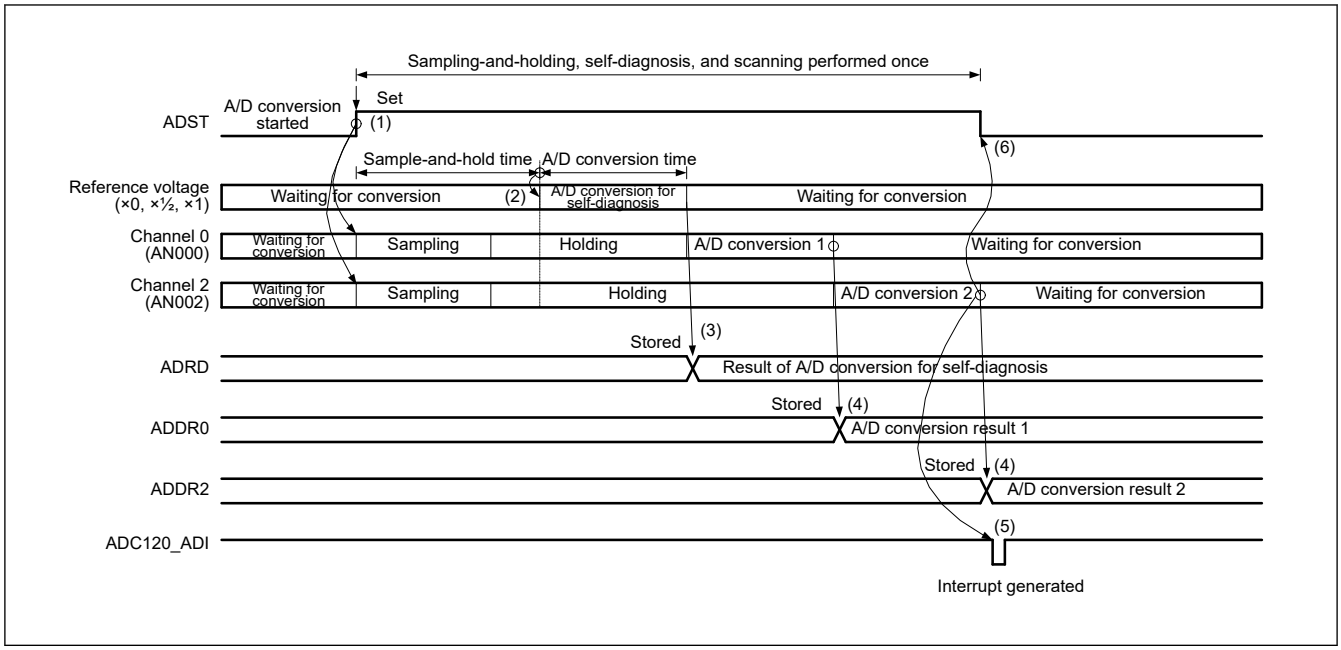


Figure 45.12 Example operation in single scan mode when channel-dedicated sample-and-hold circuits are used, AN000 and AN002 are selected with self-diagnosis, and continuous sampling is disabled

45.3.2.6 Channel Selection and Self-Diagnosis (With Channel-Dedicated Sample-and-Hold Circuits and Continuous Sampling Enabled)

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used while continuous sampling is enabled, sample-and-hold operations are performed first, followed by A/D conversion of the reference voltage VREFH0 (unit 0) supplied to the ADC12. After that, A/D conversion is performed only once on the analog input of the selected channels.

The operation is as follows:

1. When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuits selected in the ADSHCR.SHANS[2:0] bits start continuous sampling.
2. Analog input holding starts for all the channels for which the channel-dedicated sample-and-hold circuits are to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (GPT, ELC), or input of an asynchronous trigger. Set the ADCSR.ADST bit to 1 after at least 400 ns (when the permissible signal source impedance is 1 kΩ) elapse after the ADSHMSR.SHMD bit is set to 1.
3. After the stabilization time of the sample-and-hold circuits elapse, A/D conversion for self-diagnosis starts.
4. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
5. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy), and the sample-and-hold circuit restarts continuous sampling.
6. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0) interrupt request is generated (no register setting).
7. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels completes. Then, the ADC12 enters a wait state. If this is followed by single scanning, set the continuous sampling time for the sample-and-hold circuits to at least 400 ns (when the permissible signal source impedance is 1 kΩ).
8. When the ADSHMSR.SHMD bit is set to 0, the sample-and-hold circuits stop.

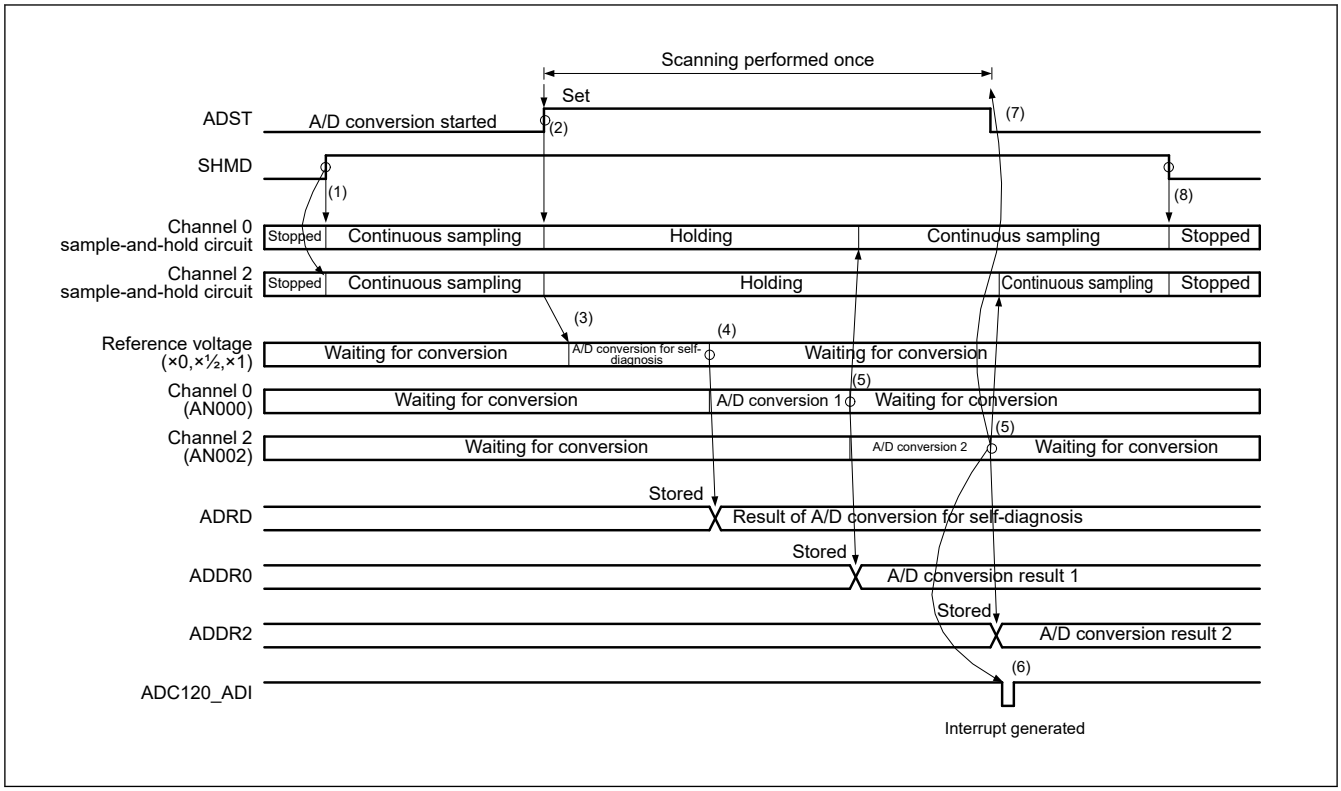


Figure 45.13 Example operation in single scan mode when channel-dedicated sample-and-hold circuits are used, AN000 to AN002 are selected with self-diagnosis, and continuous sampling is enabled

45.3.2.7 A/D Conversion of Temperature Sensor Output or Internal Reference Voltage

When the channels and temperature sensor output or internal reference voltage are selected at the same time, A/D conversion is performed first on the analog input of the selected channels, and once on the temperature sensor output or internal reference voltage. When both temperature sensor output and internal reference voltage are selected, A/D conversion of the temperature sensor output and internal reference voltage is performed, in that order. With the channels deselected, selecting only the temperature sensor output or internal reference voltage is also possible.

The operation is as follows:

1. When a software trigger, synchronous trigger (GPT, ELC), or asynchronous trigger sets the ADCSR.ADST bit to 1 (A/D conversion start), A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
2. On completion of A/D conversion on the channels, the result is stored in the associated A/D Data Register y (ADDRy), and then A/D conversion of the temperature sensor output starts.
3. On completion of A/D conversion of the temperature sensor output, the result is stored in the associated A/D Temperature Sensor Data Register (ADTSDR), and then A/D conversion of the internal reference voltage starts.
4. On completion of A/D conversion of the internal reference voltage, the result is stored in the associated A/D Internal Reference Voltage Data Register (ADOCDR), and an ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting).
5. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 on completion of A/D conversion. Then, the ADC12 enters a wait state.

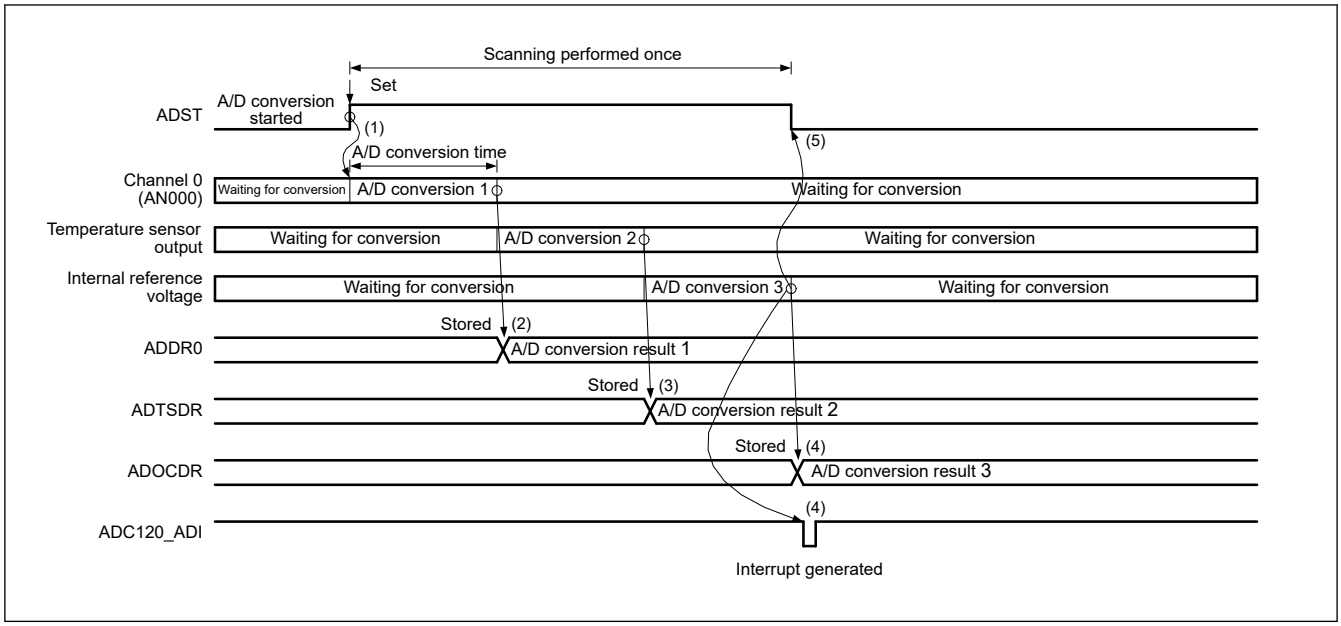


Figure 45.14 Example basic operation in single scan mode when AN000 and temperature sensor output or internal reference voltage are selected

45.3.2.8 A/D Conversion in Double-Trigger Mode

When double trigger mode is selected in single scan mode, two rounds of single scan operation started by a synchronous trigger (GPT, ELC) are performed in sequence.

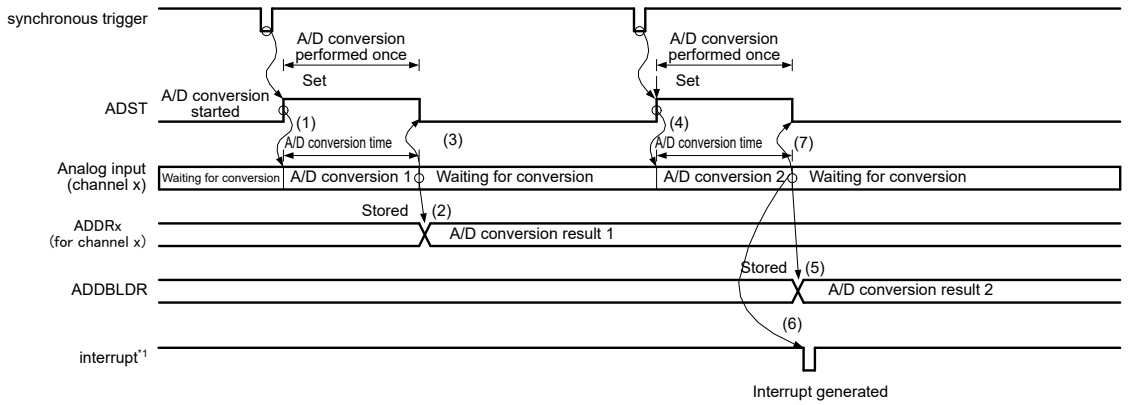
Deselect self-diagnosis and set the temperature sensor output A/D conversion select bit (ADEXICR.TSSA and ADEXICR.TSSB) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA and ADEXICR.OCSB) to 0.

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated in the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the ADCSR.DBLE bit is set to 1, channel selection using the ADANSA0 and ADANSA1 registers is invalid.

In double trigger mode, select a synchronous trigger (GPT, ELC) with the ADSTRGR.TRSA[5:0] bits. Additionally, set the ADCSR.EXTRG bit to 0 and the ADCSR.TRGE bit to 1. Do not use a software trigger.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a synchronous trigger input (GPT, ELC), A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
2. Each time A/D conversion of a single channel is completed, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
3. The ADCSR.ADST bit is automatically set to 0 and the ADC12 enters a wait state. An ADC12i_ADI (i = 0, 1) interrupt request is not generated.
4. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by the second trigger input, A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
5. When A/D conversion is completed, the result is stored in the A/D Data Duplexing Register (ADDBLDR), which is exclusively used in double-trigger mode.
6. An ADC12i_ADI (i = 0, 1) interrupt request is generated.
7. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically set to 0 when A/D conversion is completed. Then the ADC12 enters a wait state.



Note: x is the selected channel number by ADCSR.DBLANS[4:0].
Note 1. ADC12i_ADI (i = 0, 1)

Figure 45.15 Example operation in single scan mode when double-trigger mode is selected and the analog input (channel x) is duplicated

45.3.2.9 Extended Operations when Double-Trigger Mode is Selected

When double trigger mode is selected in single scan mode, and a synchronous trigger (ELC_AD00 (unit 0) and ELC_AD10 (unit 1)/ELC_AD01 (unit 0) and ELC_AD11 (unit 1), GTCIADA0 (unit 0) and GTCIADA4 (unit 1) / GTCIADB0 (Unit0) and GTCIADB4 (unit 1), GTCIADA1 (unit 0) and GTCIADA5 (unit 1) / GTCIADB1 (Unit0) and GTCIADB5 (unit 1), GTCIADA2 (unit 0) and GTCIADA6 (unit 1) / GTCIADB2 (Unit0) and GTCIADB6 (unit 1) or GTCIADA3 (unit 0) and GTCIADA7 (unit 1) / GTCIADB3 (Unit0) and GTCIADB7 (unit 1)) is selected as the trigger for the start of A/D conversion, two rounds of single scan operation are performed.

Deselect self-diagnosis and set the temperature sensor output A/D conversion select bit (ADEXICR.TSSA and ADEXICR.TSSB), and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA and ADEXICR.OCSB) to 0.

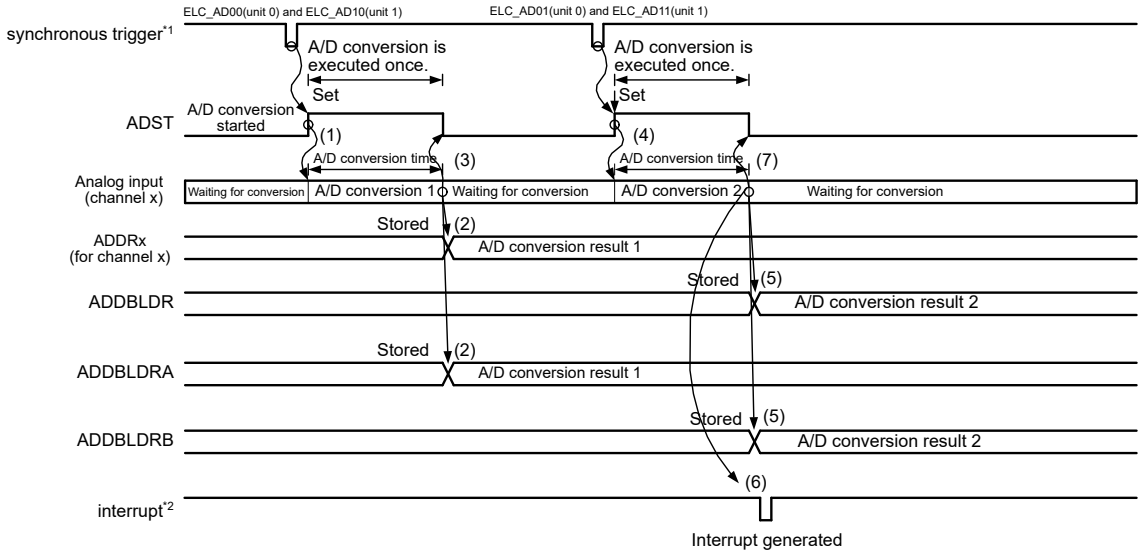
Duplication of A/D conversion data is enabled by setting the channel number to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the ADCSR.DBLE bit is set to 1, channel selection using the ADANSA0 and ADANSA1 registers is invalid.

In extended double trigger mode, select a synchronous trigger combination by setting the ADSTRGR.TRSA[5:0] bits to 0x0B, set the ADCSR.EXTRG bit to 0, and set the ADCSR.TRGE bit to 1. Do not use a software trigger.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a synchronous trigger input, A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
2. When A/D conversion completes, the A/D conversion result is stored in the associated A/D Data Register (ADDRy) and in A/D Data Duplexing Register A (ADDBLDRA) or A/D Data Duplexing Register B (ADDBLDRB) when the ELC_ADi0 or ELC_ADi1 trigger is input respectively (i = 0, 1).
3. The ADCSR.ADST bit is automatically set to 0 and the ADC12 enters a wait state. An ADC12i_ADI (i = 0, 1) interrupt request is not generated.
4. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by the second trigger (ELC_AD00 (unit 0) and ELC_AD10 (unit 1)/ELC_AD01 (unit 0) and ELC_AD11 (unit 1)), A/ D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
5. When A/D conversion completes, the A/D conversion result is stored in the A/D Data Duplexing Register (ADDBLDR) and in A/D Data Duplexing Register A (ADDBLDRA) or A/D Data Duplexing Register B (ADDBLDRB) when the ELC_ADi0 or ELC_ADi1 trigger is input respectively (i = 0, 1).
6. An ADC12i_ADI (i = 0, 1) interrupt request is generated.

7. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically set to 0 when A/D conversion completes. The ADC12 then enters a wait state.



Note: x is the selected channel number by ADANSA0/1.
Note 1. ELC_AD00 (unit 0) and ELC_AD10 (unit 1)/ELC_AD01 (unit 0) and ELC_AD11 (unit 1)
Note 2. ADC12i_ADI (i = 0, 1)

Figure 45.16 Example extended operation in double trigger mode with duplication selected for the analog input (channel x) and ELC_AD00 (unit 0) and ELC_AD10 (unit 1)/ELC_AD01 (unit 0) and ELC_AD11 (unit 1)

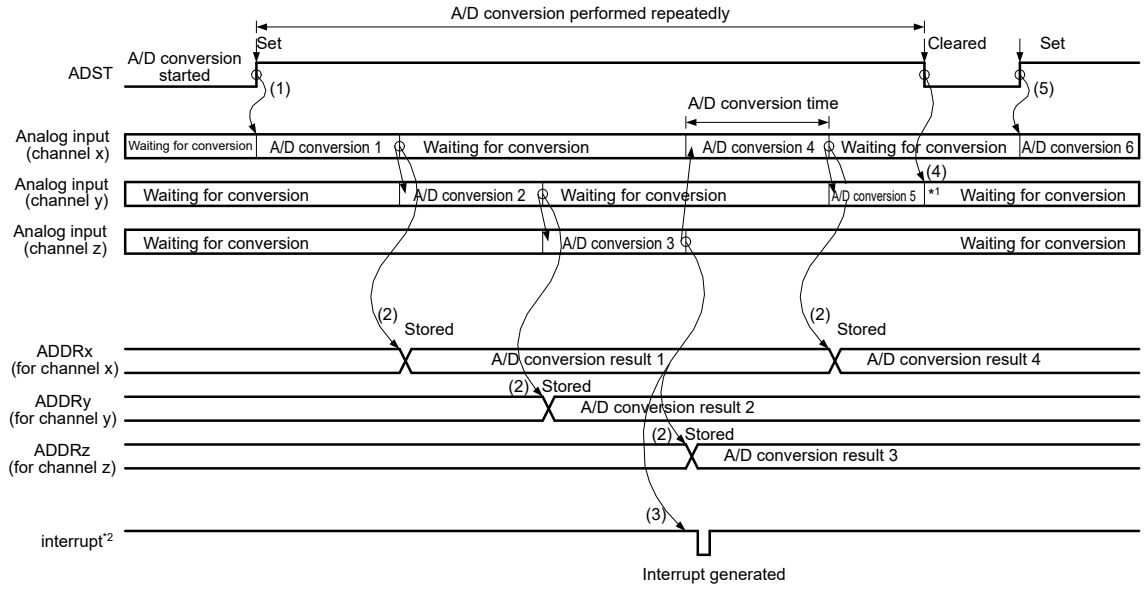
45.3.3 Continuous Scan Mode

45.3.3.1 Basic Operation (Without Channel-Dedicated Sample-and-Hold Circuits)

In continuous scan mode, A/D conversion is performed repeatedly on the analog input of the specified channels.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, a synchronous trigger input (GPT, ELC), or an asynchronous trigger input, A/D conversion is performed for ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
2. Each time A/D conversion of a single channel is completed, the A/D conversion result is stored in the associated A/D Data Register (ADDRy).
3. When A/D conversion of all the selected channels is completed, an ADC12i_ADI (i = 0, 1) interrupt request is generated. The ADC12 sequentially starts A/D conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
4. The ADCSR.ADST bit is not automatically cleared, and steps 2. and 3. are repeated as long as ADCSR.ADST remains 1 (A/ D conversion start). When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
5. When the ADCSR.ADST bit is later set to 1 (A/D conversion start), A/D conversion starts again for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.



Note: x, y, and z (x < y < z) are the selected channel numbers by ADANSA0/1.
Note 1. Data for A/D conversion 5 is ignored.
Note 2. ADC12i_ADI (i = 0, 1)

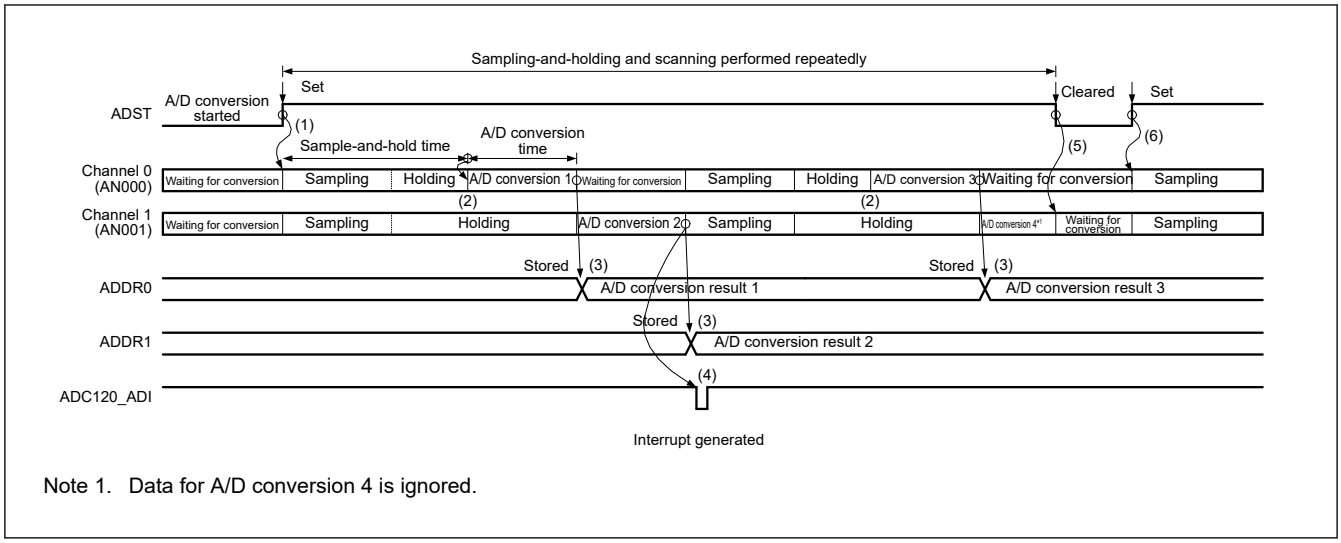
Figure 45.17 Example basic operation in continuous scan mode when the analog inputs (channel x to z) are selected

45.3.3.2 Basic Operation (With Channel-Dedicated Sample-and-Hold Circuits and Continuous Sampling Disabled)

When the channel-dedicated sample-and-hold circuit is used with the continuous sampling disabled, sample-and-hold operation is performed first, and then A/D conversion is repeated on the analog input of all the specified channels. The channels whose dedicated sample-and-hold circuit is to be used can be selected in the SHANS[2:0] bits in ADSHCR.

The operation is as follows:

1. Analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (GPT, ELC), or asynchronous trigger input.
2. After the sample-and-hold operation, A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
3. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
4. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0) interrupt request is generated (no register setting). At the same time, analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used.
5. The ADST bit is not automatically cleared, and steps 2 to 4 are repeated as long as the bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
6. When the ADST bit is then set to 1 (A/D conversion start), analog input sampling starts again for all the channels whose dedicated sample-and-hold circuit is to be used.



Note 1. Data for A/D conversion 4 is ignored.

Figure 45.18 Example operation in continuous scan mode when channel-dedicated sample-and-hold circuits are used and AN000 and AN001 are selected

45.3.3.3 Basic Operation (With Channel-Dedicated Sample-and-Hold Circuits and Continuous Sampling Enabled)

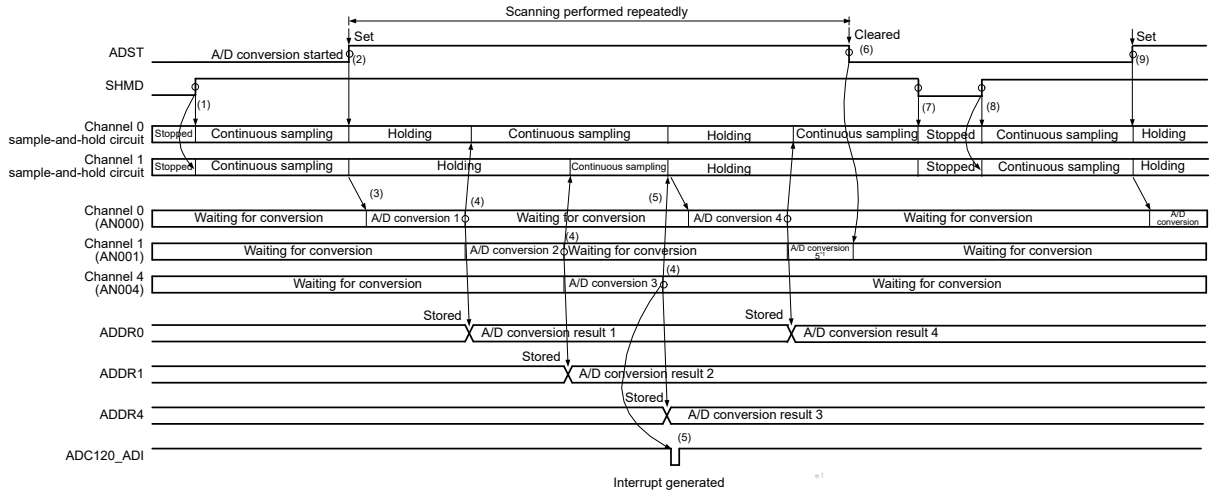
When a channel-dedicated sample-and-hold circuit is used while continuous sampling is enabled, sample-and-hold operations are performed first, after which the analog inputs on all the selected channels are A/D-converted as described in this section. The channels for which the channel-dedicated sample-and-hold circuits are to be used can be selected in the ADSHCR.SHANS[2:0] bits.

The operation is as follows:

1. When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuits selected in the ADSHCR.SHANS[2:0] bits start continuous sampling.
2. Analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (GPT, ELC), or input of an asynchronous trigger. Set the ADCSR.ADST bit to 1 after at least 400 ns (when the permissible signal source impedance is 1 kΩ) elapse after the ADSHMSR.SHMD bit is set to 1.
3. After the stabilization time of the sample-and-hold circuits elapses, A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
4. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy), and the sample-and-hold circuit restarts continuous sampling.
5. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0) interrupt request is generated (no register setting). Also, analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used.
6. The ADCSR.ADST bit is not automatically cleared, and steps 3 to 5 are repeated as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
7. When the ADSHMSR.SHMD bit is set to 0, the sample-and-hold circuits stop.
8. When the ADSHMSR.SHMD bit is then set to 1, the sample-and-hold circuits selected in the ADSHCR.SHANS[2:0] bits start continuous sampling.
9. When the ADCSR.ADST bit is then set to 1 (A/D conversion start), analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used.

Note: If continuous scanning is performed when only those channels with the sample-and-hold circuits are selected, time for continuous sampling cannot be secured in the second and subsequent continuous scans. When continuous sampling by the channel-dedicated sample-and-hold circuits is enabled for continuous scanning, select one or more channels among AN003 to AN008 and AN016 to AN019, temperature sensor output, internal reference voltage and VBATT 1/3 voltage monitor output for unit 0, and AN100 to AN106 and AN116 to AN122, temperature sensor output,

internal reference voltage and VBATT 1/3 voltage monitor output for unit 1, and set the continuous sampling time for the sample-and-hold circuits to at least 400 ns (when the permissible signal source impedance is 1 kΩ).



Note 1. Data for A/D conversion 5 is ignored.

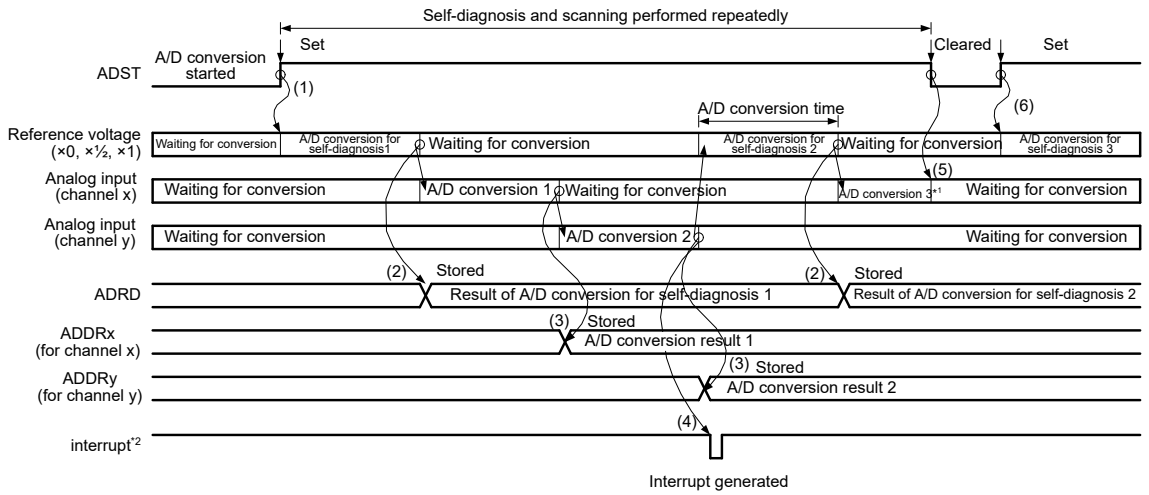
Figure 45.19 Example operation in continuous scan mode when channel-dedicated sample-and-hold circuits are used, AN000, AN001, and AN004 are selected, and continuous sampling is enabled

45.3.3.4 Channel Selection and Self-Diagnosis (Without Channel-Dedicated Sample-and-Hold Circuits)

When channels and self-diagnosis are selected at the same time, A/D conversion is first performed for the reference voltage VREFH0, VREFH (×0, ×1/2, or ×1) supplied to the ADC12, and A/D conversion is performed on the analog input of the selected channels. This sequence is repeated as described in the section that follows.

The operation is as follows:

1. A/D conversion for self-diagnosis is first started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger input, a synchronous trigger input (GPT, ELC), or an asynchronous trigger input.
2. When A/D conversion for self-diagnosis is completed, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
3. Each time A/D conversion of a single channel is completed, the A/D conversion result is stored in the corresponding A/D Data Register (ADDRy).
4. When A/D conversion of all the selected channels is completed, an ADC12i_ADI (i = 0, 1) interrupt request is generated. At the same time, the ADC12 starts A/D conversion for self-diagnosis and then on the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
5. The ADCSR.ADST bit is not automatically cleared, and steps 2. to 4. are repeated as long as the ADCSR.ADST bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
6. When the ADST bit is later set to 1 (A/D conversion start), the A/D conversion for self-diagnosis is started again.



Note: x and y (x < y) are the selected channel numbers by ADANSA0/1.
Note 1. Data for A/D conversion 3 is ignored.
Note 2. ADC12i_ADI (i = 0, 1)

Figure 45.20 Example basic operation in continuous scan mode when the analog inputs (channel x and y) are selected with self-diagnosis

45.3.3.5 Channel Selection and Self-Diagnosis (With Channel-Dedicated Sample-and-Hold Circuits and Continuous Sampling Disabled)

When the channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used while continuous sampling is disabled, sample-and- hold operation is performed first, and then A/D conversion is performed for the reference voltage VREFH0 (unit 0) (×0, ×1/2, or ×1) supplied to the ADC12, and then A/D conversion is performed on the analog input of the selected channels, and this sequence is repeated.

The operation is as follows:

1. Analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (GPT, ELC), or asynchronous trigger input.
2. After the sample-and-hold operation, A/D conversion for self-diagnosis starts.
3. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
4. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D data register (ADDRy).
5. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0) interrupt request is generated (no register setting). At the same time, analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used.
6. The ADST bit is not automatically cleared, and steps 2 to 5 are repeated as long as the bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
7. When the ADST bit is then set to 1 (A/D conversion start), analog input sampling starts again for all the channels whose dedicated sample-and-hold circuits are to be used.

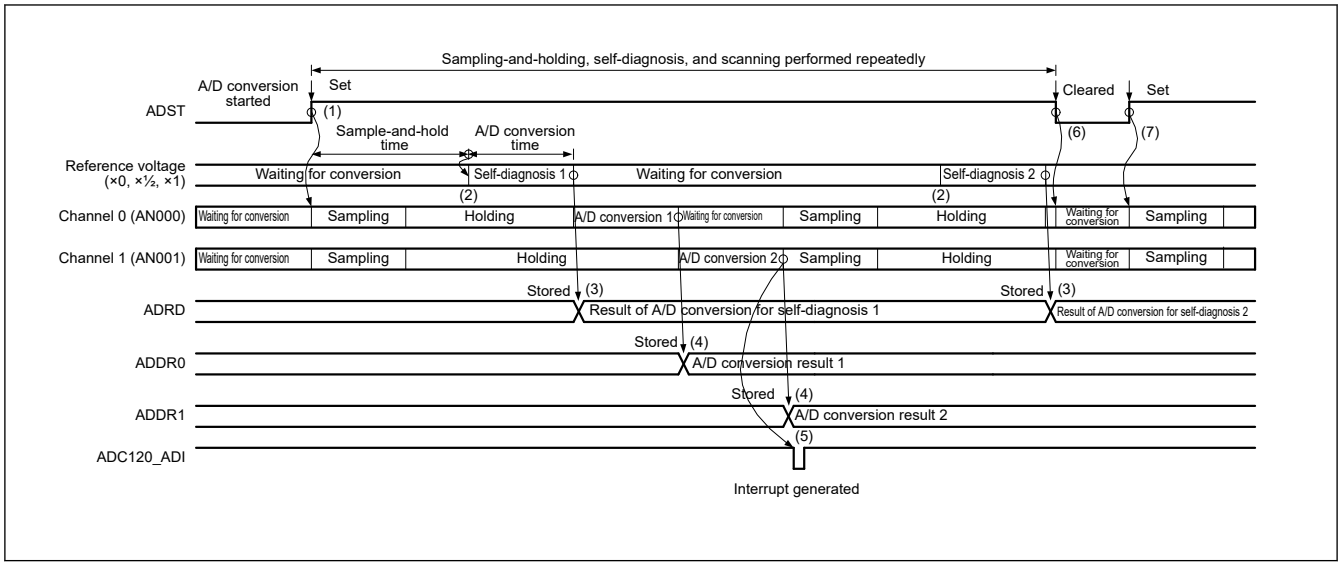


Figure 45.21 Example operation in continuous scan mode when channel-dedicated sample-and-hold circuits are used and AN000 and AN001 are selected with self-diagnosis

45.3.3.6 Channel Selection and Self-Diagnosis (With Channel-Dedicated Sample-and-Hold Circuits and Continuous Sampling Enabled)

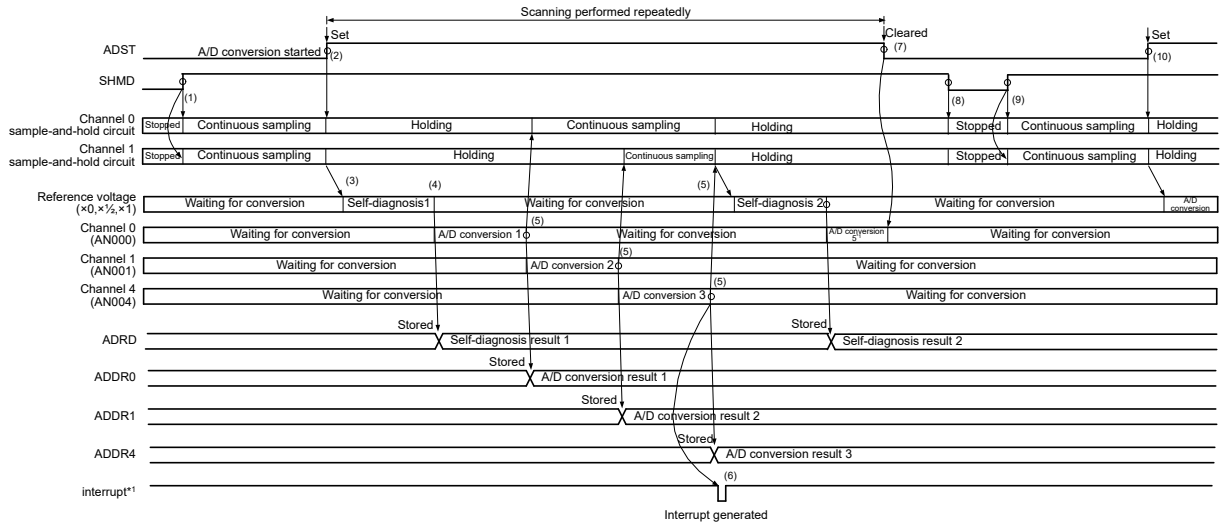
When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used while continuous sampling is enabled, the sample-and-hold operation is performed first, and followed by A/D conversion of the reference voltage VREFH0 (unit 0) (×0, ×1/2, or ×1) supplied to the ADC12. After that, A/D conversion is performed on the analog input of the selected channels, and this sequence is repeated.

The operation is as follows:

1. When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuits selected in the ADSHCR.SHANS[2:0] bits start continuous sampling.
2. Analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (GPT, ELC), or input of an asynchronous trigger. Set the ADCSR.ADST bit to 1 after at least 400 ns (when the permissible signal source impedance is 1 kΩ) elapse after the ADSHMSR.SHMD bit is set to 1.
3. After the stabilization time of the sample-and-hold circuits elapses, A/D conversion for self-diagnosis starts.
4. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
5. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy), and the sample-and-hold circuit restarts continuous sampling.
6. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0) interrupt request is generated (no register setting). Also, analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used.
7. The ADCSR.ADST bit is not automatically cleared, and steps 3 to 6 are repeated as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
8. When the ADSHMSR.SHMD bit is set to 0, the sample-and-hold circuits stop.
9. When the ADSHMSR.SHMD bit is then set to 1, the sample-and-hold circuits selected in the ADSHCR.SHANS[2:0] bits start continuous sampling.
10. When the ADCSR.ADST bit is then set to 1 (A/D conversion start), analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used.

Note: If continuous scanning is performed when only those channels with the sample-and-hold circuits are selected, time for continuous sampling cannot be secured in the second and subsequent continuous scans. When continuous sampling by the channel-dedicated sample-and-hold circuits is enabled for continuous scanning, select one or more

channels among AN003 to AN008 and AN016 to AN019, temperature sensor output, internal reference voltage and VBATT 1/3 voltage monitor output for unit 0, and AN100 to AN106 and AN116 to AN122, temperature sensor output, internal reference voltage and VBATT 1/3 voltage monitor output for unit 1, and set the continuous sampling time for the sample-and-hold circuits to at least 400 ns (when the permissible signal source impedance is 1 kΩ).



Note: Data for A/D conversion 5 is ignored.
Note 1. ADC12i_ADI (i = 0)

Figure 45.22 Example operation in continuous scan mode when channel-dedicated sample-and-hold circuits are used, AN000, AN001, and AN004 are selected with self-diagnosis, and continuous sampling is enabled

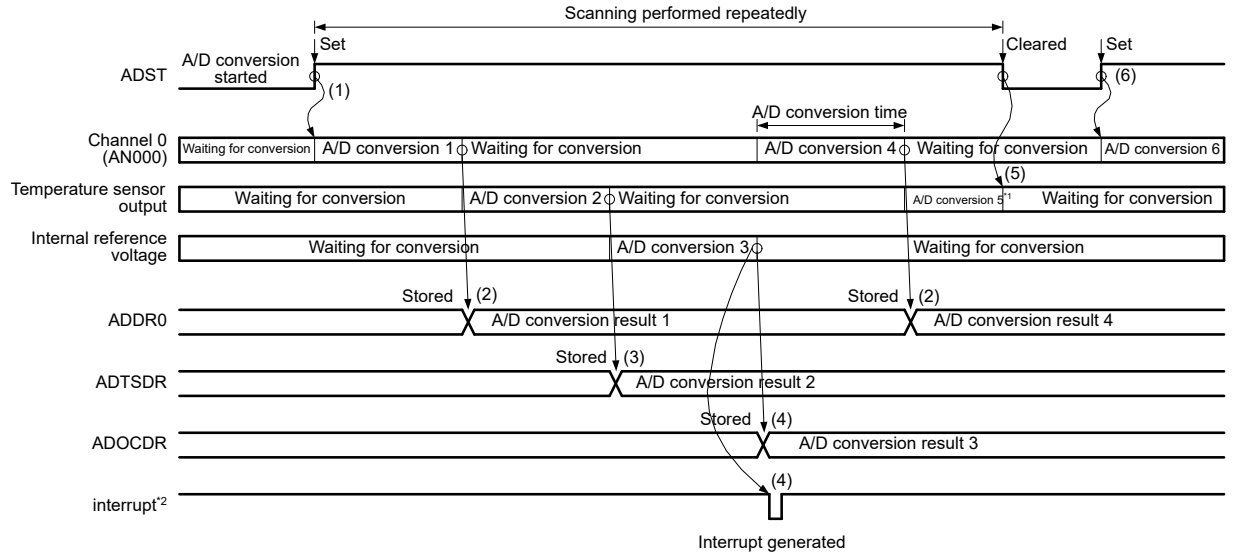
45.3.3.7 A/D Conversion of Temperature Sensor Output or Internal Reference Voltage

When the channels and temperature sensor output or internal reference voltage are selected at the same time, A/D conversion is first performed on the analog input of the selected channels, and then the A/D conversion of the temperature sensor output or internal reference voltage is repeated. When both the temperature sensor output and internal reference voltage are selected, A/D conversion of the temperature sensor output and internal reference voltage is performed, in that order.

With the channels deselected, selecting only the temperature sensor output or internal reference voltage is also possible.

The operation is as follows:

1. When a software trigger, synchronous trigger (GPT, ELC), or asynchronous trigger sets the ADCSR.ADST bit to 1 (A/D conversion start), A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
2. On completion of A/D conversion on the channels, the result is stored in the associated A/D Data Register y (ADDRy), and then A/D conversion of temperature sensor output starts.
3. On completion of A/D conversion of the temperature sensor output, the result is stored in the associated A/D Temperature Sensor Data Register (ADTSDR), and then A/D conversion of the internal reference voltage starts.
4. On completion of A/D conversion of the internal reference voltage, the result is stored in the associated A/D Internal Reference Voltage Data Register (ADOCDR), and an ADC12i_ADI (i = 0, 1) interrupt request is generated. In addition, the ADC12 continuously starts A/D conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the lowest number n.
5. The ADCSR.ADST bit is not cleared automatically, and steps 2 to 4 are repeated as long as this bit remains set to 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
6. When the ADCSR.ADST bit is then set to 1 (A/D conversion start), A/D conversion starts again for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the lowest number n.



Note 1. Data for A/D conversion 5 is ignored.
Note 2. ADC12i_ADI (i = 0, 1)

Figure 45.23 Example basic operation in continuous scan mode when AN000 and temperature sensor output or internal reference voltage are selected

45.3.4 Group Scan Mode

45.3.4.1 Basic Operation

In group scan mode, A/D conversion is performed once on the analog input of all the specified channels in group A and B after scanning is started by a synchronous trigger (GPT, ELC). The scan operation of each group is similar to the scan operation in single scan mode.

The synchronous triggers can be selected in the ADSTRGR.TRSA[5:0] bits for group A and in the ADSTRGR.TRSB[5:0] bits for group B. Use different triggers for group A and B to prevent simultaneous A/D conversion of the two groups. Do not use a software trigger.

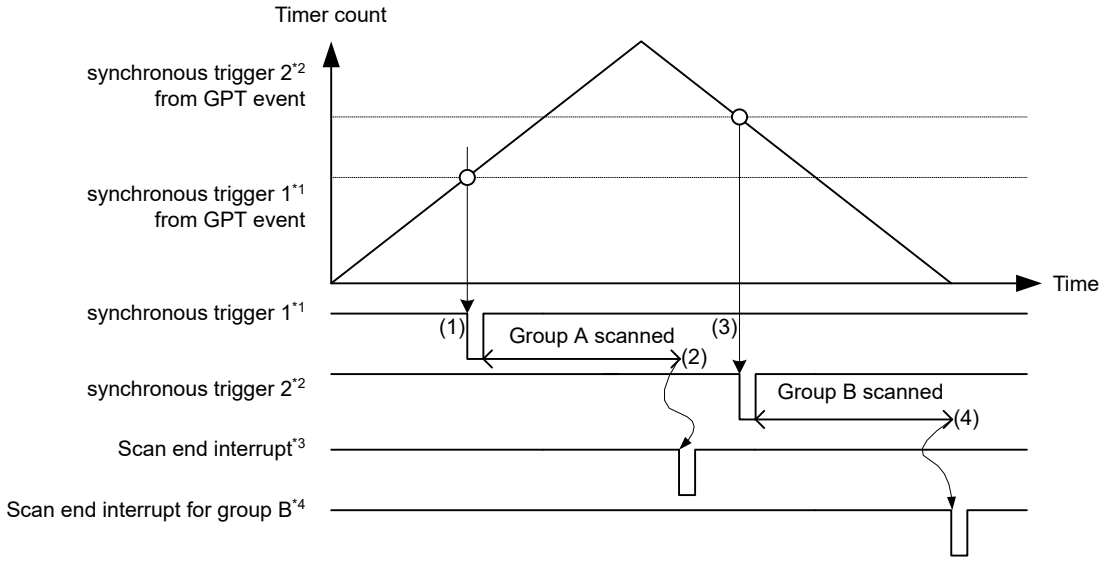
The group A channels to be A/D-converted are selected using the ADANSA0 and ADANSA1 registers and the ADEXICR.TSSA and OCSA bits. The group B channels to be A/D-converted are selected using the ADANSB0 and ADANSB1 registers and the ADEXICR.TSSB and OCSB bits. Group A and B cannot use the same channels.

When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed for Group A and B.

The following sequence describes operation in group scan mode using a synchronous trigger from the ELC. In this example, the ELC_AD00 (unit 0) and ELC_AD10 (unit 1) trigger from the ELC is used to start conversion of group A and the ELC_AD01 (unit 0) and ELC_AD11 (unit 1) trigger from the ELC is used to start conversion of group B. In addition, ELC_AD00 (unit 0) and ELC_AD10 (unit 1) and ELC_AD01 (unit 0) and ELC_AD11 (unit 1) are selected for the GPT event in the associated ELC.ELSRn registers.

The operation is as follows:

1. Scanning of group A is started by ELC_AD00 (unit 0) and ELC_AD10 (unit 1).
2. When group A scanning completes, an ADC12i_ADI (i = 0, 1) interrupt is generated (no register setting).
3. Scanning of group B is started by ELC_AD01 (unit 0) and ELC_AD11 (unit 1).
4. When group B scanning completes, an ADC12i_GBADI (i = 0, 1) interrupt is generated if the ADCSR.GBADIE bit is 1 (ADC12i_GBADI (i = 0, 1) interrupt when scanning completion is enabled).



Note 1. ELC_AD00 (unit 0) and ELC_AD10 (unit 1)
Note 2. ELC_AD01 (unit 0) and ELC_AD11 (unit 1)
Note 3. ADC12i_ADI (i = 0, 1)
Note 4. ADC12i_GBADI (i = 0, 1)

Figure 45.24 Example basic operation in group scan mode when synchronous triggers from the ELC are used

45.3.4.2 A/D Conversion in Double-Trigger Mode

When double trigger mode is selected in group scan mode, two rounds of single scan operation started by a synchronous trigger (GPT, ELC) are performed as a sequence for group A. For group B, single scan operation started by a synchronous trigger (GPT, ELC) is performed once.

In group scan mode, the synchronous trigger can be selected in the ADSTRGR.TRSA[5:0] bits for group A and in the ADSTRGR.TRSB[5:0] bits for group B. Use different triggers for group A, B to prevent simultaneous A/D conversion of the two groups. Do not use a software trigger or an asynchronous trigger.

When an ELC_AD00 (unit 0) and ELC_AD10 (unit 1)/ELC_AD01 (unit 0) and ELC_AD11 (unit 1) is selected as group A synchronous triggers by setting the ADSTRGR.TRSA[5:0] bits to 0x0B, operation proceeds in extended double trigger mode.

The group A channel to be A/D-converted is selected using the DBLANS[4:0] bits in the ADCSR register, while the group B channels to be A/D-converted are selected using the ADANSB0 and ADANSB1 registers. Group A, B cannot use the same channels.

When double-trigger mode is selected in group scan mode, set the A/D conversion select bits for both the temperature sensor output (ADEXICR.TSSA) and the internal reference voltage (ADEXICR.OCSA) to 0 (deselected).

Self-diagnosis cannot be selected when double trigger mode is selected in group scan mode.

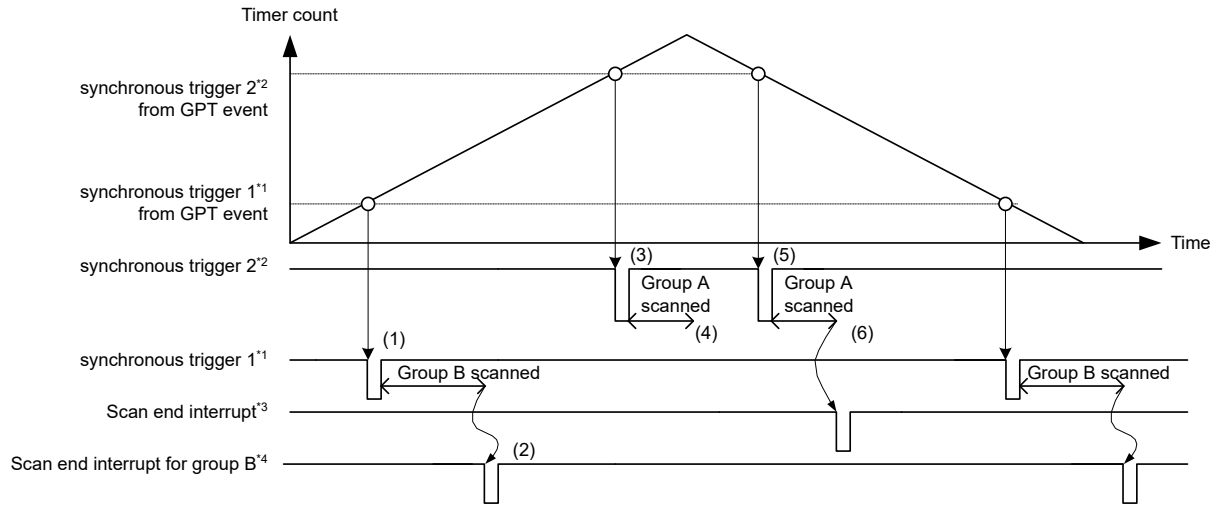
Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated in the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1.

The following sequence describes operation in group scan mode with double trigger mode selected and using a synchronous trigger from the ELC. In this example, the ELC_AD00 (unit 0) and ELC_AD10 (unit 1) trigger is used to start conversion of group A and the ELC_AD01 (unit 0) and ELC_AD11 (unit 1) trigger is used to start conversion of group B. In addition, ELC_AD00 (unit 0) and ELC_AD10 (unit 1) and ELC_AD01 (unit 0) and ELC_AD11 (unit 1) are selected for the GPT event in the associated ELC.ELSRn registers.

The operation is as follows:

1. Scanning of group B is started by the ELC_AD00 (unit 0) and ELC_AD10 (unit 1) trigger from the ELC.

2. When group B scanning completes, an ADC12i_GBADI (i = 0, 1) interrupt is generated if the GBADIE bit in ADCSR is 1 (ADC12i_GBADI (i = 0, 1) interrupt when scanning completion is enabled).
3. The first scan of group A is started by the first ELC_AD01 (unit 0) and ELC_AD11 (unit 1) trigger.
4. When the first scan of group A completes, the conversion result is stored in the associated A/D Data Register y (ADDRy). An ADC12i_ADI (i = 0, 1) interrupt request is not generated.
5. The second scan of group A is started by the second ELC_AD01 (unit 0) and ELC_AD11 (unit 1) trigger.
6. When the second scan of group A completes, the conversion result is stored in ADDBLDR. An ADC12i_ADI (i = 0, 1) interrupt is generated.



Note 1. ELC_AD00 (unit 0) and ELC_AD10 (unit 1)
Note 2. ELC_AD01 (unit 0) and ELC_AD11 (unit 1)
Note 3. ADC12i_ADI (i = 0, 1)
Note 4. ADC12i_GBADI (i = 0, 1)

Figure 45.25 Example basic operation in group scan mode with double-trigger mode when synchronous triggers from the ELC are used

45.3.4.3 Group Priority Operation

Group priority operation is performed by setting the ADGSPCR.PGS bit to 1 in group-scan mode. The priority of groups is group A > group B.

When setting the PGS bit in the ADGSPCR register to 1, follow the procedure described in Figure 45.26. If the procedure is not followed, A/D conversion operation and stored data are not guaranteed.

As the basic operation in group-scan mode, a trigger input generated during A/D conversion of group A, B is ignored, and the A/D conversion operation of each group is similar to the operation in single-scan mode.

In group priority operation, if a trigger for a priority group is input during scanning of a lower-priority group, A/D conversion for the lower-priority group is stopped and A/D conversion for the priority group is performed.

If the setting of the ADGSPCR.GBRSCN bit is 0, the lower-priority group enters a wait state when A/D conversion for the priority group completes. A trigger input of the lower-priority group generated during A/D conversion is ignored.

If the setting of the ADGSPCR.GBRSCN bit is 1, A/D conversion for the lower-priority group automatically restarts upon completion of A/D conversion for the priority group. A trigger input of the lower-priority group generated during A/D conversion on the priority group takes effect, and A/D conversion for the lower-priority group is automatically performed upon completion of A/D conversion on the priority group.

If the ADGSPCR.GBRSCN bit is 1 and the ADGSPCR.LGRRS bit is 0, A/D conversion for the lower-priority group is restarted from the first channel. If the setting of the ADGSPCR.LGRRS bit is 1, A/D conversion for the lower-priority group is restarted from the channel for which the conversion stopped. However, if the self-diagnosis function is used, the A/D conversion is restarted from the channel for which the conversion stopped after self-diagnosis completed.

Table 45.26 summarizes operations in response to the input of a trigger during A/D conversion with the settings of the ADGSPCR.GBRSCN bit.

If the setting of the ADGSPCR.GBRP bit is 1, A/D conversion operation for the lowest-priority group is to continuously perform single scans.

For the trigger settings in group-scan mode, select a synchronous trigger for group A by using the ADSTRGR.TRSA[5:0] bits, a synchronous trigger for group B by using the ADSTRGR.TRSB[5:0] bits. Each trigger must be different from each other. Set the ADSTRGR.TRSB[5:0] bits to 0x3F when setting the ADGSPCR.GBRP bit to 1.

The channels to be scanned must be selected in the registers shown in section 45.3.4. Group Scan Mode.

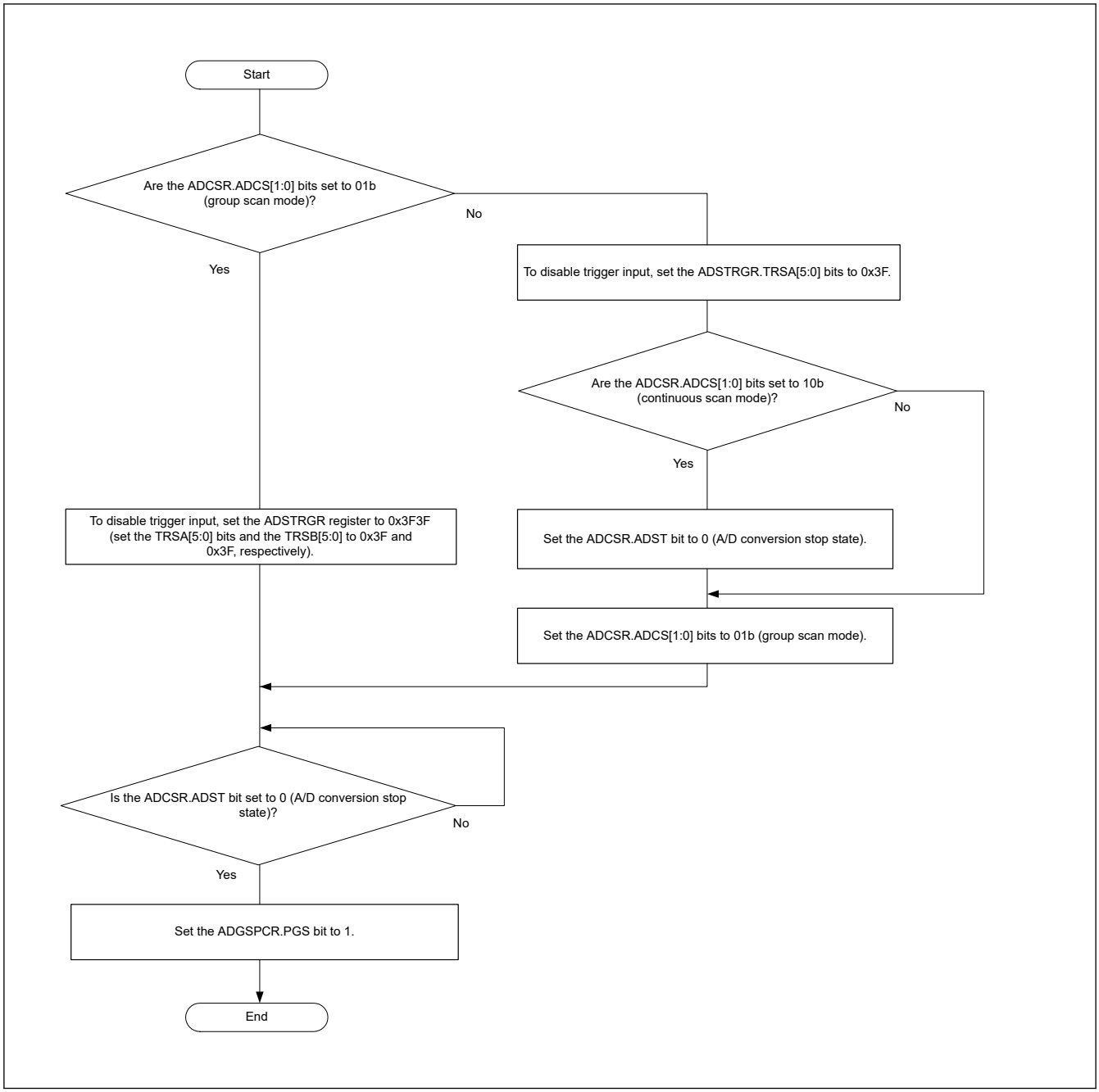


Figure 45.26 Flowchart for ADGSPCR.PGS bit setting

Table 45.26 Control of A/D conversion operations according to ADGSPCR.GBRSCN bit setting

A/D conversion operation	Trigger input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for group A is in progress	Input of trigger for group A	Trigger input is ineffective.	Trigger input is ineffective.
	Input of trigger for group B	Trigger input is ineffective.	A/D conversion for group B is performed after A/D conversion for group A completes.
When A/D conversion for group B is in progress	Input of trigger for group A	A/D conversion for group B is discontinued and A/D conversion for group A starts.	• A/D conversion for group B is discontinued and A/D conversion for group A starts. • A/D conversion for group B starts after A/D conversion for group A completes.
	Input of trigger for group B	Trigger input is ineffective.	Trigger input is ineffective.

To use group priority operation mode, select the operation mode to be implemented and set the registers according to the following table.

Table 45.27 Group priority operation setting and operation mode for two groups (ADGSPCR.PGS = 1)

ADGSPCR			Operation category
GBRSCN	LGRRS	GBRP	
0	x	0	Group priority operation for two groups (groups A and B) • When a trigger of group A is input, A/D conversion for group B is terminated (and will not be restarted).
1	0	0	Group priority operation for two groups (groups A and B) • After A/D conversion for group B stopped, when A/D conversion for group A completes, A/D conversion for the group B channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order of smaller channel number.
1	1	0	Group priority operation for two groups (groups A and B) • After A/D conversion for group B stopped, when A/D conversion for group A completes, A/D conversion for the group B channels selected in the ADANSB0/1 register restarts according to the conversion order of smaller channel number, beginning from the channel for which A/D conversion stopped.*1
x	0	1	Group priority operation for two groups (groups A and B) • Single scanning for group B is continuously performed without a start trigger input. After A/D conversion for group B stopped, when A/D conversion for group A completes, single scanning for the channels selected in the ADANSB0/1 register restarts according to the conversion order of smaller channel number.
1	1	1	Group priority operation for two groups (groups A and B) • Single scanning for group B is continuously performed without a start trigger input. After A/D conversion for group B stopped, when A/D conversion for group A completes, single scanning for the channels selected in the ADANSB0/1 register restarts according to the conversion order of smaller channel number, beginning from the channel for which A/D conversion stopped.*1

Note: x: Don't care.
Note 1. When the self-diagnosis function is enabled (ADCER.DIAGM = 1), A/D conversion for the channel that has been stopped is started after self-diagnosis is performed.

(1) Group priority operation for two groups (when ADGSPCR.PGS = 1)

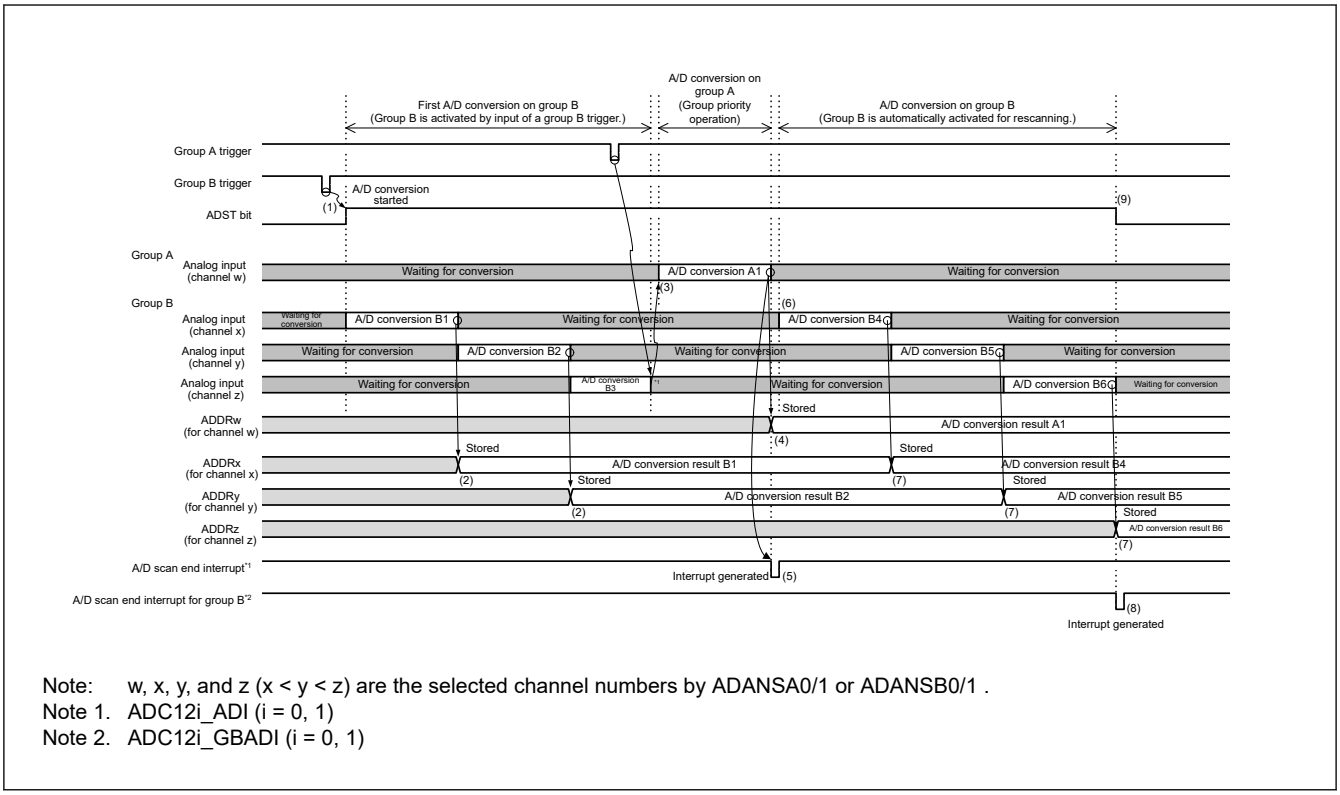
Operation examples 1-1 to 1-3 show group priority operations in group-scan mode (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0).

Operation example 1-1: "Group A trigger input during group B scan" when rescanning is enabled

1. When input of a trigger for group B sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion for the analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order from the channel with the smallest number n.
2. On completion of A/D conversion for each channel in group B, the result is stored in the corresponding A/D Data Register y (ADDRy).
3. When a trigger for group A is input during A/D conversion for group B, A/D conversion for group B stops while the ADCSR.ADST bit remains 1. Then A/D conversion for the group A analog input channels selected in the ADANSA0

and ADANSA1 registers starts according to the conversion order from the channel with the smallest number n. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D Data Register y (ADDRy).

4. On completion of A/D conversion on the channels, the result is stored in the corresponding A/D Data Register y (ADDRy).
5. An ADC120_ADI interrupt request is generated.
6. If the setting of the ADGSPCR.GBRSCN bit is 1 (enabling rescanning of the group that was stopped in group priority operation), A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order from the channel with the smallest number n while the ADCSR.ADST remains 1.
7. On completion of A/D conversion on the channels, the result is stored in the corresponding A/D Data Register y (ADDRy).
8. If the setting of the ADCSR.GBADIE bit is 1 (enabling interrupt generation on completion of group B scan), a group B scan end interrupt request is generated.
9. When A/D conversion for all the channels completes, the ADCSR.ADST bit is automatically cleared and the A/D converter enters a wait state.



Note: w, x, y, and z (x < y < z) are the selected channel numbers by ADANSA0/1 or ADANSB0/1 .
Note 1. ADC12i_ADI (i = 0, 1)
Note 2. ADC12i_GBADI (i = 0, 1)

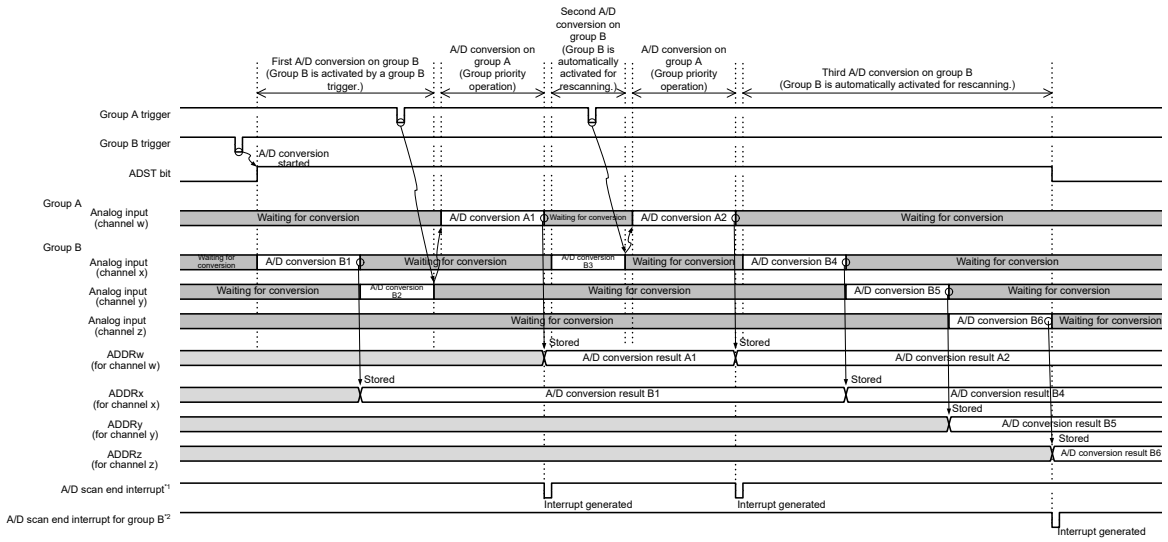
Figure 45.27 Example of group priority operation 1-1: Group A trigger input during group B scanning when rescanning is enabled (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0)

Operation example 1-2: “Group A trigger input during rescanning of group B” when rescanning is enabled

Figure 45.28 shows the operation when a group A trigger is input during rescanning operation for group B.

Even during rescanning operation, when a trigger for group A is input, A/D conversion on group B stops and A/D conversion for group A starts. A/D conversion for group B starts after A/D conversion for group A completes.

Operations for setting the ADCSR.ADST bit, storing the A/D conversion result in the corresponding A/D Data Register y (ADDRy), and generating interrupt requests are the same as those in operation example 1-1.



Note: w, x, y, and z (x < y < z) are the selected channel numbers by ADANSA0/1 or ADANSB0/1.
Note 1. ADC12i_ADI (i = 0, 1)
Note 2. ADC12i_GBADI (i = 0, 1)

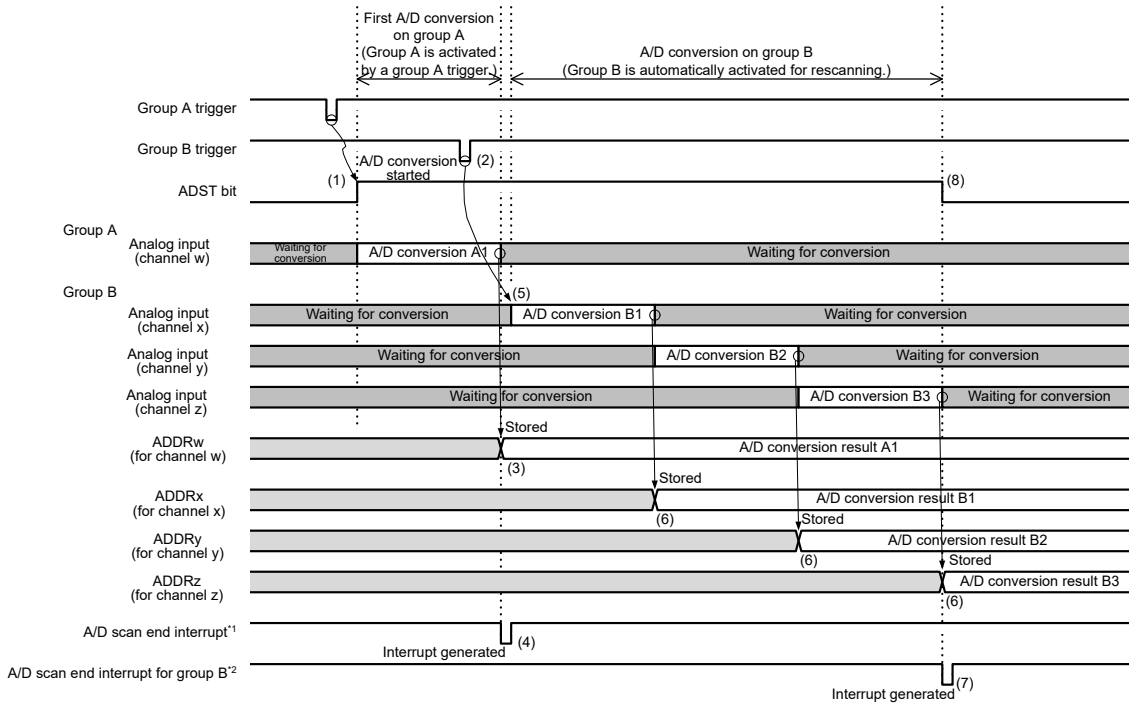
Figure 45.28 Example of group priority operation 1-2: Group A trigger input during rescanning of group B when rescanning is enabled (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0)

Operation example 1-3: “Group B trigger input during group A scan” when rescanning is enabled

The following describes the operation when the setting of the ADGSPCR.GBRSCN bit is 1 (enabling rescanning of the group that was stopped in group priority operation) and a trigger for group B is input during scanning operation for group A.

If the setting of the ADGSPCR.GBRSCN bit is 0, any trigger for group B that is input during scanning operation for group A is invalid.

1. When input of a trigger for group A sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion for the group A analog input channels selected in the ADANSA0 and ADANSA1 registers starts according to the conversion order from the channel with the smallest number n.
2. When a trigger for group B is input during A/D conversion for group A, group B is ready for A/D conversion.
3. On completion of A/D conversion for each channel in group A, the result is stored in the corresponding A/D Data Register y (ADDRy).
4. An ADC120_ADI interrupt request is generated.
5. When A/D conversion for group A completes, while the ADCSR.ADST bit remains 1, A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order from the channel with the smallest number n.
   As with the case of operation example 1-1, if a trigger for group A is input during A/D conversion for group B, A/D conversion for group A starts. Then A/D conversion for group B starts upon completion of A/D conversion for group A.
6. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
7. Upon completion of A/D conversion for group B, a group B scan end interrupt request is generated if the setting of the ADCSR.GBADIE bit is 1 (enabling interrupt generation on completion of group B scan).
8. When A/D conversion for all the channels completes, the ADCSR.ADST bit is automatically cleared and the A/D converter enters a wait state.



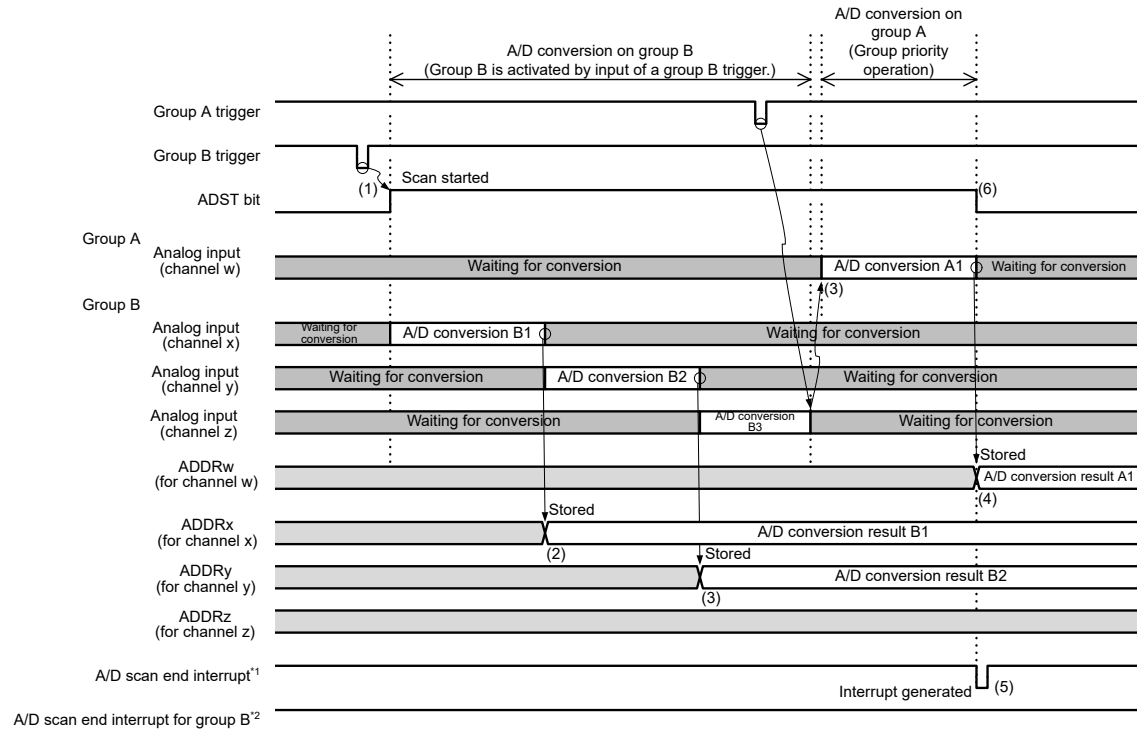
Note: w, x, y, and z (x < y < z) are the selected channel numbers by ADANSA0/1 or ADANSB0/1.
Note 1. ADC12i_ADI (i = 0, 1)
Note 2. ADC12i_GBADI (i = 0, 1)

Figure 45.29 Example of group priority operation 1-3: Group B trigger input during group A scan when rescanning is enabled (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0)

Operation example 1-4 shows the group priority operation in group-scan mode (when ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0).

Operation example 1-4: “Group A trigger input during group B scan” when rescanning is disabled

1. When input of a trigger for group B sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion for the analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order from the channel with the smallest number n.
2. On completion of A/D conversion for each channel in group B, the result is stored in the corresponding A/D Data Register y (ADDRy).
3. When a trigger for group A is input during A/D conversion for group B, A/D conversion for group B stops while the ADCSR.ADST bit remains 1, and then A/D conversion for the group A analog input channels selected in the ADANSA0 and ADANSA1 registers starts according to the conversion order from the channel with the smallest number n. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D Data Register y (ADDRy).
4. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
5. On completion of A/D conversion for group A, an ADC120_ADI interrupt request is generated.
6. When A/D conversion for group A completes, the ADCSR.ADST bit is automatically cleared and the A/D converter enters a wait state. A/D conversion for group B is not performed until a trigger for group B is input the next time.



Note: w, x, y, and z (x < y < z) are the selected channel numbers by ADANSA0/1 or ADANSB0/1.
Note 1. ADC12i_ADI (i = 0, 1)
Note 2. ADC12i_GBADI (i = 0, 1)

Figure 45.30 Group priority operation example 1-4: "Group A trigger is input during group B scan" when rescanning is disabled (when ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0)

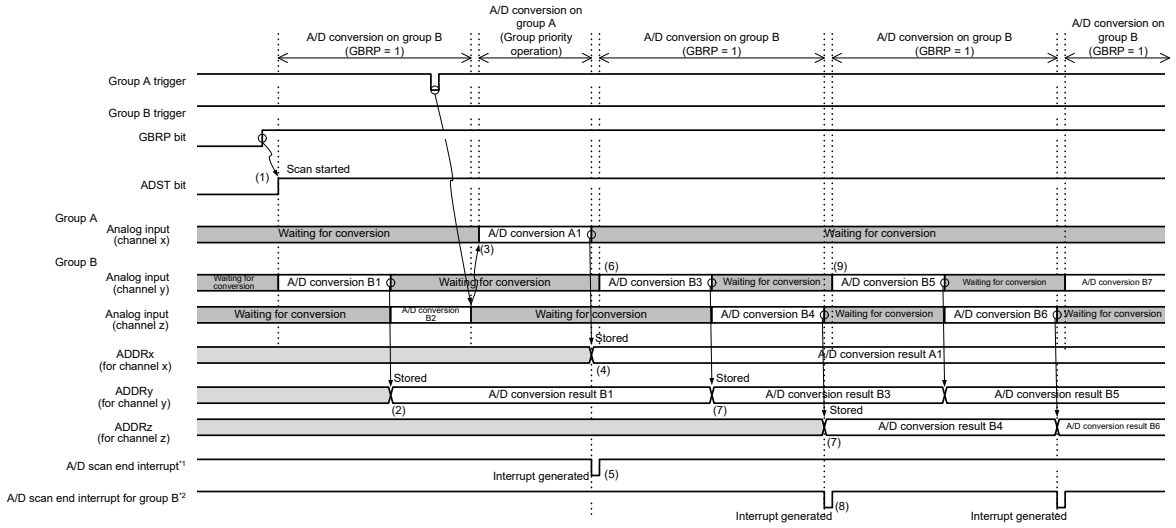
Operation example 1-5 shows the group priority operation in group-scan mode (when ADGSPCR.GBRP = 1, and ADGSPCR.LGRRS = 0).

Operation example 1-5: Continuously activating single-scan operation for group B

1. When ADGSPCR.GBRP = 1 is set, the ADCSR.ADST bit is set to 1 (starting A/D conversion) and A/D conversion for the analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order from the channel with the smallest number n.
2. On completion of A/D conversion for each channel in group B, the result is stored in the corresponding A/D Data Register y (ADDRy).
3. When a trigger for group A is input during A/D conversion for group B, A/D conversion for group B stops while the ADCSR.ADST bit remains 1, and then A/D conversion for group A analog input channels selected in the ADANSA0 and ADANSA1 registers starts according to the conversion order from the channel with the smallest number n. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D Data Register y (ADDRy).
4. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
5. On completion of A/D conversion for group A, an ADC120_ADI interrupt request is generated.
6. If ADGSPCR.GBRP = 1 is set (performing single scan continuously), A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order from the channel with the smallest number n while the ADCSR.ADST remains 1 (starting A/D conversion).
7. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).

8. If the setting of the ADCSR.GBADIE bit is 1 (enabling interrupt generation on completion of group B scan), a group B scan end interrupt request is generated.
9. If ADGSPCR.GBRP = 1 is set (performing single scan continuously), A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order from the channel with the smallest number n while the ADCSR.ADST remains 1 (starting A/D conversion).

Steps 6 to 9 are repeated as long as the ADGSPCR.GBRP bit remains 1. Do not clear the ADCSR.ADST bit as long as the ADGSPCR.GBRP bit is 1. To forcibly stop A/D conversion while ADGSPCR.GBRP = 1, follow the procedure shown in Figure 45.43.



Note: x, y, and z (y < z) are the selected channel numbers by ADANSA0/1 or ADANSB0/1.
Note 1. ADC12i_ADI (i = 0, 1)
Note 2. ADC12i_GBADI (i = 0, 1)

Figure 45.31 Group priority operation example 1-5: Continuously activating single scan for group B (when ADGSPCR.GBRP = 1, ADGSPCR.LGRRS = 0)

Note: To continuously activate single-scan operation for group B, disable group B trigger input.

45.3.5 Compare Function for Windows A and B

45.3.5.1 Compare Function Windows A and B

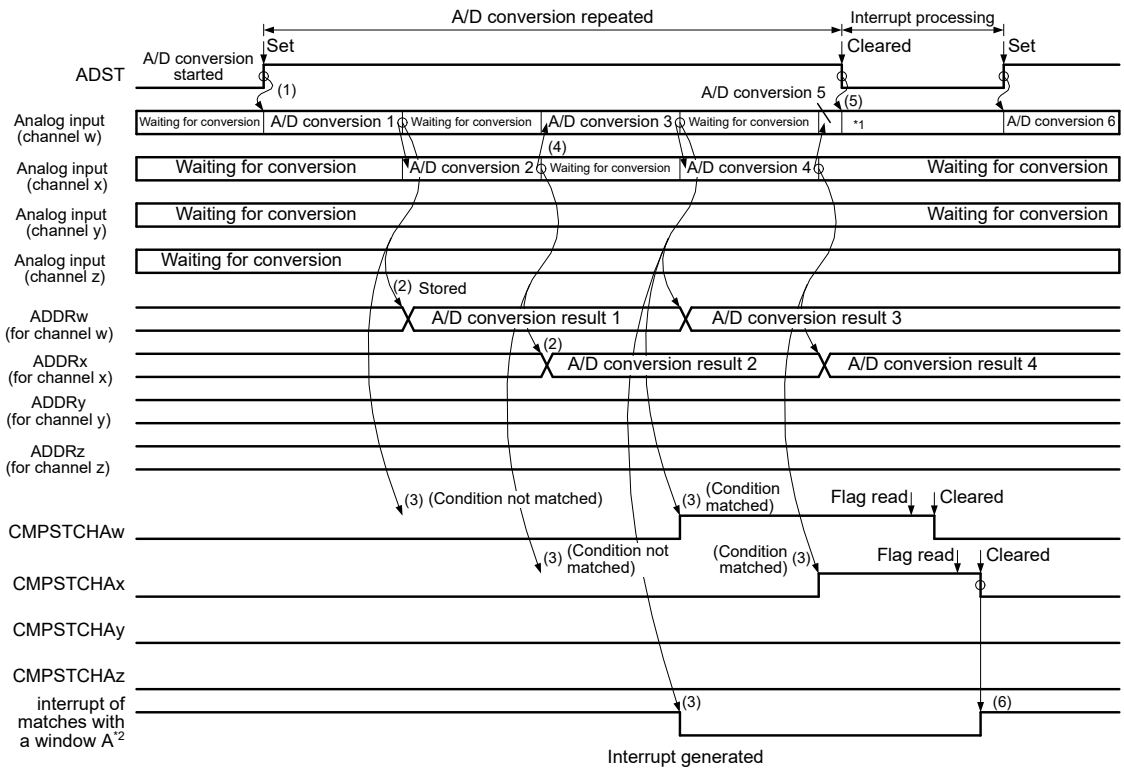
The compare function compares a reference value with the A/D conversion result. The reference value can be set for Window A and Window B independently. When the compare function is in use, the self-diagnosis function and double trigger mode cannot be used. The main differences between Window A and Window B are their different interrupt output signals and the constraint on Window B of only one selectable channel.

This section provides an example operation that combines continuous scan mode and the compare function.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, a synchronous trigger (GPT, ELC) or an asynchronous trigger, A/D conversion starts in the order of the selected channels , temperature sensor, and internal reference voltage.
2. On completion of A/D conversion, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy, ADTSDR, or ADOCDR). When ADCMPCR.CMPAE = 1, if bits in the ADCMPANSRy register or the ADCMPANSER register are set for Window A, the A/D conversion result is compared with the set ADCMPDR0/1 register value. When ADCMPCR.CMPBE = 1, if bits in the ADCMPBNSR register are set for Window B, the A/D conversion result is compared with the ADWINULB/ADWINLLB register setting.

3. As a result of the comparison, when Window A meets the condition set in ADCMPLR0/1 or ADCMPLER, the Compare Window A Flag (ADCMPSR0.CMPSTCHAn, ADCMPSR1.CMPSTCHAn, ADCMPSER.CMPSTTSA or ADCMPSER.CMPSTOCA) sets 1. At this time, if the ADCMPCR.CMPAIE bit is 1, an ADC12i_CMPAI (i = 0, 1) interrupt request is generated. In the same way, when Window B meets the condition set in ADCMPBNSR.CMPLB, the Compare Window B Flag (ADCMPBSR.CMPSTB) sets to 1. At this time, if the ADCMPCR.CMPBIE bit is 1, an ADC12i_CMPBI (i = 0, 1) interrupt request is generated.
4. On completion of all selected A/D conversions and comparisons, scan restarts.
5. After the ADC12i_CMPAI (i = 0, 1) and ADC12i_CMPBI (i = 0, 1) interrupts are accepted, the ADCSR.ADST bit is set to 0 (A/D conversion stop) and processing is performed for channels for which the compare flag is set to 1.
6. When all compare flags of Window A are cleared, the ADC12i_CMPAI (i = 0, 1) interrupt request is canceled. In the same way, when all compare flags of Window B are cleared, the ADC12i_CMPBI (i = 0, 1) interrupt request is reset. To perform comparison again, restart the A/D conversion.



Note: w, x, y, and z (w < x < y < z) are the selected channel numbers by ADANSA0/1.
Note 1. Data on conversion is ignored.
Note 2. ADC12i_CMPAI (i = 0, 1)

Figure 45.32 Example of compare function operation, when the analog inputs (channel w to z) are compared

45.3.5.2 Event Output of Compare Function

The event output of the compare function specifies the upper-side reference value and the lower-side reference value for window A and window B, respectively. The output compares the A/D converted value of the selected channel with the upper and lower side reference value and outputs events (ADC12i_WCMPM (i = 0, 1)/ADC12i_WCMPUM (i = 0, 1)) based on event conditions (A OR B, A AND B, A XOR B) and comparison result of window A and window B.

If more than one channel is selected for window A, and even when one channel in window A meets the comparison condition, the comparison result of window A is met. When using this function, perform A/D conversion in single scan mode.

Any channels from analog input, internal reference voltage, temperature sensor output and VBATT 1/3 voltage monitor output are selectable for window A.

One channel from analog input, internal reference voltage, temperature sensor output and VBATT 1/3 voltage monitor output is selectable for window B.

The following sequence is an example of how to set up and use the event output of the compare function:

1. Confirm that the value in the ADCSR.ADCS bits is 00b (single scan mode).
2. Select the channel for window A in the ADCMPANSR0/1 and ADCMPANSER registers. Set the window comparison conditions in the ADCMPLR0/1 and ADCMPLER registers. Set the upper-side and lower-side reference values in the ADCMPDR0/1 registers.
3. Select the channel and comparison conditions for Window B in the ADCMPBNSR register, and set the upper and lower reference values in the ADWINULB and ADWINLLB registers.
4. Set the composite conditions for window A/B, window A/B operation enable, and interrupt output enable in the ADCMPCR register.

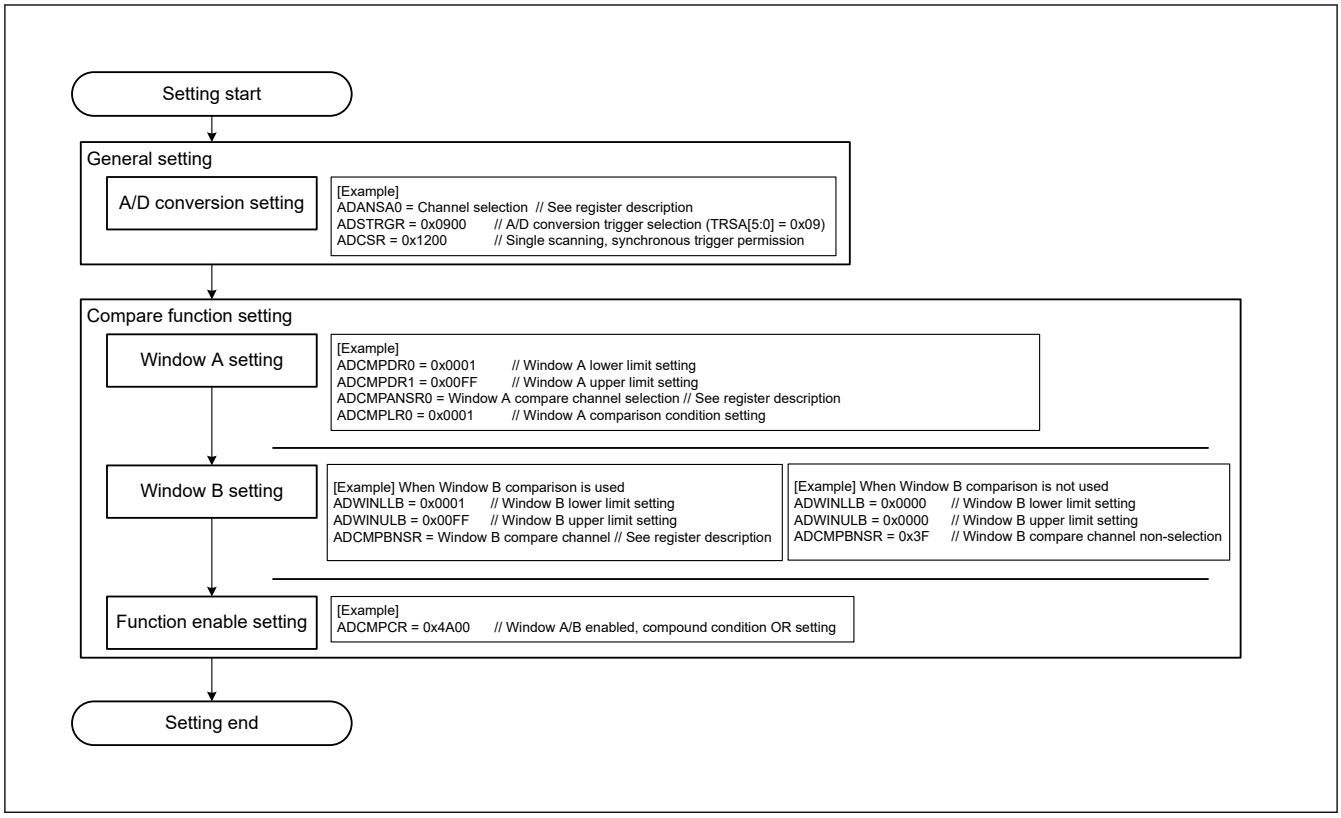


Figure 45.33 Setting example when using the event output of the compare function

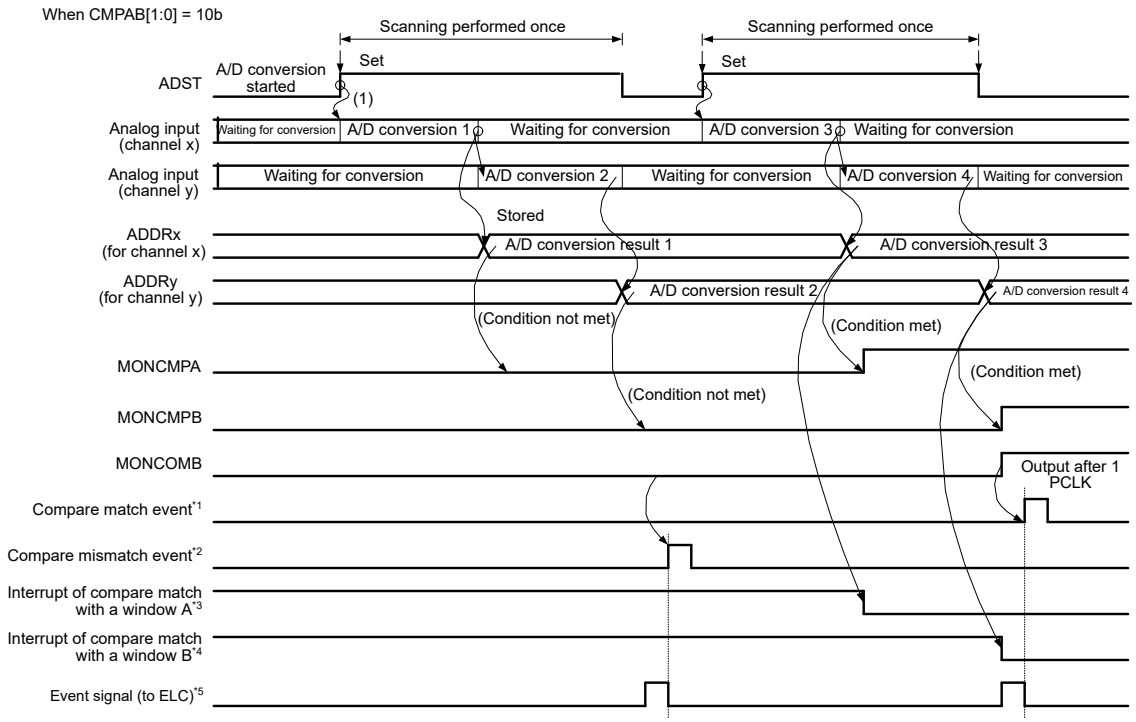
For event output usage when using only window A for the compare function, note the following:

- Enable both Window A and Window B (ADCMPCR.CMPAE = 1, ADCMPCR.CMPBE = 1)
- Set the compound condition of Window A and Window B to “OR condition” (ADCMPCR.CMPAB[1:0] = 00b)
- Set the compared channel of Window B to “No selection” (ADCMPBNSR.CMPCHB[5:0] = 0x3F)
- Set the compare condition of Window B to “0 < results < 0 always means mismatch”. (ADCMPCR.WCMPE = 1, ADWINLLB[15:0] = ADWINULB[15:0] = 0x0000, and ADCMPBNSR.CMPLB = 1)

Figure 45.34 shows the event output operation example of compare function.

A scan end event (ADC12i_ADI (i = 0, 1)) is output with the same timing as single scan completion. A match or mismatch event (ADC12i_WCMPM (i = 0, 1)/ADC12i_WCMPUM (i = 0, 1)) is output with 1 PCLKA cycle delay depending on the ADCMPCR.CMPAB[1:0] settings.

Note: The match and mismatch events are exclusive, so both events are never output simultaneously.



Note: x, and y (x < y) are the selected channel numbers by ADANSA0/1.
Note 1. ADC12i_WCMPM (i = 0, 1)
Note 2. ADC12i_WCMPUM (i = 0, 1)
Note 3. ADC12i_CMPAI (i = 0, 1)
Note 4. ADC12i_CMPBI (i = 0, 1)
Note 5. ADC12i_ADI (i = 0, 1)

Figure 45.34 Example operation of the compare function event output, when the analog inputs (channel x and y) are compared

Note: Event output of compare function outputs match/mismatch from the comparison results of Window A and Window B, based on the ADCMPCR.CMPAB[1:0] settings.

Note: The comparison result of Window A is the logical addition of the comparison results of the comparison target channels of Window A. The comparison results of Window A and Window B are updated by each A/D conversion, and are kept even when single scan ends. Set ADCMPCR.CMPAE and ADCMPCR.CMPBE to 0 to clear the comparison results to 0.

45.3.5.3 Restrictions on Compare Function

The following constraints apply for the compare function:

- The compare function cannot be used together with the self-diagnosis function or double-trigger mode. The compare function is not available for ADRD, ADDBLDR, ADDBLDRA, and ADDBLDRB.
- Specify single scan mode when using match/mismatch event outputs.
- When the temperature sensor output, internal reference voltage is selected for Window A, Window B operations are prohibited.
- When the temperature sensor output, internal reference voltage is selected for Window B, Window A operations are prohibited.
- Setting the same channel for Window A and Window B is prohibited.
- Specify single scan mode when using the buffer function. It is prohibited to use double trigger mode together.
- Set the reference voltage values so that the upper-side reference voltage value is equal to or larger than the lower-side reference voltage value.

45.3.6 Analog Input Sampling and Scan Conversion Time

Scan conversion can be activated either by a software trigger, a synchronous trigger (GPT, ELC), or an asynchronous trigger (ADTRGn (n = 0, 1)). After the start-of-scanning-delay time (t_D) has elapsed, processing by the channel-dedicated sample-and-hold circuits, processing for disconnection detection assistance, and processing of conversion for self-diagnosis all proceed, followed by processing for A/D conversion.

Figure 45.35 shows the scan conversion timing, in which scan conversion is activated by a software trigger or a synchronous trigger (GPT, ELC). Figure 45.36 shows the scan conversion timing, in which scan conversion is activated by an asynchronous trigger (ADTRGn (n = 0, 1)). The scan conversion time (t_{SCAN}) includes the start-of-scanning-delay time (t_D), channel-dedicated sample-and-hold circuit processing time (tSPLSH)*1, disconnection detection assistance processing time (t_{DIS})*2, self-diagnosis A/D conversion processing time (t_{DIAG} and t_{DSD})*3, A/D conversion processing time (t_{CONV} and t_{DSD}), channel-dedicated sample-and-hold circuit end time (tSHED)*4 and end-of-scanning-delay time (t_{ED}).

The A/D conversion processing time (t_{CONV}) consists of input sampling time (t_{SPL}) and time for conversion by successive approximation (t_{SAM}). The sampling time (t_{SPL}) is used to charge sample-and-hold circuits in the A/D converter. If there is not sufficient sampling time due to the high impedance of an analog input signal source, or if the A/ D conversion clock (ADCLK) is slow, sampling time can be adjusted using the ADSSTRn register.

The time for conversion by successive approximation (t_{SAM}) is the following:

- 13 ADCLK states with 12-bit accuracy selected.
- 11 ADCLK states with 10-bit accuracy selected.
- 9 ADCLK states with 8-bit accuracy selected.

Table 45.28 shows the time for conversion by successive approximation (t_{SAM}).

The scan conversion time (t_{SCAN}) in single scan mode for which the number of selected channels is n can be determined as follows:

$$t_{SCAN} = t_D + t_{SPLSH} + (t_{DIS} \times n) + t_{DIAG} + t_{ED} + (t_{CONV} \times n) + t_{ED}$$ *5

The scan conversion time for the first cycle in continuous scan mode is t_{SCAN} for single scan minus t_{ED} . The scan conversion time for the second and subsequent cycles in continuous scan mode is fixed in the following:

$$t_{SPLSH} + (t_{DIS} \times n) + t_{DIAG} + t_{DSD} + t_{SHED} + (t_{CONV} \times n) + t_{SHED}$$ *5

Note 1. When no channel-dedicated sample-and-hold circuits are used, tSPLSH = 0.
Note 2. When disconnection detection assistance is not selected, t_{DIS} = 0.
Note 3. When the self-diagnosis function is not used, t_{DIAG} = 0, t_{DSD} = 0.
Note 4. When no channel-dedicated sample-and-hold circuits are used, t_{SHED} = 0, assuming continuous scan mode is active. In single scan mode and group scan mode, t_{SHED} is included in the end-of-scanning-delay (t_{ED}).
Note 5. When input sampling times (t_{SPL}) of all selected channels are the same, this element equals t_{CONV} × n. If each channel has a different sampling time, this element equals that of t_{SPL} and t_{SAM} set to each selected channel.

Table 45.28 shows the times for conversion during scanning.

Table 45.28 Conversion times during scanning (in numbers of cycles of ADCLK and PCLKA)

Item			Symbol		Type/Conditions			Unit
					Synchronous trigger*4	Asynchronous trigger	Software trigger	
Scan start processing time*1 *2	A/D conversion on group A under group A priority control.	Group B is to be stopped (Group A is activated after group B is stopped by of an A/D conversion source from group A).	t_D		3 PCLKA + 6 ADCLK 5 PCLKA + 3 ADCLK*5	—	—	Cycles
		Group B is not to be stopped (activation by an A/D conversion source from group A).			2 PCLKA + 4 ADCLK	—	—	
	A/D conversion when self-diagnosis is enabled.	A/D conversion for self-diagnosis is to be started.			2 PCLKA + 6 ADCLK	4 PCLKA + 6 ADCLK	6 ADCLK	
	All other				2 PCLKA + 4 ADCLK	2 PCLKA + 4 ADCLK	4 ADCLK	
Channel-dedicated sample-and-hold processing time*1	Sampling time		t_{SPLSH}	t_{SH}	Without continuous sampling: setting in ADSHCR.SSTSH[7:0] (initial value = 0x18 × ADCLK) With continuous sampling: 0			
	Wait time between sampling and A/D Conversion			t_W	12 ADCLK			
Disconnection detection assistance processing time			t_{DIS}		Setting in ADNDIS[3:0] (initial value = 0x0) × ADCLK			
Self-diagnosis conversion processing time*1	Sampling time		t_{DIAG}	t_{SPL}	Setting in ADSSTR00 (initial value = 0x0B) × ADCLK*3	—	—	
	Time for conversion by successive approximation	12-bit conversion accuracy		t_{SAM}	15 ADCLK	—	—	
		10-bit conversion accuracy			13 ADCLK	—	—	
		8-bit conversion accuracy			11 ADCLK	—	—	
	Wait time between self-diagnosis conversion end and analog channel sampling start.			t_{DED}	2 ADCLK			
	Wait time between last channel conversion end and self-diagnosis sampling start in continuous scan mode.		t_{DSD}		2 ADCLK			
A/D conversion processing time*1	Sampling time		t_{CONV}	t_{SPL}	Setting in ADSSTRn (n = 0 to 8 (unit 0), 0 to 6 (unit 1), L, T, O, V) (initial value = 0x0B) × ADCLK + 0.5 ADCLK			
	Time for conversion by successive approximation	12-bit conversion accuracy		t_{SAM}	13 ADCLK			
		10-bit conversion accuracy			11 ADCLK			
		8-bit conversion accuracy			9 ADCLK			
Channel-dedicated sample-and-hold end processing time			t_{SHED}		2 ADCLK			
Scan end processing time*1			t_{ED}		1 PCLKA + 3 ADCLK 2 PCLKA + 3 ADCLK*5			

Note 1. See Figure 45.35 and Figure 45.36 for an illustration of times t_D, t_{SPLSH}, t_{DIAG}, t_{CONV}, and t_{ED}.
Note 2. This is the maximum time required from software writing or trigger input to A/D conversion start.
Note 3. The sampling time setting must satisfy the electrical characteristics.
Note 4. This does not include the time consumed in the path from timer output to trigger input.
Note 5. If ADCLK is faster than PCLKA (PCLKA to ADCLK frequency ratio = 1:2 or 1:4), the scan end processing time changes.

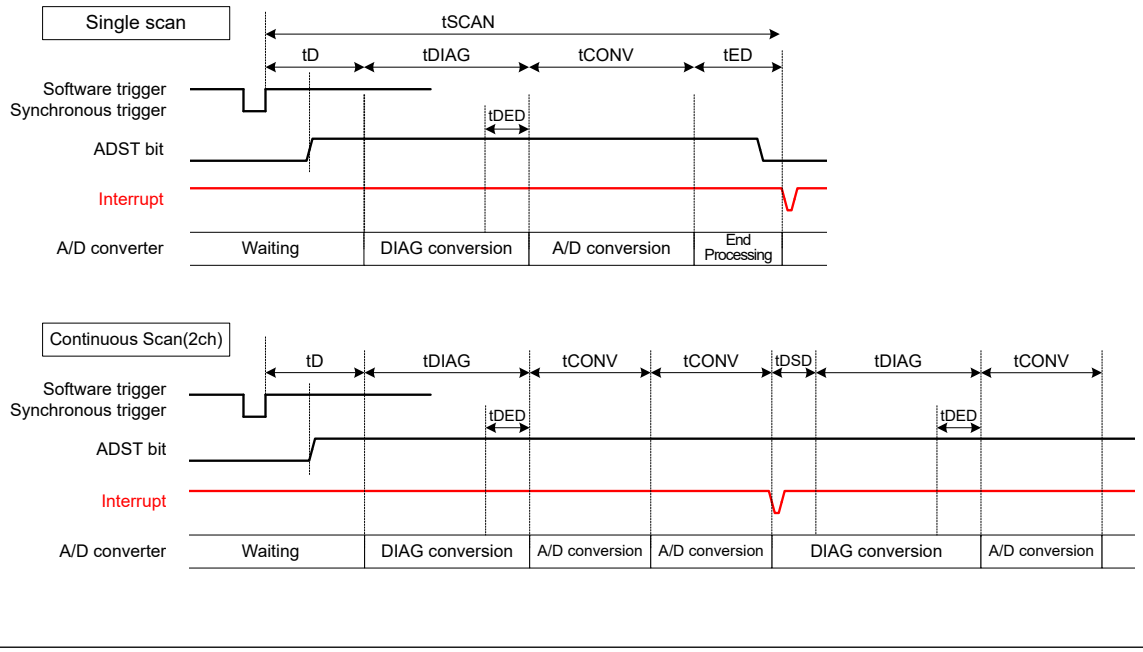


Figure 45.35 Scan conversion timing when activated by software or a synchronous trigger input (GPT, ELC)

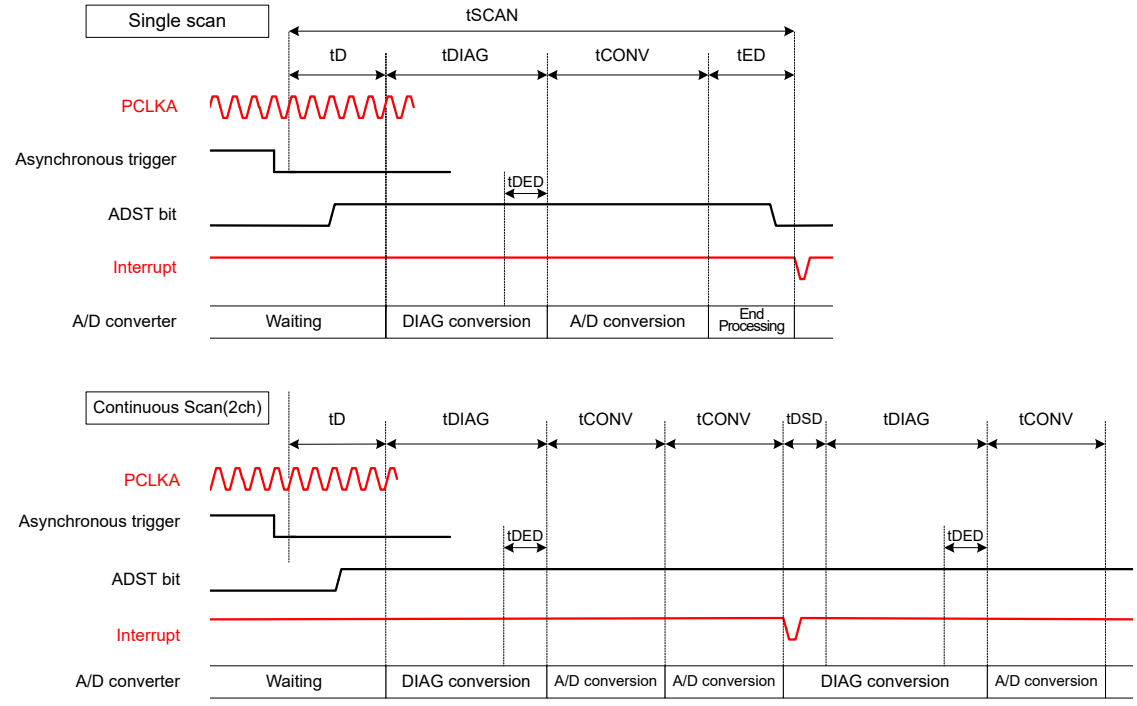


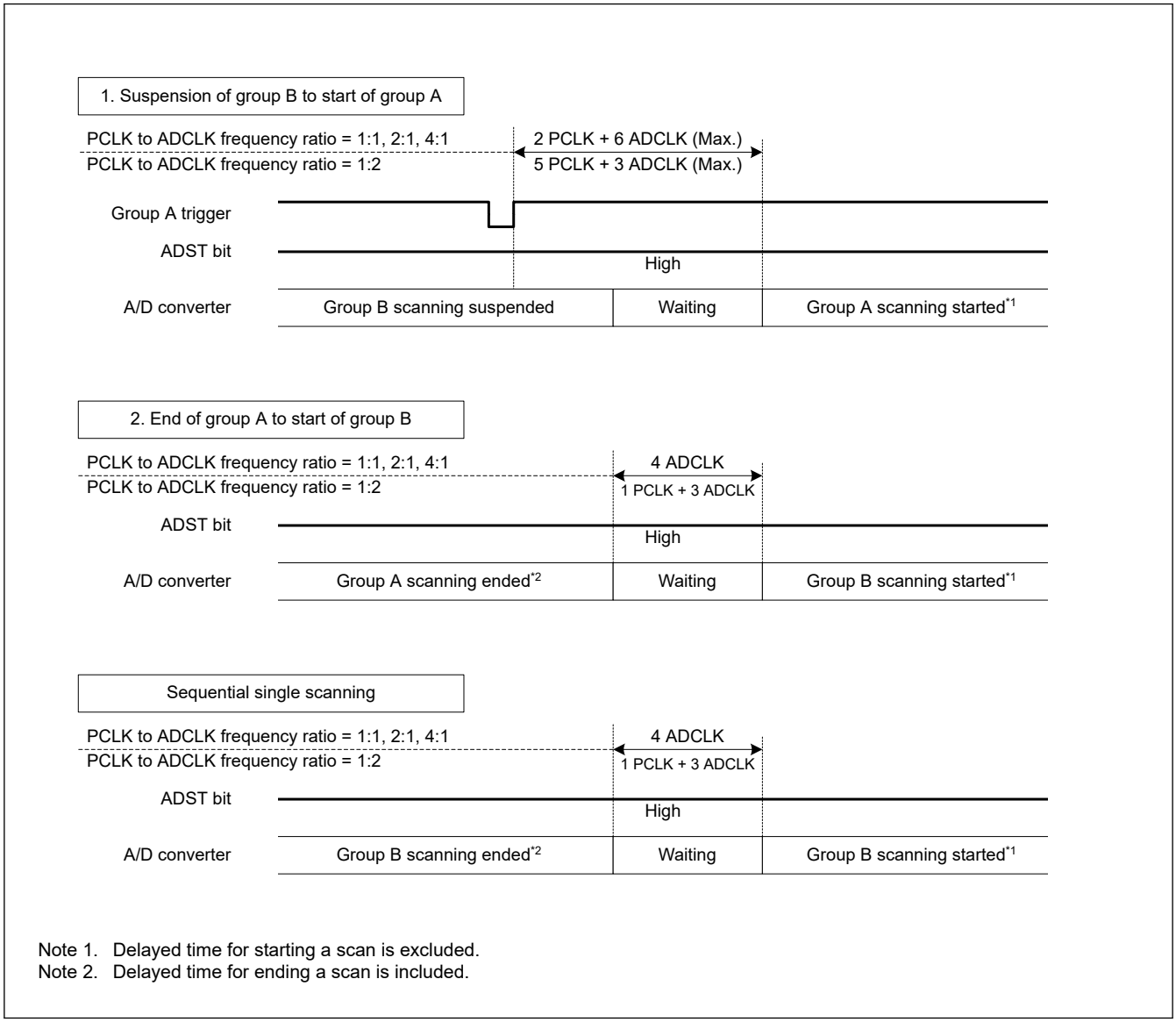
Figure 45.36 Scan conversion timing when activated by an asynchronous trigger input (ADTRGn (n = 0, 1))

45.3.6.1 Timing of Suspension and Starting of Scanning in Operation under Group Priority Control

The timings for suspension and starting of scanning in operation under group priority control that must considered are listed below.

1. The timing for suspending a scan of a group with a lower-priority and the timing for starting a scan of a group with a higher-priority.
2. The time at which scanning by the group with a lower-priority is resumed on completion of scanning by the higher-priority group when the trigger for scanning by the lower-priority group is accepted during scanning by the higher-priority group.
3. The timing for performing sequential single scans by a lower-priority group.

Figure 1.37 shows the timing diagram of each of the above cases.



Note 1. Delayed time for starting a scan is excluded.
Note 2. Delayed time for ending a scan is included.

Figure 45.37 Timing diagram of suspension and starting of scanning in operation under group priority control

45.3.7 Usage Example of A/D Data Register Automatic Clearing Function

Setting the ADCER.ACE bit to 1 automatically clears the A/D data registers (ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR, ADVMDR) to 0x0000 when the A/D data registers are read by the CPU or DTC or DMAC.

This function enables detection of update failures of the A/D data registers (ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR, ADVMDR). This section describes examples in which the function to automatically clear the ADDRy register is enabled and disabled.

- If the ADCER.ACE bit is 0 (automatic clearing is disabled) and for some reason, if the A/D conversion result (0x0222) is not written to the ADDRy register, the ADDRy value retains the old data (0x0111). In addition, if this ADDRy value is read into a general-purpose register using an A/D scan end interrupt, the old data (0x0111) can be saved in the general-purpose register. When checking whether there is an update failure, it is necessary to frequently save the old data in SRAM or in a general-purpose register.
- If the ADCER.ACE bit is 1 (automatic clearing is enabled), when ADDRy = 0x0111 is read by the CPU or DTC or DMAC, ADDRy is automatically set to 0x0000. Next, if the A/D conversion result of 0x0222 cannot be transferred to ADDRy for some reason, the cleared data (0x0000) remains as the ADDRy value. If this ADDRy value is read into a general-purpose register using an A/D scan end interrupt, 0x0000 is saved in the general-purpose register. Occurrence of an ADDRy update failure can be determined by checking that the read data value is 0x0000.

45.3.8 A/D-Converted Value Addition/Average Mode

A/D-converted value addition/average mode can be used when A/D conversion of the analog input of the selected channels, the temperature sensor output, the internal reference voltage and VBATT 1/3 voltage monitor output is selected.

In A/D-converted value addition mode, the same channel is A/D-converted 1, 2, 3, 4, or 16 consecutive times, and the sum of the converted values is stored in the data register. The conversion count of the addition function can be set to 16 only when 12-bit accuracy is selected. In A/D-converted value average mode, the same channel is A/D-converted 2 or 4 consecutive times, and the mean of the converted values is stored in the data register. The use of the average of these results can improve the accuracy of A/D conversion, depending on the types of noise components that are present. This function, however, cannot always guarantee an improvement in A/D conversion accuracy.

The A/D-converted value addition/average function can be used when A/D conversion of the analog inputs of the selected channels or A/D conversion of the temperature sensor output or A/D conversion of the internal reference voltage or A/D conversion of the VBATT 1/3 voltage monitori output is selected. The A/D-converted value addition/average function can also be used for channels for which the double-trigger function is selected.

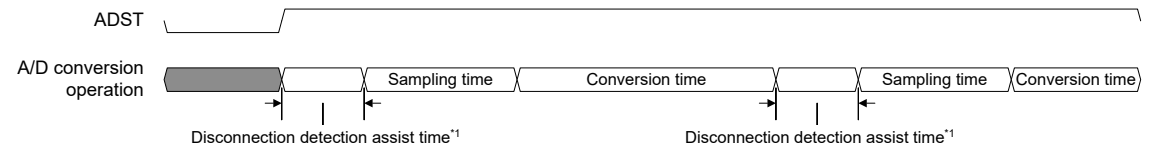
The addition function for self-diagnosis is not provided.

45.3.9 Disconnection Detection Assist Function

The ADC12 incorporates a function to fix the charge for sampling capacitance to the specified state (VREFH0 or VREFL0 for unit 0, VREFH or VREFL for unit 1) before the start of A/D conversion. This function enables disconnection detection in wiring of analog inputs.

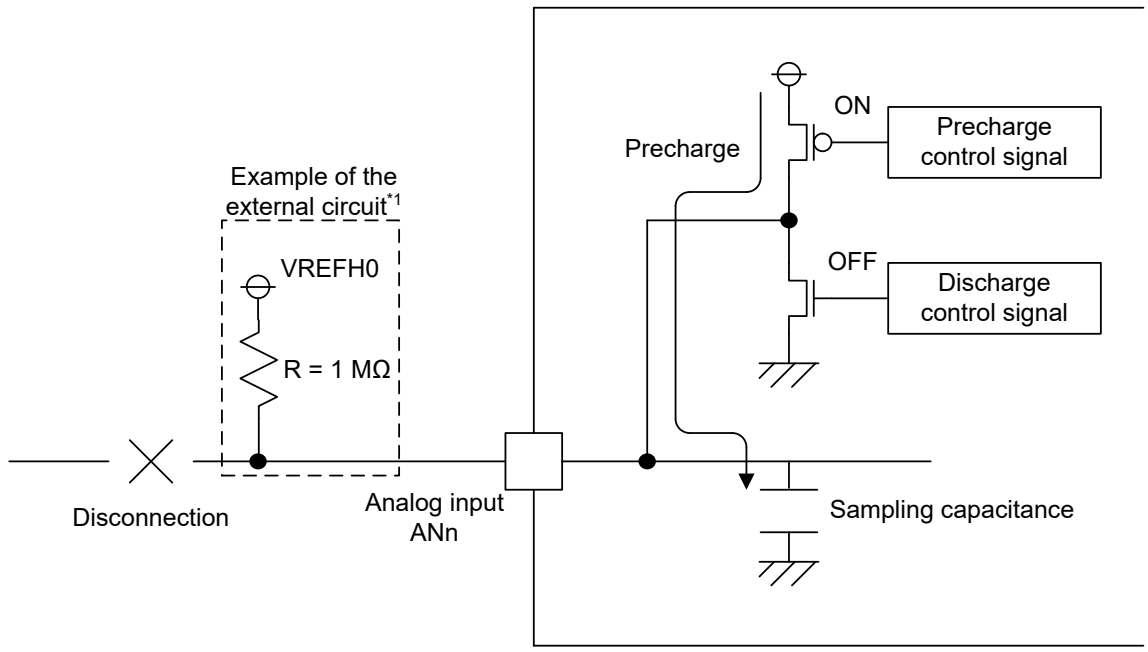
When using the disconnection detection assist function for the channel-dedicated sample-and-hold circuit, set the ADSHMSR.SHMD bit to 0 (select disable continuous sampling function).

Figure 45.38 shows the A/D conversion operation when the disconnection detection assist function is used. Figure 45.39 shows an example of disconnection detection when precharge is selected. Figure 45.40 shows an example of disconnection detection when discharge is selected.



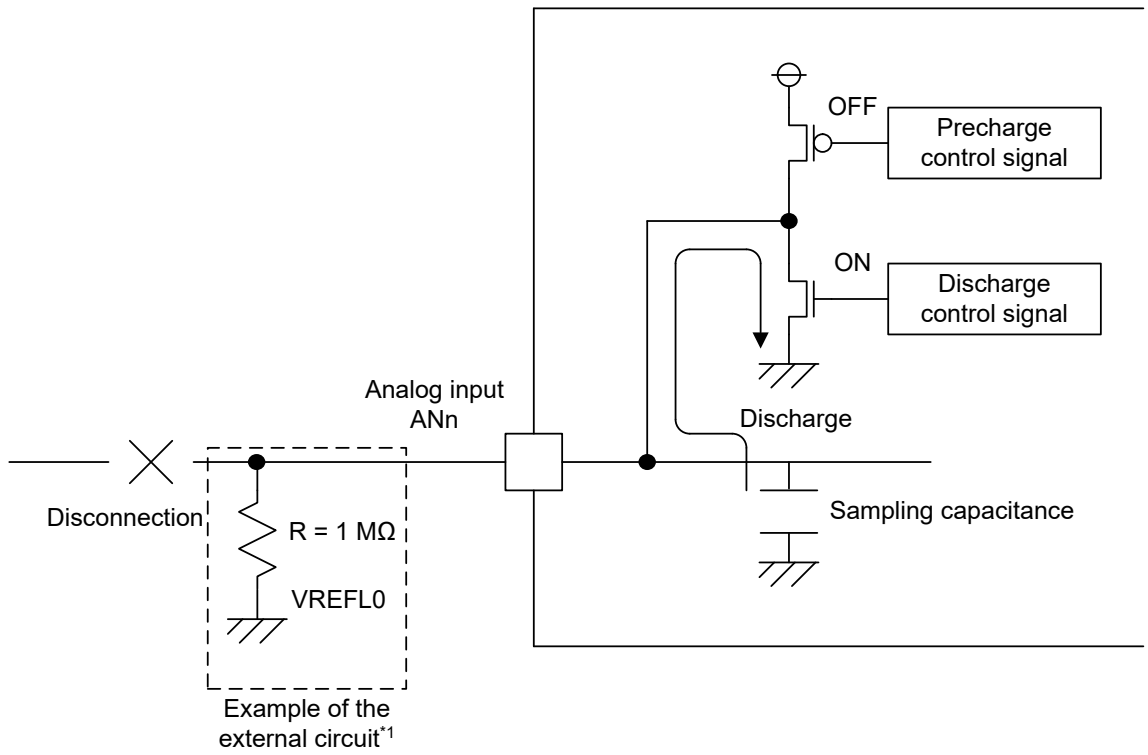
Note 1. 0 to 15 cycles of ADCLK

Figure 45.38 Operation of A/D conversion when disconnection detection assist function is used



Note 1. The converted result should be used after it is fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 45.39 Example of disconnection detection when precharge is selected



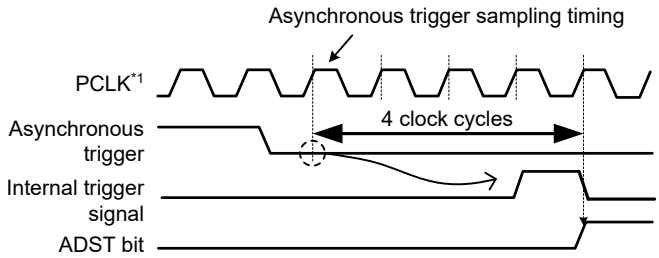
Note 1. The converted result should be used after it is fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 45.40 Example of disconnection detection when discharge is selected

45.3.10 Starting A/D Conversion with an Asynchronous Trigger

A/D conversion can be started by the input of an asynchronous trigger. To start A/D conversion by an asynchronous trigger, set the pin function in the PmnPFS register, set the A/D Conversion Start Trigger Select bits (ADSTRGR.TRSA[5:0]) to 0x00, then input a high-level signal to the asynchronous trigger (ADTRGn (n = 0, 1) pin). Finally, set both the ADCSR.TRGE and ADCSR.EXTRG bits to 1. Figure 45.41 shows timing of the asynchronous trigger input.

An asynchronous trigger cannot be selected in the A/D conversion start trigger for group B used in group scan mode. For details on setting the pin function, see section 19, I/O Ports.



Note 1. PCLKA

Figure 45.41 Asynchronous trigger input timing

45.3.11 Starting A/D Conversion with a Synchronous Trigger from a Peripheral Module

A/D conversion can be started by a synchronous trigger (GPT, ELC). To do this, set the ADCSR.TRGE bit to 1 and the ADCSR.EXTRG bit to 0, and select the relevant sources in the ADSTRGR.TRSA[5:0] bits and ADSTRGR.TRSB[5:0] bits.

45.3.12 Security Attribution between ADCn (n = 0, 1) and PORT (PmSAR (m = 0 to 9, A, B))

Only when the security attribution of ADC12n (n = 0, 1) match the security attribution of PORT (PmSAR), the correspond analog input channel can be set for ADC conversion. Otherwise the conversion result is unexpected.

Table 45.29 shows the condition of ADC for each Security Attribution setting.

Table 45.29 Analog input pin conditions by security settings

Security Attribution setting*1		ADC input channel select	
ADC12n (n = 0, 1) security setting PSARD	PORT security setting PmSAR	ADANSA0/ADANSA1 ADANSB0/ADANSB1	Setting Combination result
0 (Secure)	0 (Secure)	1 (channel correspond bit)	Input Valid
1 (Non-secure)	1 (Non-secure)	1 (channel correspond bit)	Input Valid
1 (Non-secure)	0 (Secure)	1 (channel correspond bit)	Input invalid and the conversion result is unexpected
0 (Secure)	1 (Non-secure)	1 (channel correspond bit)	Input invalid and the conversion result is unexpected

Note 1. VBATT 1/3 voltage monitor output do not have the Security Attribution judgement function.

45.3.13 Using Data Buffers

A ring buffer function consisting of 16 A/D data buffers is available. This function sequentially stores A/D conversion results other than self-diagnosis result (including addition/average results) in data buffers (ADBUFn, n = 0 to 15).

Each conversion result is stored at the timing when the A/D conversion result is stored in the data register, and most recent 16 conversion result data are retained.

The figure-below shows the schematic of data buffers, pointer, and overflow flag operations. When the BUFEN bit is set to 1, the A/D conversion result is transferred at each end of A/D conversion. The pointer indicates the number of data buffer to which the next transferred data is to be written. When data is written to up to buffer 15, the pointer is reset to 0000b and the overflow flag is set to 1. Subsequently transferred data overwrites the previously written data.

The overflow flag is reset to the initial value by writing 0x00 to the ADBUFPTR register.

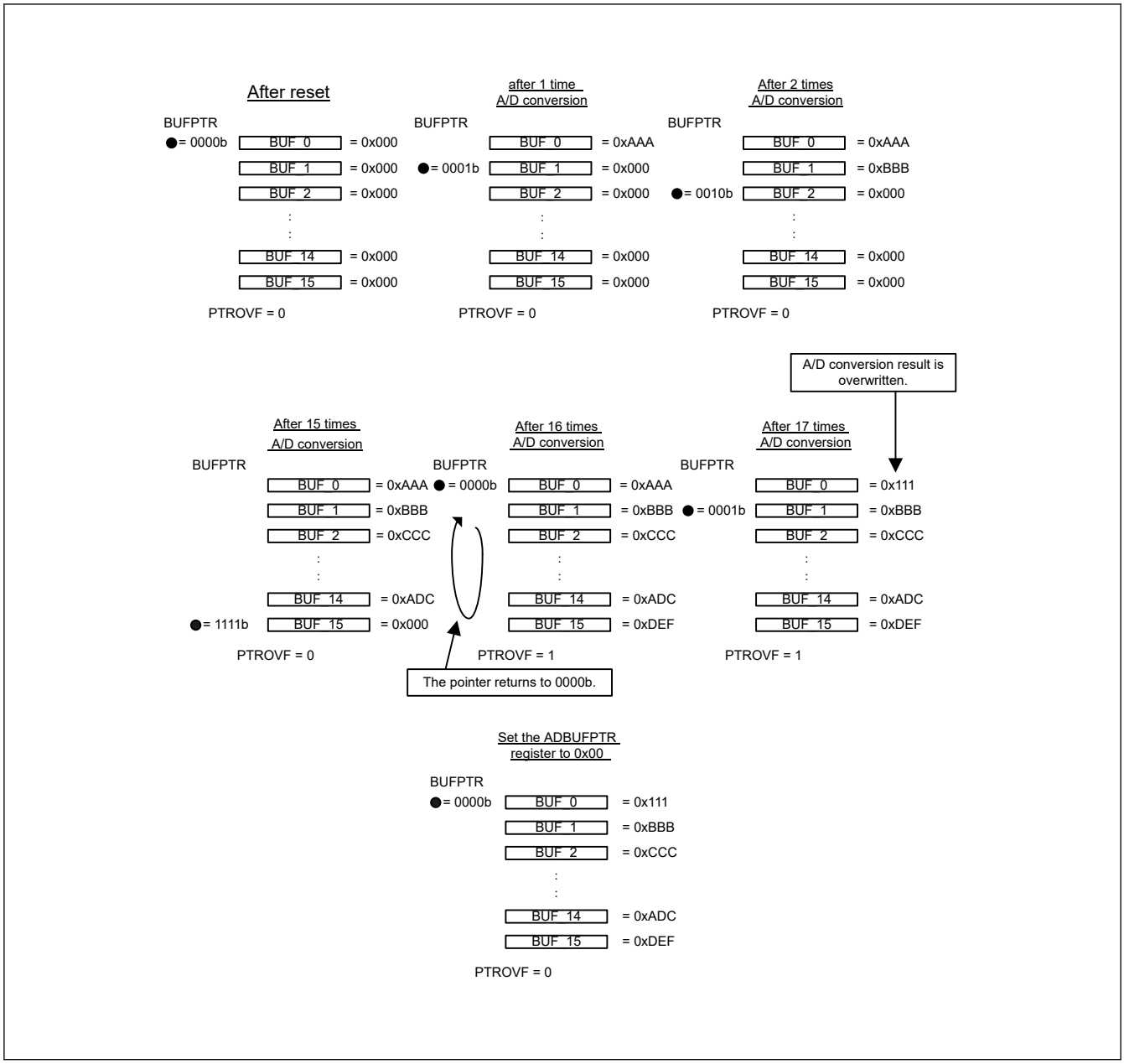


Figure 45.42 Data Buffers, Pointer, and Overflow Flag Operations

45.4 Interrupt Sources and DTC, DMAC Transfer Requests

45.4.1 Interrupt Requests

The ADC12 can send scan end interrupt requests ADC12i_ADI (i = 0, 1) and ADC12i_GBADI (i = 0, 1) to the CPU. The ADC12 also generates the ADC12i_CMPAI (i = 0, 1)/ADC12i_CMPBI (i = 0, 1) interrupt for the CPU in response to matches with a condition for comparison.

An ADC12i_ADI (i = 0, 1) interrupt is always generated. An ADC12i_GBADI (i = 0, 1) interrupt can be generated by setting the ADCSR.GBADIE bit to 1. Similarly, ADC12i_CMPAI (i = 0, 1) and ADC12i_CMPBI (i = 0, 1) interrupts can be generated by setting the ADCMPCR.CMPAIE and ADCMPCR.CMPBIE bit to 1.

In addition, the DTC or DMAC can be started when an ADC12i_ADI (i = 0, 1) or an ADC12i_GBADI (i = 0, 1) interrupt is generated. Using an ADC12i_ADI (i = 0, 1) or ADC12i_GBADI (i = 0, 1) interrupt to activate the DTC or DMAC to read the converted data enables continuous conversion without a burden on software.

Table 45.30 describes the interrupt sources and ELC events available for the ADC12.

Table 45.30 The interrupt source and ELC event of ADC12 (1 of 2)

Operation			Interrupt request or ELC event	Interrupt request	DTC or DMAC activation	ELC event request	Function
Scan mode	Double trigger mode	Compare function Window A/B					
Single scan mode	Deselected	Deselected	ADC12i_ADI (i = 0, 1)	✓	✓	✓	ADC12i_ADI (i = 0, 1) generated at the end of single scan
		Selected	ADC12i_ADI (i = 0, 1)	✓	✓	✓	ADC12i_ADI (i = 0, 1) generated at the end of single scan
			ADC12i_CMPAI (i = 0, 1)	✓	—	—	ADC12i_CMPAI (i = 0, 1) generated on a match comparison condition of Window A
			ADC12i_CMPBI (i = 0, 1)	✓	—	—	ADC12i_CMPBI (i = 0, 1) generated on a match comparison condition of Window B
			ADC12i_WCMPM (i = 0, 1)	—	✓	✓	ADC12i_WCMPM (i = 0, 1) generated on a match condition of the Window A/B compare function
			ADC12i_WCMPUM (i = 0, 1)	—	✓	✓	ADC12i_WCMPUM (i = 0, 1) generated on a mismatch condition of the Window A/B compare function
	Selected	Deselected	ADC12i_ADI (i = 0, 1)	✓	✓	✓	ADC12i_ADI (i = 0, 1) generated at the end of scans in the even numbered times
Continuous scan mode	Deselected	Deselected	ADC12i_ADI (i = 0, 1)	✓	✓	✓	ADC12i_ADI (i = 0, 1) generated at the end of scan of all selected channels
		Selected	ADC12i_CMPAI (i = 0, 1)	✓	—	—	ADC12i_CMPAI (i = 0, 1) generated on a match comparison condition of Window A
			ADC12i_CMPBI (i = 0, 1)	✓	—	—	ADC12i_CMPBI (i = 0, 1) generated on a match comparison condition of Window B

Table 45.30 The interrupt source and ELC event of ADC12 (2 of 2)

Operation			Interrupt request or ELC event	Interrupt request	DTC or DMAC activation	ELC event request	Function
Scan mode	Double trigger mode	Compare function Window A/B					
Group scan mode	Deselected	Deselected	ADC12i_ADI (i = 0, 1)	✓	✓	✓	ADC12i_ADI (i = 0, 1) generated at the end of group A scan
			ADC12i_GBADI (i = 0, 1)	✓	✓	—	ADC12i_GBADI (i = 0, 1) dedicated to group B generated at the end of group B scan
		Selected	ADC12i_ADI (i = 0, 1)	✓	✓	✓	ADC12i_ADI (i = 0, 1) generated at the end of group A scan
			ADC12i_GBADI (i = 0, 1)	✓	✓	—	ADC12i_GBADI (i = 0, 1) dedicated to group B generated at the end of group B scan
			ADC12i_CMPAI (i = 0, 1)	✓	—	—	ADC12i_CMPAI (i = 0, 1) generated on a match comparison condition of Window A
			ADC12i_CMPBI (i = 0, 1)	✓	—	—	ADC12i_CMPBI (i = 0, 1) generated on a match comparison condition of Window B
	Selected	Deselected	ADC12i_ADI (i = 0, 1)	✓	✓	✓	ADC12i_ADI (i = 0, 1) generated at the end of group A scans in the even-numbered times
			ADC12i_GBADI (i = 0, 1)	✓	✓	—	ADC12i_GBADI (i = 0, 1) dedicated to group B generated at the end of group B scan

Note: ✓ available
—: unavailable

For details on DTC settings, see section 17, Data Transfer Controller (DTC).

45.5 Event Link Function

45.5.1 Event Output to the ELC

The ELC uses the ADC12i_ADI (i = 0, 1) interrupt request signal as an event signal ADC12i_ADI (i = 0, 1), enabling link operation for the preset module. The ADC12i_GBADI (i = 0, 1) interrupt and ADC12i_CMPAI (i = 0, 1)/ADC12i_CMPBI (i = 0, 1) interrupts cannot be used as an event signal. For details, see Table 45.30.

An event signal can be output regardless of the settings of the corresponding interrupt request enable bits. For the scan end event (ADC12i_ADI (i = 0, 1)), a high-level pulse for one PCLKA cycle is output at the same output timing as the interrupt output (ADC12i_ADI (i = 0, 1)) shown in Table 45.30. For a compare function match (ADC12i_WCMPM (i = 0, 1)) and mismatch event (ADC12i_WCMPUM (i = 0, 1)) to the ELC, a high-level pulse for one PCLKA cycle is output at the timing delayed by one cycle (PCLKA) from the interrupt output (ADC12i_ADI (i = 0, 1)) shown in Table 45.30.

To use compare function match (ADC12i_WCMPM (i = 0, 1)) or mismatch event (ADC12i_WCMPUM (i = 0, 1)) to the ELC, specify single-scan mode.

45.5.2 ADC12 Operation through an Event from the ELC

The ADC12 can start A/D conversion by the preset event specified in the ELSRn settings for the ELC as follows:

- Select the ELC_AD00 (unit 0) signal in the ELC.ELSR8 register
- Select the ELC_AD01 (unit 0) signal in the ELC.ELSR9 register

- Select the ELC_AD10 (unit 1) signal in the ELC.ELSR10 register
- Select the ELC_AD11 (unit 1) signal in the ELC.ELSR11 register.

If an ELC event occurs during A/D conversion, the event is disabled.

45.6 Usage Notes

45.6.1 Constraints on Setting the Registers

Set each register while the ADCSR.ADST bit is 0.

45.6.2 Constraints on Reading the Data Registers

The following registers must be read in halfword units:

- A/D Data Registers
- A/D Data Duplexing Register
- A/D Data Duplexing Register A
- A/D Data Duplexing Register B
- A/D Temperature Sensor Data Register
- A/D Internal Reference Voltage Register
- A/D VBATT monitor Data Register
- A/D Self-Diagnosis Data Register
- A/D Data Buffer Registers n (N = 0 to 15)

If a register is read twice in byte units, that is, the upper byte and lower byte are read separately, the A/D-converted value read initially might disagree with the A/D-converted value read subsequently. To prevent this, never read the data registers in byte units.

45.6.3 Constraints on Stopping A/D Conversion

(1) A/D Conversion Stop Procedure

To stop A/D conversion when an asynchronous trigger or a synchronous trigger is selected as the condition for starting A/D conversion, follow the procedure shown in Figure 45.43.

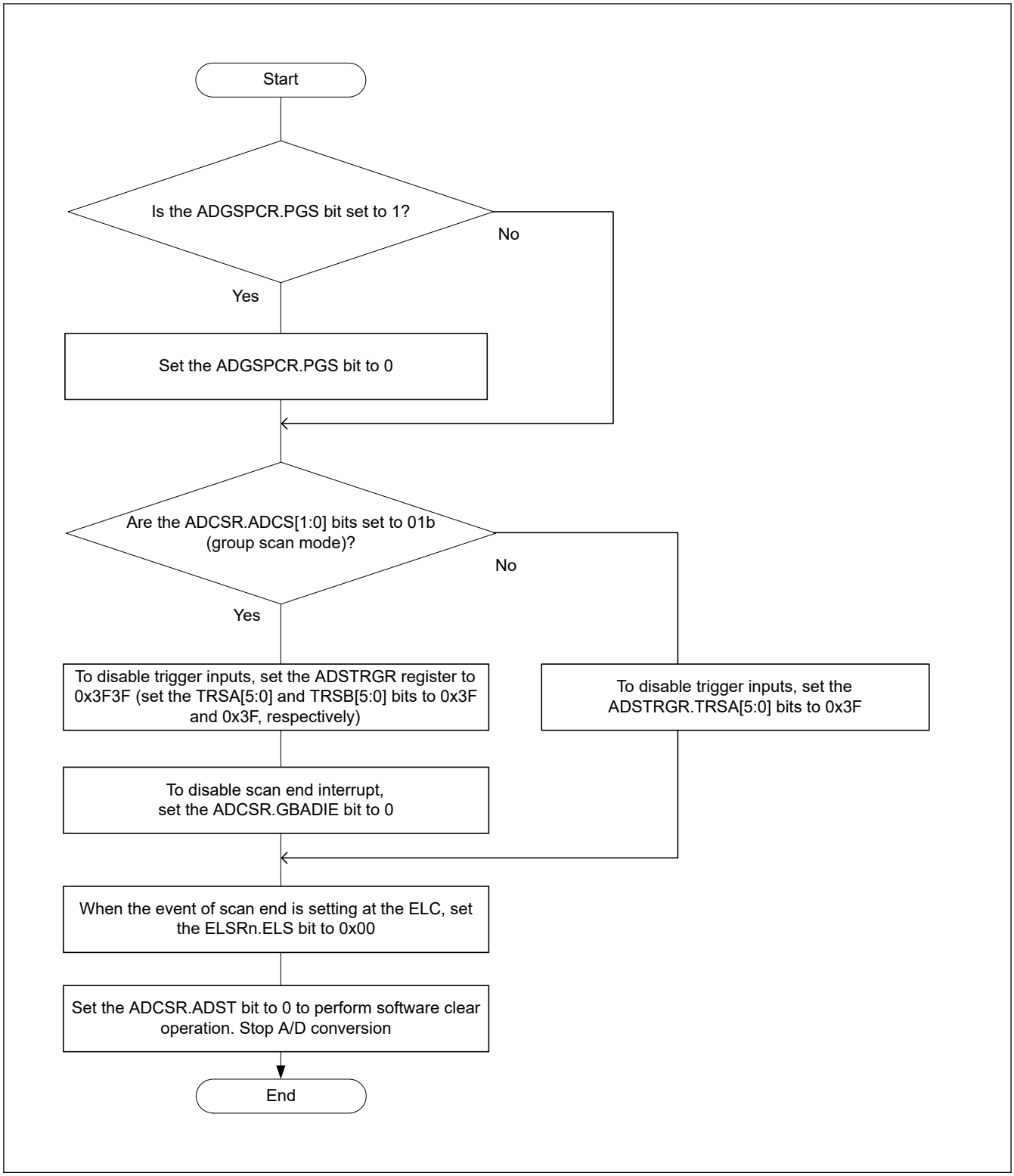


Figure 45.43 Procedures for clearing the ADCSR.ADST bit by software

To specify the following settings after performing the clear operation by software, provide a wait period for at least two ADCLK cycles.

- Enabling scan end interrupts
- Enabling scan end events for the event link controller
- Starting A/D conversion by software
- Enabling trigger input

(2) Notes on Modes and Status Bits

If necessary, individually initialize or set again the voltage status for self-diagnosis, the judgment of the even number or odd number specified for double-trigger mode, and the monitor flags of the compare function.

- To set again the voltage status for self-diagnosis, set the ADCER.DIAGLD bit to 1 and then set a desired value in the ADCER.DIAGVAL[1:0] bits.
- If the setting of the ADCSR.DBLE bit is changed from 0 to 1, the double-trigger mode operation starts from the first scanning.
- To initialize the monitor flags of the compare function (MONCMPA, MONCMPB, and MONCOMB), set the ADCMPCR.CMPAE and ADCMPCR.CMPBE bits to 0.
- To initialize the data storage buffer pointer, set the ADBUFPTR register to 0x00.
- To initialize the continuous sampling function (ADSHMSR.SHMD = 1), set ADSHMSR.SHMD to 0. To use the continuous sampling function again after initialization, set ADSHMSR.SHMD to 1 after waiting for 1 cycle of ADCLK or longer.

45.6.4 A/D Conversion Restart and Termination Timing

A maximum of 6 ADCLK cycles is required for the idle analog unit of the ADC12 to restart on setting the ADCSR.ADST bit to 1. A maximum of 2 ADCLK cycles is required for the operating analog unit of the ADC12 to terminate on setting the ADCSR.ADST bit to 0. A maximum of 1 PCLKA + 1 ADCLK cycles is required for the operating analog unit of the ADC12 to terminate on setting the ADCSR.ADST bit to 0 when PCLKA to ADCLK frequency ratio is 1:2 or 1:4.

45.6.5 Constraints on Scan End Interrupt Handling

When scanning the same analog input twice using any trigger, the first A/D-converted data is overwritten with the second A/D-converted data. This occurs when the CPU does not complete the reading of the A/D-converted data by the time the A/D conversion of the first analog input for the second scan ends after the first scan end interrupt is generated.

45.6.6 Settings for the Module-Stop Function

The Module Stop Control Register can enable or disable ADC12 operation. The ADC12 is initially stopped after a reset. The registers become accessible on release from the module-stop state. After release from the module-stop state, wait for at least 1 µs before starting A/D conversion. For details, see section 10, Low Power Modes.

45.6.7 Notes on Entering the Low-Power States

Before entering the module-stop state or Software Standby mode, be sure to stop A/D conversion. Set the ADCSR.ADST bit to 0 and secure certain period until the analog unit of the ADC12 stops. Follow the procedure shown in Figure 45.43 to clear the ADCSR.ADST bit with software. Then, wait for 2 clock cycles of ADCLK before entering the module-stop state or Software Standby mode.

45.6.8 Notes on Canceling Software Standby Mode

After software standby mode is canceled, wait at least 1 µs after the stabilization time for the oscillator elapses and before starting A/D conversion. For details, see section 10, Low Power Modes

45.6.9 Error in Absolute Accuracy When Disconnection Detection Assistance Is in Use

Using disconnection detection assistance leads to an error in absolute accuracy of the ADC12. This error arises because an erroneous voltage is input to the analog input pins due to the resistive voltage division between the pull-up or pull-down resistor (Rp) and the resistance of the signal source (Rs). This error in absolute accuracy is calculated from the following formula:

$$\text{Maximum error in absolute accuracy (LSB)} = \left(2^{\text{Resolution}} - 1\right) \times \text{Rs} / \left(\text{Rs} + \text{Rp}\right)$$

Only use disconnection detection assistance after thorough evaluation.

45.6.10 Constraints on Operating Modes and Status Bits

Initialize or set again individually, if necessary, the voltage values in self-diagnosis, the value of the first scan or second scan in double trigger mode, the data buffer pointer, and status monitor in the compare function.

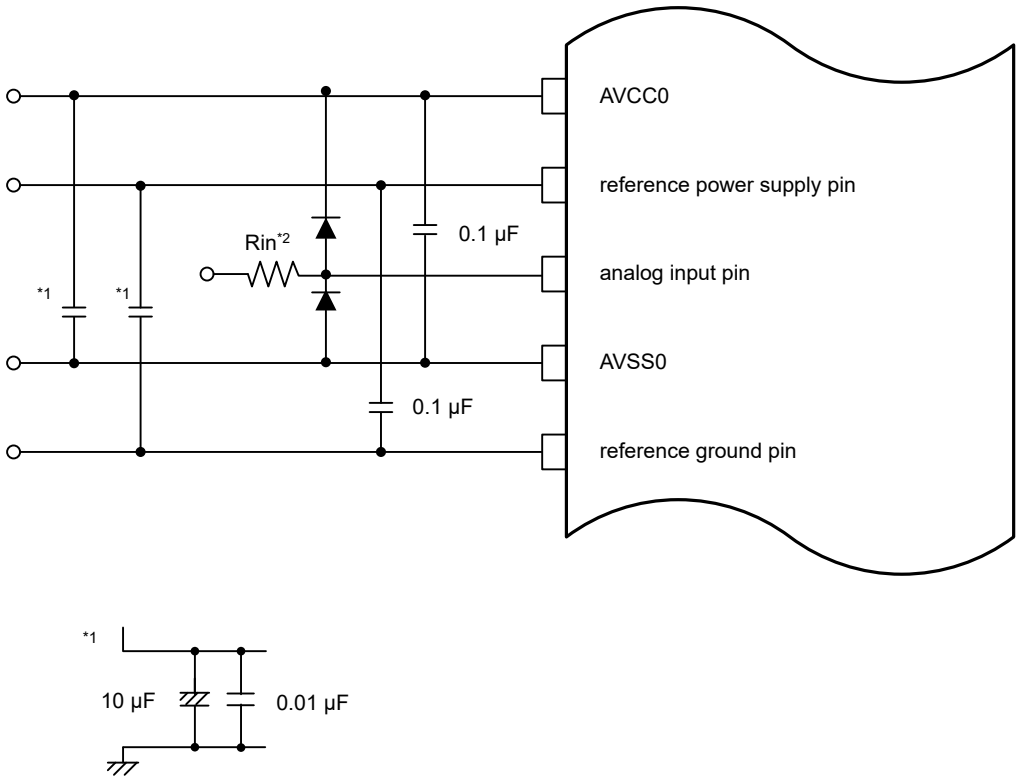
- Select the voltage values in self-diagnosis (ADCER.DIAGVAL[1:0]) after setting ADCER.DIAGLD to 1.
- Double-trigger mode operates as the first scan after setting ADCSR.DBLE from 0 to 1.
- The status monitor bits (MONCMPA, MONCMPB, MONCOMB) in the compare function are initialized after setting ADCMPCR.CMPAE and ADCMPCR.CMPBE to 0.

45.6.11 Notes on Board Design

The board should be designed so that digital circuits and analog circuits are separated from each other as far as possible. In addition, digital circuit signal lines and analog circuit signal lines should not intersect or be placed near each other. If these rules are not followed, noise can occur on analog signals and A/D conversion accuracy is affected. The analog input pins, reference power supply pin (VREFH0, VREFH), reference ground pin (VREFL0, VREFL), and analog power supply (AVCC0) should be separated from digital circuits using the analog ground (AVSS0). The analog ground (AVSS0) should be connected to a stable digital ground (VSS) on the board (single-point ground plane connection).

45.6.12 Constraints on Noise Prevention

To prevent the analog input pins from being destroyed by abnormal voltage such as excessive surge, insert a capacitor between AVCC0 and AVSS0 and between VREFH0 and VREFL0 and between VREFH and VREFL. Additionally, connect a protection circuit to protect the analog input pins as shown in Figure 45.44.



Note 1. The values shown here are reference values.
Note 2. Rin: Signal source impedance

Figure 45.44 Example protection circuit for analog inputs

45.6.13 Port Settings When Using the ADC12 Input

When using the high-precision channels, do not use PORT0 as digital output ports. Renesas recommends that you do not use the digital output that is also used as the AD analog input if normal-precision channel is used. If the digital output that is also used as the AD analog input is used for output signals, perform A/D conversion several times, eliminate the maximum and minimum values, and obtain the average of the other results.

45.6.14 Relationship between ADC12 Units 0 and 1 and the ACMPHS

For the A/D conversion targets in Table 45.31, unit 0 and 1 cannot perform A/D conversion at the same time.

Table 45.31 A/D conversion targets that are mutually exclusive with each other

A/D conversion target	
Unit 0	**Unit 1**
Temperature sensor	
Internal reference voltage	
VBATT 1/3 voltage monitor output	
AN016	AN116
AN017	AN117
AN018	AN118
AN019	AN119

The A/D conversion targets in Table 45.32 should not be selected as ACMPHS input during A/D conversion, because these pins are multiplexed with the ADC12 and ACMPHS.

Table 45.32 A/D conversion targets that are mutually exclusive with ACMPHS

A/D conversion target		
Unit 0	**Unit 1**	**ACMPHS**
AN000	-	ACMPHS0.IVCMP2
AN002	-	ACMPHS0.IVCMP3
AN005	-	ACMPHS0.IVCMP0
AN007/DA0	-	ACMPHS0.IVREF3
AN016	AN116	ACMPHS1.IVCMP0
Internal reference voltage	-	ACMPHS0.IVREF2
-	AN100	ACMPHS1.IVCMP2
-	AN101	ACMPHS1.IVREF0
-	AN102	ACMPHS1.IVCMP3
-	AN104	ACMPHS1.IVREF1
-	AN105/DA1	ACMPHS1.IVCMP1
-	Internal reference voltage	ACMPHS1.IVREF2

45.6.15 Calculation for Sampling Time

The sampling time can be easily estimated by the following figure and formula. This is the time to reach the voltage within 1/4 LSB.

$$t_{SPL} = (R_{EXT} + R_{AD}) \times (C_{EXT} + C_{AD}) \times \ln (C_{AD} / (C_{EXT} + C_{AD}) \times 2^{N+2})$$

R_{EXT} shows external signal source impedance
C_{EXT} shows external capacitance (pin capacitance*1 + PCB parasitic capacitance)
N = 12, 10 or 8 (conversion resolution)
C_{AD} = 5 pF (internal capacitance)
R_{AD} = 1.0 kΩ (internal resistance, case of high-speed channels)
R_{AD} = 2.0 kΩ (internal resistance, case of normal-speed channels)

Note 1. Typical value of analog input pin is 5 pF

For example, if R_{EXT} is 1 kΩ, C_{EXT} is 10 pF and N is 12 bits, t_{SPL} of high-speed channel is 258 ns.

This formula simplifies the general use case. This formula is not guaranteed and should be used only for estimation.

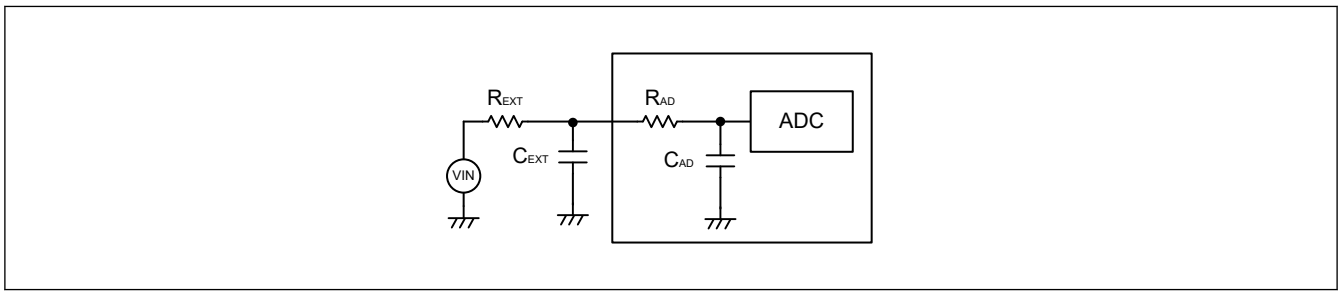


Figure 45.45 Sample and hold circuit simplified diagram

45.6.16 Notes on simultaneous operation with other modules

The following combinations may degrade the ADC accuracy depending on the specific timing of operation.

1. AN000 without channel-dedicated sample-and-hold and AN104 mutually.
2. AN007 and AN105 mutually.
3. Normal-precision channel of unit 0 and 1 mutually when these pins are adjacent or one pin apart.
4. AN000 to AN002 using channel-dedicated sample-and-hold when ADC unit 1, DAC12 or ACMPHS is operated.
5. High-precision channels of ADC unit1 when AN000 to AN002 using channel-dedicated sample-and-hold are operated.
6. AN019 when DAC12 or ACMPHS is operated.
7. AN119 to AN122 when DAC12 is operated in DCDC mode.
8. All channels of ADC unit 1 when ACMPHS is operated.
9. All channels of ADC unit 1 when DAC12 is operated.

To mitigate deterioration, it is recommended to take following measures.

- Performing A/D conversion with average mode.
- For even better results, performing A/D conversion several times, eliminate the maximum and minimum values, and obtain the average of the other results. (For the condition No.4, only performing average mode may not be effective depending on the noise. Eliminating the maximum and minimum is recommended.)
- For the condition No.9, using DA AD synchronous conversion set by DAADSCR.DAADST and DAADUSR.AMADSEL1 bits of DAC12.

Depending on the environment, ADC accuracy may be deteriorated even under conditions not listed here.

46. 12-Bit D/A Converter (DAC12)

46.1 Overview

The MCU provides a 12-bit D/A Converter (DAC12) with an output amplifier. Table 46.1 lists the DAC12 specifications, Figure 46.1 shows a block diagram, and Table 46.2 lists the I/O pins.

Table 46.1 DAC12 specifications

Parameter	Specifications
Resolution	12 bits
Output channels	2 channels
Interference reduction between analog modules	Methods provided to minimize interference between D/A and A/D conversion: • D/A converted data update timing is controlled by the synchronous D/A conversion enable input signal from the ADC12 (unit 1) • Degradation of A/D conversion accuracy caused by interference is reduced by controlling the DAC12 inrush current generation timing with the enable signal.
Module-stop function	Module-stop state can be set to reduce power consumption
Event link function (input)	The DA0 and DA1 conversion can be started on input of an event signal
D/A output amplifier control function	Controls whether the output amplifier (for both amplifier-through and amplifier-bias controls) is used
Destination of D/A output control function	Controls whether the output to the external pin or to the internal modules (ACMPHS) is used
TrustZone Filter	Security and Privilege attribution can be set

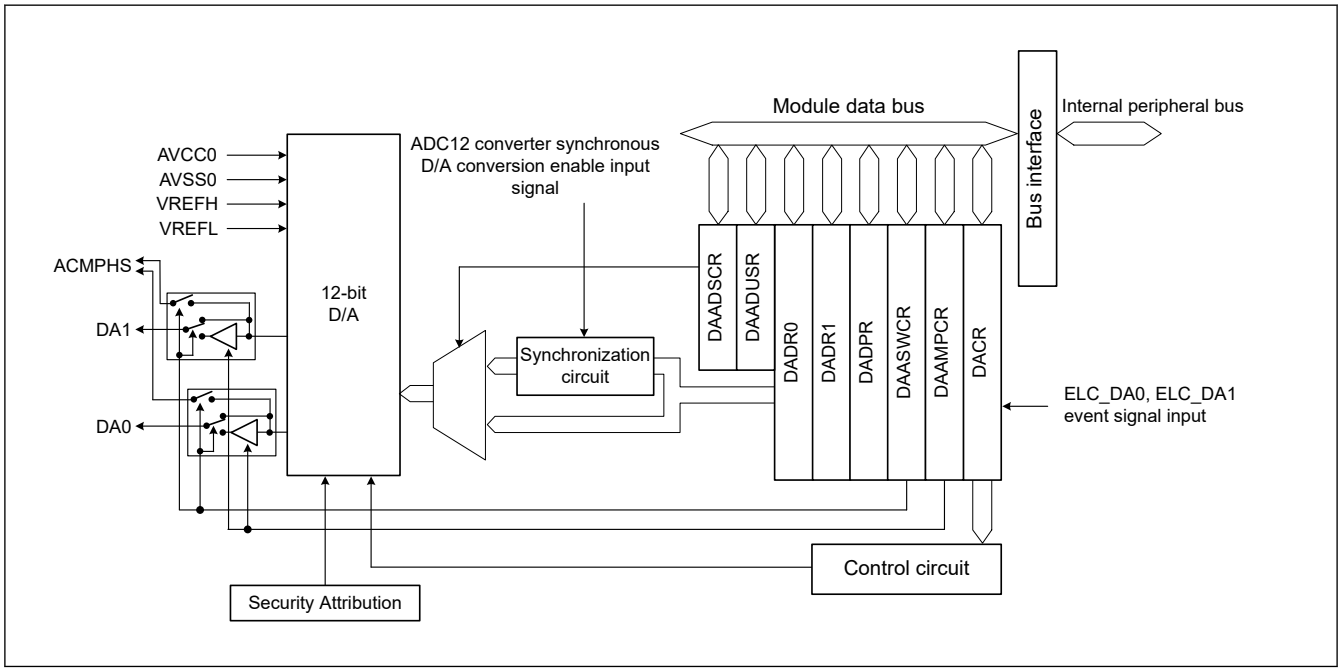


Figure 46.1 DAC12 block diagram

Table 46.2 lists the pin configuration of the DAC12.

Table 46.2 DAC12 I/O pins (1 of 2)

Pin name	I/O	Function
AVCC0	Input	• Analog power and analog reference top voltage supply pin for TSN, ADC12, and DAC12. • Connect to VCC when these modules are not used.
AVSS0	Input	• Analog ground and analog reference ground supply pin for TSN, ADC12, and DAC12. • Connect to VSS when these modules are not used.
VREFH	Input	Analog reference top voltage supply pin for the ADC12 (unit 1) and the DAC12

Table 46.2 DAC12 I/O pins (2 of 2)

Pin name	I/O	Function
VREFL	Input	Analog reference ground pin for the ADC12 (unit 1) and the DAC12
DA0	Output	Channel 0 output pin for the analog signals processed by the DAC12
DA1	Output	Channel 1 output pin for the analog signals processed by the DAC12

46.2 Register Descriptions

46.2.1 DADRn : D/A Data Register n (n = 0, 1)

Base address: DAC12 = 0x4033_3000
DAC12_NS = 0x5033_3000

Offset address: 0x00 + 0x02 × n

Bit position:	15															0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: S-TYPE-3, P-TYPE-3

DADRn register is 16-bit read/write registers that store data for D/A conversion. When an analog output is enabled, the values in DADRn are converted and output to the analog output pins.

12-bit data can be formatted as left- or right-justified in the DADPR.DPSEL bit setting. In right-justified format (DADPR.DPSEL = 0), the lower 12 bits, [11:0], are valid. In left-justified format (DADPR.DPSEL = 1), the upper 12 bits, [15:4], are valid.

46.2.2 DACR : D/A Control Register

Base address: DAC12 = 0x4033_3000
DAC12_NS = 0x5033_3000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DAOE1	DAOE0	DAE	—	—	—	—	—
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Function	R/W
4:0	—	These bits are read as 1. The write value should be 1.	R/W
5	DAE*1	D/A Enable 0: Control D/A conversion of channels 0 and 1 individually 1: Control D/A conversion of channels 0 and 1 collectively	R/W
6	DAOE0	D/A Output Enable 0 0: Disable D/A conversion and analog output of channel 0 (DA0) 1: Enable D/A conversion and analog output of channel 0 (DA0)	R/W
7	DAOE1	D/A Output Enable 1 0: Disable D/A conversion and analog output of channel 1 (DA1) 1: Enable D/A conversion and analog output of channel 1 (DA1)	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit controls D/A conversion and analog output in combination with the DAOEi bit (i = 0, 1), which controls the output of the conversion results. For details, see Table 46.3.

Table 46.3 D/A conversion controls

DAE	DAOE1	DAOE0	Description
0	0	0	Disable D/A conversion and analog output pins (DA0, DA1)*1
		1	• Enable D/A conversion of channel 0 and disable D/A conversion of channel 1 • Enable analog output of channel 0 (DA0) and disable analog output of channel 1 (DA1)*1
	1	0	• Disable D/A conversion of channel 0 and enable D/A conversion of channel 1 • Disable analog output of channel 0 (DA0)*1 and enable analog output of channel 1 (DA1)
		1	• Enable D/A conversion of channels 0 and 1 • Enable analog output of channels 0 and 1 (DA0, DA1)
1	x	x	• Enable D/A conversion of channels 0 and 1 • Collective enable analog output of channels 0 and 1 (DA0, DA1)

Note: x: Don't care
Note 1. When analog output is disabled, the analog output signal is placed in the Hi-Z state.

Only set this register while the ADC12 is halted when the DAADSCR.DAADST bit is 1 (interference reduction between D/A and A/D conversion is enabled). Only set DACR while the ADCSR.ADST bit is 0 and after selecting the software trigger, for the ADC12 trigger to securely stop the ADC12. This MCU only supports ADC12 (unit 1).

DAE bit (D/A Enable)

The DAE bit controls D/A conversion, amplifier operation, and analog output in combination with the DAOEi bit (i = 0, 1), the DAAMPCR.DAAMPi bit (i = 0, 1) and the DAASWCR.DAASWi bit (i = 0, 1). See Table 46.4.

When interference reduction between D/A and A/D conversions is enabled (DAADSCR.DAADST = 1), set the ADCSR.ADST bit of the ADC12 (unit 1) to 0. Then, select the software trigger for the ADC12 (unit 1) trigger to securely stop the ADC12 (unit 1).

DAOEi bit (D/A Output Enable i)

The DAOEi bit (i = 0, 1) controls D/A conversion, amplifier operation, and analog output in combination with the DAE bit, DAAMPCR.DAAMPi bit (i = 0, 1) and the DAASWCR.DAASWi bit (i = 0, 1). See Table 46.4.

When both the DAOEi bit (i = 0, 1) and DAE bit are 0, D/A conversion of channel i (i = 0, 1) is not processed, and no conversion result is output.

When interference reduction between D/A and A/D conversions is enabled (DAADSCR.DAADST = 1), set the DAOEi bit while the ADCSR.ADST bit of the ADC12 (unit 1) is set to 0. Then, select the software trigger for the ADC12 (unit 1) trigger to securely stop the ADC12 (unit 1).

The event link function can be used to set the DAOEi bit to 1. The DAOE0 bit is set to 1 when the event specified in the ELSR12 register of the ELC (ELC_DA0 event) occurs, and output of the D/A conversion results starts. The DAOE1 bit is set to 1 when the event specified in the ELSR13 register of the ELC (ELC_DA1 event) occurs, and output of the D/A conversion results starts.

Table 46.4 D/A conversion and analog output control

DACR		DAAMPCR	DAASWCR	Channel i operation	Amplifier operation of channel i	Analog external output of channel i*1	Analog internal output of channel i*2
DAE	DAOEi	DAAMPi	DAASWi				
0	0	x	x	Stop	Stop	Hi-Z	Hi-Z
0	1	0	0	Run	Stop	Amplifier-through	Hi-Z
0	1	0	1	Run	Stop	Hi-Z	Amplifier-through
0	1	1	0	Run	Run	Amplifier output	Hi-Z
0	1	1	1	Run	Run	Hi-Z	Hi-Z
1	x	0	0	Run	Stop	Amplifier-through	Hi-Z
1	x	0	1	Run	Stop	Hi-Z	Amplifier-through
1	x	1	0	Run	Run	Amplifier output	Hi-Z
1	x	1	1	Run	Run	Hi-Z	Hi-Z

Note: x : Don't care
Note 1. output to pin
Note 2. output to ACMPHS

46.2.3 DADPR : DADRn Format Select Register

Base address: DAC12 = 0x4033_3000
DAC12_NS = 0x5033_3000

Offset address: 0x05

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DPSEL	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	DPSEL	DADRn Format Select 0: Right-justified format 1: Left-justified format	R/W

Note: S-TYPE-3, P-TYPE-3

46.2.4 DAADSCR : D/A A/D Synchronous Start Control Register

Base address: DAC12 = 0x4033_3000
DAC12_NS = 0x5033_3000

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DAADST	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	DAADST	D/A A/D Synchronous Conversion 0: Do not synchronize DAC12 with ADC12 (unit 1) operation (disable interference reduction between D/A and A/D conversion). 1: Synchronize DAC12 with ADC12 (unit 1) operation (enable interference reduction between D/A and A/D conversion).	R/W

Note: S-TYPE-3, P-TYPE-3

To minimize interference between D/A and A/D conversion, the DAADSCR register enables synchronization of the start timing of D/A conversion with the ADC12 synchronous D/A conversion enable input signal.

Only set this register while the ADC12 (unit 1) is halted, that is, while the ADCSR.ADST bit is 0 after selecting the software trigger as the ADC12 (unit 1) trigger.

Select unit 1 as the target ADC12 unit before setting the DAADST bit to 1. Set DAADUSR.AMADSEL1 bit to 1 to select unit 1. This MCU only supports ADC12 unit 1.

DAADST bit (D/A A/D Synchronous Conversion)

Setting the DAADST bit to 0 allows the DADRn register value to be converted into analog data at any time. Setting the DAADST bit to 1 allows synchronization of D/A conversion with the synchronous D/A conversion enable input signal from the ADC12 (unit 1). With this bit set, D/A conversion does not start until the ADC12 (unit 1) completes A/D conversion, even when the DADRn register is changed.

Set this bit while the ADCSR.ADST bit is set to 0. Then, select the software trigger for the ADC12 (unit 1) trigger to securely stop the ADC12 (unit 1). Set the DAADUSR.AMADSEL1 bit to 1 before setting the DAADST bit to 1.

The event link function cannot be used when the DAADST bit is set to 1. Stop the event link function by setting the ELSR12 and ELSR13 registers of the ELC. The setting of the DAADST bit is shared by channels 0 and 1 of the DAC12.

46.2.5 DAAMPCR : D/A Output Amplifier Control Register

Base address: DAC12 = 0x4033_3000
DAC12_NS = 0x5033_3000

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DAAMP1	DAAMP0	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	DAAMP0	Amplifier Control 0 0: Do not use channel 0 output amplifier 1: Use channel 0 output amplifier	R/W
7	DAAMP1	Amplifier Control 1 0: Do not use channel 1 output amplifier 1: Use channel 1 output amplifier	R/W

Note: S-TYPE-3, P-TYPE-3

The DAAMPCR register selects D/A output with or without using the amplifier.

DAAMP0 bit (Amplifier Control 0)

When the DAAMP0 bit is 0, analog values are output for D/A output of channel 0 without using the amplifier. When the DAAMP0 bit is 1, analog values are output for D/A output of channel 0 through the amplifier.

When both the DACR.DAE and DACR.DAOE0 bits are 0, the amplifier is not used regardless of the setting of the DAAMP0 bit. See Table 46.4 for details.

DAAMP1 bit (Amplifier Control 1)

When the DAAMP1 bit is 0, analog values are output for D/A output of channel 1 without using the amplifier. When the DAAMP1 bit is 1, analog values are output for D/A output of channel 1 through the amplifier.

When both the DACR.DAE and DACR.DAOE1 bits are 0, the amplifier is not used regardless of the setting of the DAAMP1 bit. See Table 46.4 for details.

46.2.6 DAASWCR : D/A Amplifier Stabilization Wait Control Register

Base address: DAC12 = 0x4033_3000
DAC12_NS = 0x5033_3000

Offset address: 0x1C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DAASW1	DAASW0	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	DAASW0	D/A Amplifier Stabilization Wait 0 and D/A internal output control 0: For output to external pin: Amplifier stabilization wait off (output) for channel 0 For output to internal module: Disable output for channel 0 1: For output to external pin: Amplifier stabilization wait on (high-Z) for channel 0 For output to internal module: Enable output for channel 0	R/W

Bit	Symbol	Function	R/W
7	DAASW1	D/A Amplifier Stabilization Wait 1 and D/A internal output control 0: For output to external pin: Amplifier stabilization wait off (output) for channel 1 For output to internal module: Disable output for channel 1 1: For output to external pin: Amplifier stabilization wait on (high-Z) for channel 1 For output to internal module: Enable output for channel 1	R/W

Note: S-TYPE-3, P-TYPE-3

The DAASWCR register controls D/A output with the output amplifier or D/A output for internal modules. This register is used in the initialization procedure to wait for stabilization of the D/A output amplifier. Each bit in DAASWCR should be set to 1 when both the DACR.DAE bit and the DACR.DAOEi (i = 0, 1) bit are 0. See section 46.7.5. Initialization Procedure with the Output Amplifier.

DAASW0 bit (D/A Amplifier Stabilization Wait 0)

Set the DAASW0 bit to 1 in the initialization procedure to wait for the stabilization of the D/A channel 0 output amplifier. When DAASW0 is set to 1, D/A conversion operates, but the conversion result of D/A is not output from channel 0 to the DA0 pin. When the DAASW0 bit is 0, the stabilization wait time stops, and the D/A conversion result of channel 0 is output through the output amplifier to the DA0 pin. When the amplifier is not used (DAAMPCR.DAAMP0 bit is 0) and DAASW0 is set to 1, D/A conversion result of channel 0 is output to the internal modules.

DAASW1 bit (D/A Amplifier Stabilization Wait 1)

Set the DAASW1 bit to 1 in the initialization procedure to wait for the stabilization of the D/A channel 1 output amplifier. When DAASW1 is set to 1, D/A conversion operates, but the conversion result of D/A is not output from channel 1 to the DA1 pin. When the DAASW1 bit is 0, the stabilization wait time stops, and the D/A conversion result of channel 1 is output through the output amplifier to the DA1 pin. When the amplifier is not used (DAAMPCR.DAAMP1 bit is 0) and DAASW1 is set to 1, D/A conversion result of channel 1 is output to the internal modules.

46.2.7 DAADUSR : D/A A/D Synchronous Unit Select Register

Base address: DAC12 = 0x4033_3000
DAC12_NS = 0x5033_3000

Offset address: 0x10C0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	AMADSEL1	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	AMADSEL1	A/D Unit 1 Select 0: Do not select unit 1 1: Select unit 1	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The DAADUSR register selects the target ADC12 unit for D/A and A/D synchronous conversions. Set the AMADSEL1 bit to 1 to select unit 1 as the target synchronous unit for the MCU. When setting the DAADSCR.DAADST bit to 1 for synchronous conversions, select the target unit in this register in advance.

Only set the DAADUSR register while the ADCSR.ADST bit of the ADC12 is set to 0 and the DAADSCR.DAADST bit is set to 0.

46.3 Operation

The DAC12 includes D/A conversion circuits for two channels, each of which can operate independently. When the DAOEn bit (n = 0, 1) in DACR is set to 1, DAC12 is enabled and the conversion result is output.

This following example shows D/A conversion on channel 0. Figure 46.2 shows the timing of this operation.

To process D/A conversion on channel 0:

1. Set the data for D/A conversion in the DADR0 register and the data format in the DADPR.DPSEL bit.
2. Set the DACR.DAOE0 bit to 1 to start D/A conversion. The conversion result is output from the analog output pin DA0 after the conversion time t_{DCONV} elapses. The conversion result continues to be output until DADR0 is written to again or the DAOE0 bit is set to 0. The output value (reference) is expressed by the following formula:

$$\frac{\text{Setting in DADR0}}{4096} \times \text{VREFH}$$

3. To start conversion again, write another value to DADR0. The conversion result is output after the conversion time t_{DCONV} elapses.
   When the DAADSCR.DAADST bit is 1 (interference reduction between D/A and A/D conversion is enabled), a maximum of one A/D conversion time is required for D/A conversion to start. When ADCLK is faster than the peripheral clock, a longer time might be required.
4. To disable analog output, set the DAOE0 bit to 0.

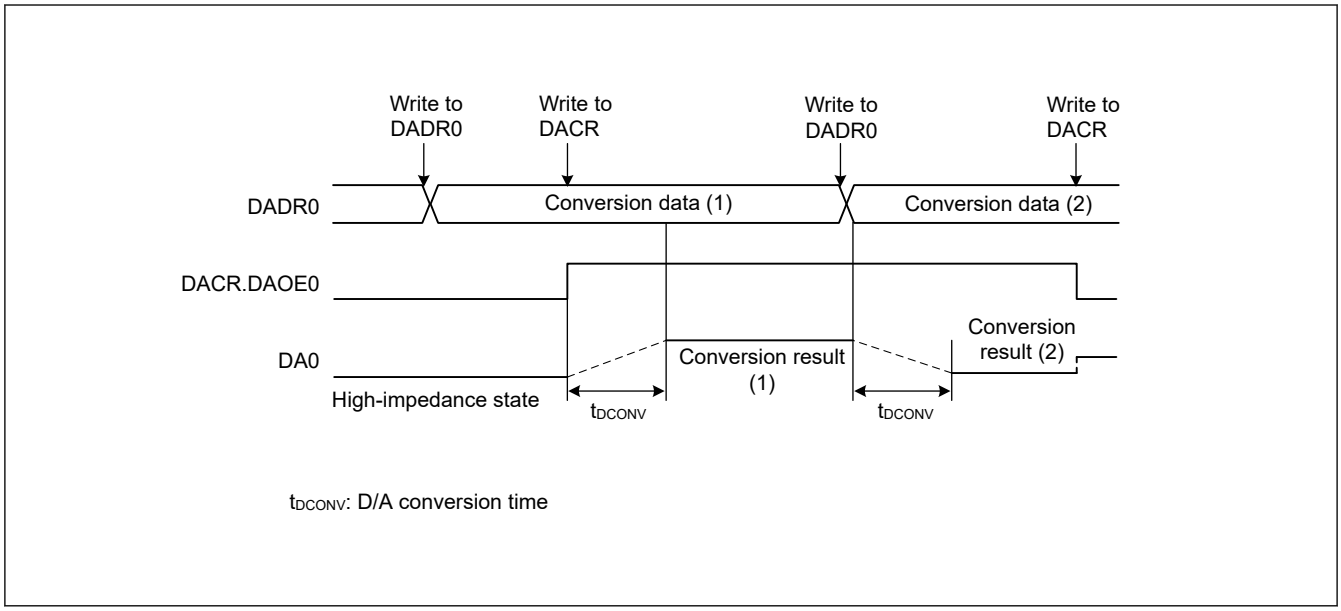


Figure 46.2 Example of DAC12 operation

46.3.1 Reducing Interference between D/A and A/D Conversion

When D/A conversion starts, the DAC12 generates inrush current. Because the DAC12 and ADC12 (unit 1) share the same analog power supply, the generated inrush current can interfere with ADC12 (unit 1) operation.

While the DAADSCR.DAADST bit is 1, D/A conversion does not start immediately on updating the DADRm register. Instead:

- If the DADRm register data is modified while the ADC12 is halted, D/A conversion starts in 1 PCLKA cycle.
- If the DADRm register data is modified while the ADC12 is performing a 12-bit A/D conversion, D/A conversion starts on A/D conversion completion. Therefore, it takes up to one A/D conversion time period for the DADRm register data update to be reflected as the D/A conversion circuit output. Until the D/A conversion completes, the DADRm register value does not correspond to the analog output value.

When the DAADSCR.DAADST bit is 1, it is not possible to check through software whether the DADRm register value was D/A-converted.

The following sequence provides an example of channel 0 D/A conversion, in which the DAC12 is synchronized with the ADC12 (unit 1). Figure 46.3 shows the timing of this operation.

To perform D/A conversion on channel 0 in synchronization with the ADC12 (unit 1):

1. Confirm that the ADC12 (unit 1) is halted and set the DAADUSR.AMADSEL1 bit to 1.

2. Confirm that the ADC12 (unit 1) is halted and set the DAADSCR.DAADST bit to 1.
3. Confirm that the ADC12 (unit 1) is halted and set the DACR.DAOE0 bit to 1.
4. Set the DADR0 register. If ADCLK is faster than the peripheral clock, D/A conversion might be delayed for longer than one A/D conversion time.
   - If the ADC12 (unit 1) is halted (ADCSR.ADST = 0) when the DADR0 register is modified, D/A conversion starts in 1 PCLKA cycle.
   - If the 12-bit A/D conversion is in progress (ADCSR.ADST = 1) when the DADR0 register is modified, D/A conversion starts on A/D conversion completion. If the DADR0 register is modified twice during A/D conversion, the first update might not be converted.

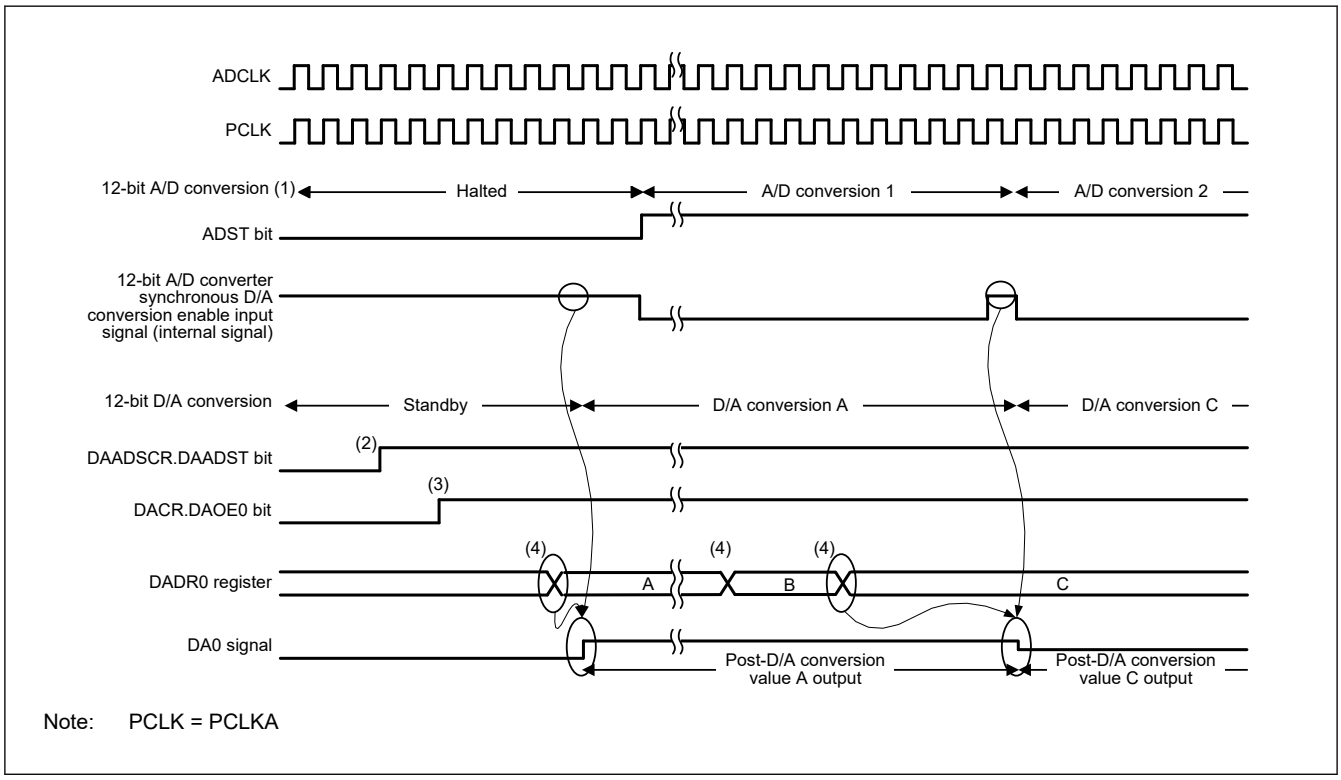


Figure 46.3 Example conversion when DAC12 is synchronized with ADC12 (unit 1)

When ADCLK is faster than PCLKA, the DAC12 might not be able to capture the synchronous D/A conversion enable input signal from the ADC12 (unit 1) during the 1 ADCLK cycle that is output between A/D conversion 1 and A/D conversion 2, as shown in Figure 46.4. In this case, post-D/A conversion value A is continuously output as the DA0 signal.

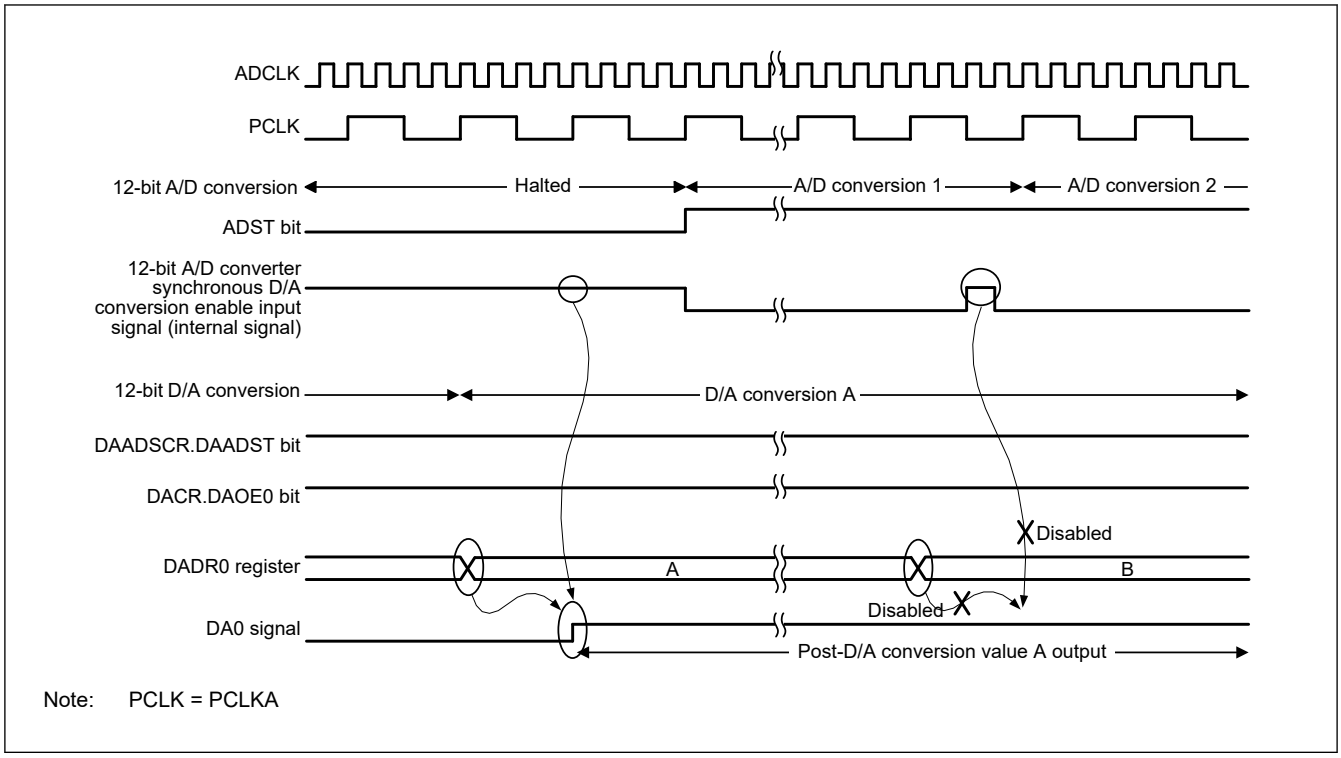


Figure 46.4 Example when the DAC12 cannot capture the synchronous D/A conversion enable input signal from the ADC12 (unit 1)

46.4 Event Link Operation Setting Procedure

This section describes the procedures used in event link operation.

46.4.1 DA0 Event Link Operation Setting Procedure

To set up DA0 event link operation:

1. Set the DADPR.DPSEL bit and the data for D/A conversion in the DADR0 register.
2. Set the ELC_DA0 event signal to be linked to each peripheral module in the ELSR12 register.
3. Set the ELCR.ELCON bit to 1. This enables event link operation for all modules with the event link function selected.
4. Set the event output source module to activate the event link. After the event is output from the module, the DACR.DAOE0 bit becomes 1, and D/A conversion starts on channel 0.
5. Set the ELSR12 register to 0x0000 to stop event link operation of DAC12 channel 0. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

46.4.2 DA1 Event Link Operation Setting Procedure

To set up DA1 event link operation:

1. Set the DADPR.DPSEL bit and set the data for D/A conversion in the DADR1 register.
2. Set the ELC_DA1 event signal to be linked to each peripheral module in the ELSR13 register.
3. Set the ELCR.ELCON bit to 1. This enables the event link operation for all modules with the event link function selected.
4. Set the event output source module to activate the event link. After the event is output from the module, the DACR.DAOE1 bit is becomes 1, and D/A conversion starts on channel 1.
5. Set the ELSR13 register to 0x0000 to stop event link operation on DAC12 channel 1. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

46.5 Analog Output Control by Security Attribution

Analog output to DA0/DA1 pin is controlled by the security attribution of Port and DAC12 as shown in Table 46.5.

Analog output to ACMPHS is not controlled by the security attribution.

Table 46.5 Analog output control by security attribution

Security Attribution			Analog Output	
DAC12 (PSARD)	P014 (PmSAR (m = 0~9, A, B))	P015 (PmSAR (m = 0~9, A, B))	P014 pin output	P015 pin output
0 (secure)	1 (non-secure)	1 (non-secure)	enable	enable
0 (secure)	1 (non-secure)	0 (secure)	enable	enable
0 (secure)	0 (secure)	1 (non-secure)	enable	enable
0 (secure)	0 (secure)	0 (secure)	enable	enable
1 (non-secure)	1 (non-secure)	1 (non-secure)	enable	enable
1 (non-secure)	1 (non-secure)	0 (secure)	enable	disable
1 (non-secure)	0 (secure)	1 (non-secure)	disable	enable
1 (non-secure)	0 (secure)	0 (secure)	disable	disable

46.6 Usage Notes on Event Link Operation

- When the event link function is used, do not use the amplifier output function.
- When the event link function is used, set the DACR.DAE bit to 0.
- When the event specified for the ELC_DA0 event signal is generated while a write to the DACR.DAOE0 bit is performed, the write cycle is stopped, and the generated event takes precedence in setting the bit to 1.
- When the event specified for the ELC_DA1 event signal is generated while a write to the DACR.DAOE1 bit is performed, the write cycle is stopped, and the generated event takes precedence in setting the bit to 1.
- Use of the event link function is prohibited when the DAADSCR.DAADST bit is set to 1 to reduce interference between D/A and A/D conversions.

46.7 Usage Notes

46.7.1 Settings for the Module-Stop Function

DAC12 operation can be disabled or enabled using the Module Stop Control Register. The DAC12 is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

46.7.2 DAC12 Operation in the Module-Stop State

When the MCU enters the module-stop state with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current must be reduced in the module-stop state, disable D/A conversion by setting the DACR.DAOE1, DAOE0, and DAE bits to 0.

46.7.3 DAC12 Operation in Software Standby Mode

When the MCU enters Software Standby mode with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current must be reduced in Software Standby mode, disable D/A conversion by setting the DACR.DAOE1, DAOE0, and DAE bits to 0.

46.7.4 Constraint on Entering Deep Software Standby Mode

When the MCU enters Deep Software Standby mode with D/A conversion enabled, the outputs of the DAC12 are placed in a high impedance state.

46.7.5 Initialization Procedure with the Output Amplifier

Use the following initialization procedures with the output amplifier. The example shows the case for channel 0.

To initialize the DAC12 with the output amplifier:

1. Write 0x0000 to the DADR0 register.
2. Set the DAASWCR.DAASW0 bit to 1.
3. Set the DAAMPCR.DAAMP0 bit to 1.
4. Set the DACR.DAE bit or the DACR.DAOE0 bit to 1 to start operation of the amplifier.
5. Clear the DAASWCR.DAASW0 bit to 0 after waiting for the duration of D/A conversion time t_{DCONV}.
6. Write the value to be converted in the DADR0 register.

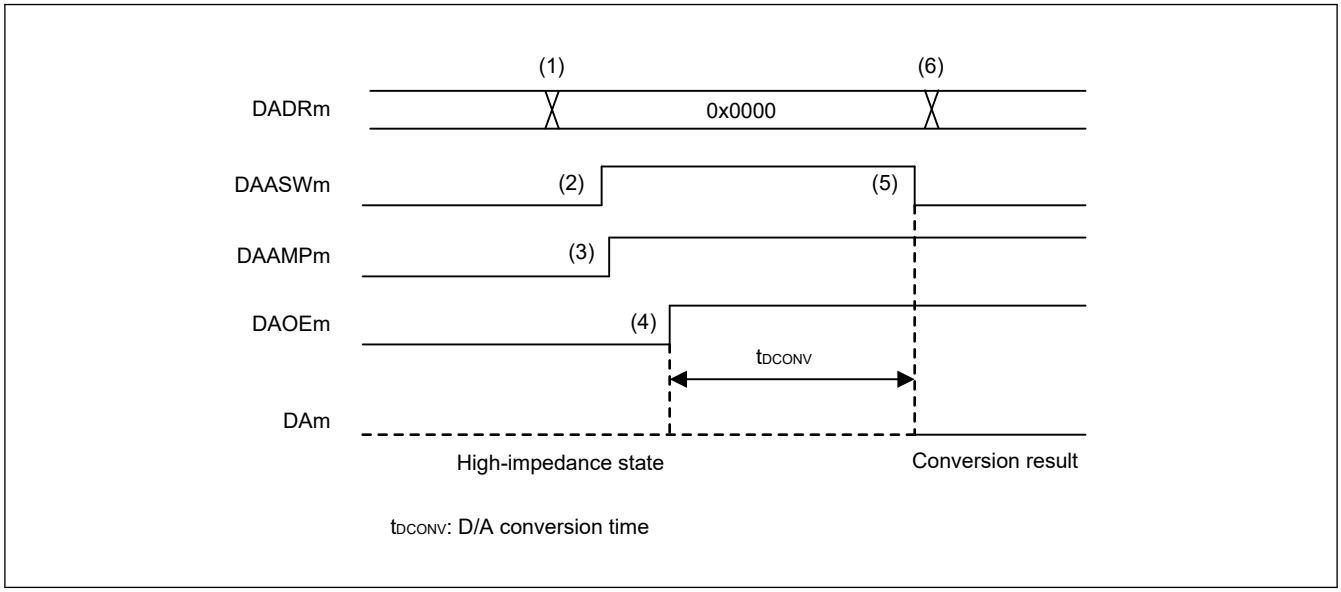


Figure 46.5 Example of the initial flow with the output amplifier in DAC12

While the amplifier is running, clearing the DACR.DAE and DACR.DAOE0 bits to 0 allows the amplifier to stop operation. To use the amplifier again, repeat steps 1 to 6.

46.7.6 Initialization Procedure of the Output to internal modules

Use the following initialization procedures for the output to internal modules.

The example shows the case for channel 0.

1. Set the DAASWCR.DAASW0 bit to 1.
2. Set the DACR.DAE bit or the DACR.DAOE0 bit to 1.
3. Write the value to be converted in the DADR0 register.

When you stop output of the DAC12, set DAASWCR.DAASWi to 0 after setting DACR.DAE to 0 or DACR.DAOEi to 0.

46.7.7 Constraint on Usage When Interference Reduction between D/A and A/D Conversion Is Enabled

When the DAADSCR.DAADST bit is 1, enabling interference reduction between D/A and A/D conversion, do not place the ADC12 in the module-stop state. Doing so can halt D/A conversion in addition to A/D conversion.

47. Temperature Sensor (TSN)

47.1 Overview

The on-chip Temperature Sensor (TSN) determines and monitors the die temperature for reliable operation of the device. The sensor outputs a voltage directly proportional to the die temperature, and the relationship between the die temperature and the output voltage is fairly linear. The output voltage is provided to the ADC12 for conversion and can be further used by the end application.

Table 47.1 lists the TSN specifications, and Figure 47.1 shows a block diagram.

Table 47.1 TSN specifications

Item	Description
Temperature sensor voltage output	Temperature sensor outputs a voltage to the 12-bit A/D converter
Module-stop function	Module-stop state can be set to reduce power consumption
Temperature sensor calibration data	Reference data measured for each chip at factory shipment is stored in a register
TrustZone Filter	Security and Privilege attribution can be set

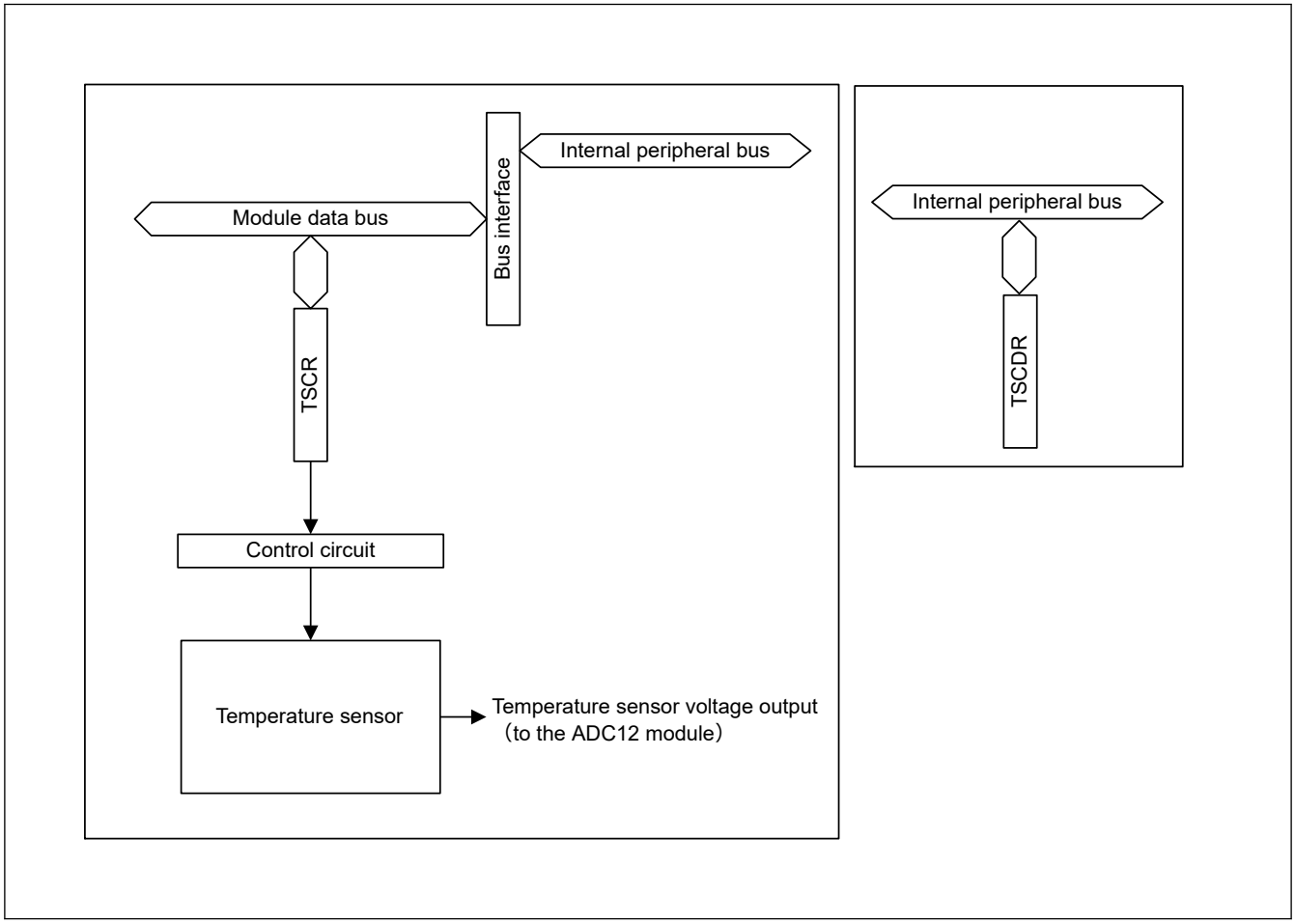


Figure 47.1 TSN block diagram

47.2 Register Descriptions

47.2.1 TSCR : Temperature Sensor Control Register

Base address: TSN = 0x4023_5000
TSN_NS = 0x5023_5000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TSEN	—	—	TSOE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	TSOE	Temperature Sensor Output Enable 0: Disable output from the temperature sensor to the ADC12 1: Enable output from the temperature sensor to the ADC12	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	TSEN	Temperature Sensor Enable 0: Stop the temperature sensor 1: Start the temperature sensor.	R/W

Note: S-TYPE3, P-TYPE3

The TSCR is a register which controls the temperature sensor. The timing constraints shown in Figure 47.3 apply to the settings of the TSCR register.

TSOE bit (Temperature Sensor Output Enable)

The TSOE bit enables or disables the temperature sensor output to ADC12.

TSEN bit (Temperature Sensor Enable)

The TSEN bit starts or stops the temperature sensor.

47.2.2 TSCDR : Temperature Sensor Calibration Data Register

Base address: TSD = 0x4011_B000
TSD_NS = 0x5011_B000

Offset Address: 0x017C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TSCDR[15:0]															
Value after reset:	Chip-specific value															

Bit	Symbol	Function	R/W
15:0	TSCDR[15:0]	Temperature Sensor Calibration Data Chip-specific value	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE5, P-TYPE5

The TSCDR register stores temperature sensor calibration data measured for each chip at factory shipment.

Temperature sensor calibration data is the output voltage of the temperature sensor under the conditions Tj = 125°C and AVCC0 = VREFH0 = 3.3 V converted to a digital value by the 12-bit A/D converter.

The TSCDR register is a read-only 32-bit register. Read from this register in 32-bit units.

Temperature sensor calibration data is stored in the lower 12 bits of the TSCDR register.

47.3 Using the Temperature Sensor

The temperature sensor outputs a voltage that varies with the temperature. This voltage is converted to a digital value by the 12-bit A/D converter. To obtain the die temperature, convert this value into the temperature.

47.3.1 Preparation for Using the Temperature Sensor

The ambient temperature (T) is proportional to the temperature sensor voltage output (Vs), so ambient temperature is calculated with the following formula:

T = (Vs - V1) / slope + T1

- T: Ambient temperature of MCU as calculation result (°C)
- Vs: Voltage output by the temperature sensor on temperature measurement (V)
- T1: Temperature experimentally measured at one point (°C)
- V1: Voltage output by the temperature sensor on measurement of T1 (V)
- T2: Temperature experimentally measured at a second point (°C)
- V2: Voltage output by the temperature sensor on measurement of T2 (V)
- Slope: Temperature gradient of the temperature sensor (V / °C), slope = (V2 - V1) / (T2 -T1)

Characteristics vary between sensors, so Renesas recommends measuring two different sample temperatures as follows:

1. Use the 12-bit A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1.
2. Again use the 12-bit A/D converter to measure the voltage V2 output by the temperature sensor at a different temperature T2.
3. Obtain the temperature gradient (slope = (V2 - V1) / (T2 - T1)) from these results.
4. Subsequently, obtain temperatures by substituting the slope into the formula for the temperature characteristic (T = (Vs -V1) / slope + T1).

In order to make the temperature Ta and Tj as close as possible, perform the measurement with the power consumption of the MCU as low as possible.

If you are using the temperature gradient given in section 60, Electrical Characteristics, use the A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1, then calculate the temperature characteristic using the following formula:

T = (Vs - V1) / slope + T1

Note: This method produces less accurate temperatures than measurement at two points.

In this MCU, the TSCDR register stores the temperature value (CAL125) of the temperature sensor measured under the condition Ta = Tj = 125°C and AVCC0 = VREFH0 = 3.3 V. If you use this value as the sample measurement result at the first point, you can omit the preparation before using the temperature sensor.

V1 is calculated from CAL125:

V1 = 3.3 × CAL125 / 4096 [V] (In case of 12 bit accuracy)

Using this value, the measured temperature can be calculated according to the following formula:

T = (Vs - V1) / slope + 125 [°C]

- T: Ambient temperature of MCU as calculation result (°C)
- Vs: Voltage output by the temperature sensor when the temperature is measured (V)
- V1: Voltage output by the temperature sensor when Ta = Tj = 125°C and AVCC0 = VREFH0 = 3.3 V (V)

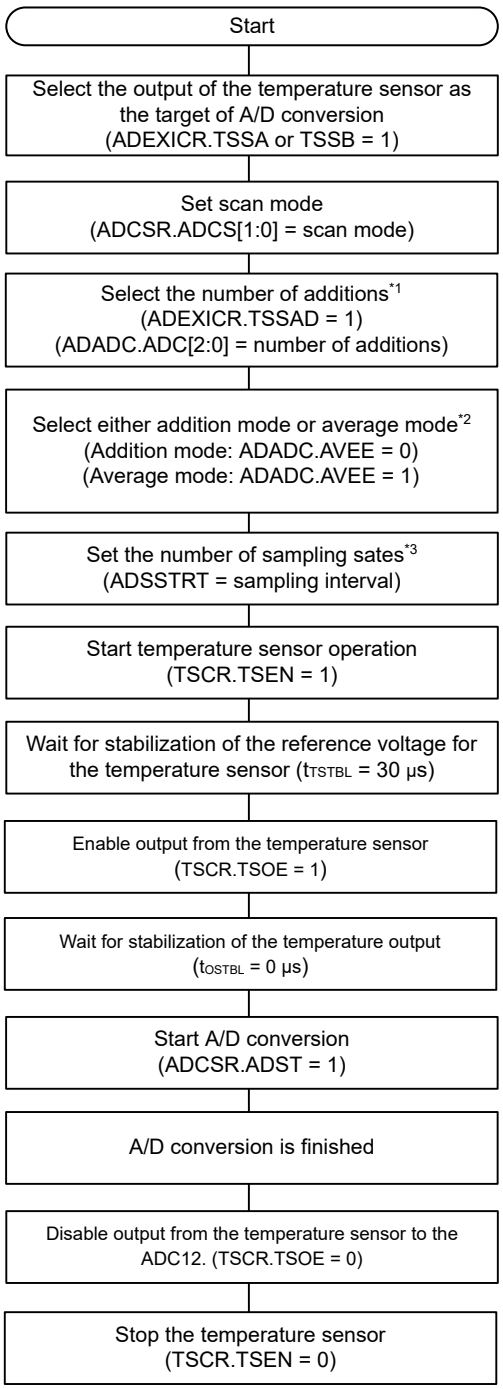
- Slope: Temperature gradient of the temperature sensor*1 / 1000 (V/°C)

Note 1. See section 60, Electrical Characteristics

47.3.2 Procedures for Using the Temperature Sensor

Figure 47.2 shows the procedure for using the TSN.

For details, see section 45, 12-Bit A/D Converter (ADC12).



Note 1. This setting is not required if addition/average mode is not set.
Note 2. The ADADC.ADC[2:0] setting is limited to some values in additions/average mode. For details on the available ADADC.ADC[2:0] settings in additions/average mode, see section 45, 12-Bit A/D Converter (ADC12).
Note 3. Set the sampling time to more than the value described in section 60, Electrical Characteristics.

Figure 47.2 Procedure example for using the TSN

Figure 47.3 shows the timing from the start of temperature sensor operation until the completion of A/D conversion when the ADC12 is in single scan mode (the conversion target is the temperature sensor output only). The times shown in the figure are described in Table 47.2

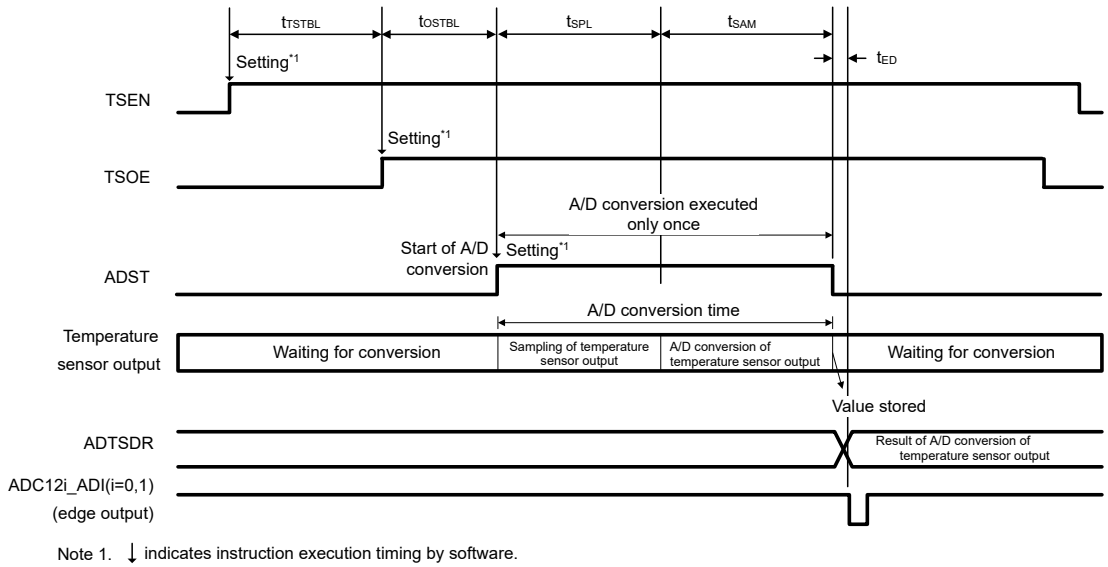


Figure 47.3 Timing from start of temperature sensor operation until completion of A/D conversion

Table 47.2 Time until completion of A/D conversion after start of temperature sensor operation

Parameter	Symbol	Time
Wait time for temperature sensor reference voltage stabilization	t_{TSTBL}	30 µs (min)
Wait time for temperature sensor output stabilization	t_{OSTBL}	0 µs (min)
A/D converter input sampling time	t_{SPL}	ADSSTRn setting × ADCLK period
A/D conversion time	t_{SAM}	See the table in section 45.3.6. Analog Input Sampling and Scan Conversion Time.
Scan conversion end delay	t_{ED}	

47.4 Usage Notes

47.4.1 Settings for the Module-Stop Function

TSN operation can be disabled or enabled using the associated bit in Module Stop Control Register D (MSTPCRD). The TSN is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

47.4.2 Settings for the Software Standby Mode

Before entering Software Standby Mode, set TSCR.TSOE bit to 0 after stopping A/D conversion, then set TSCR.TSEN bit to 0 to stop the TSN.

48. High-Speed Analog Comparator (ACMPHS)

48.1 Overview

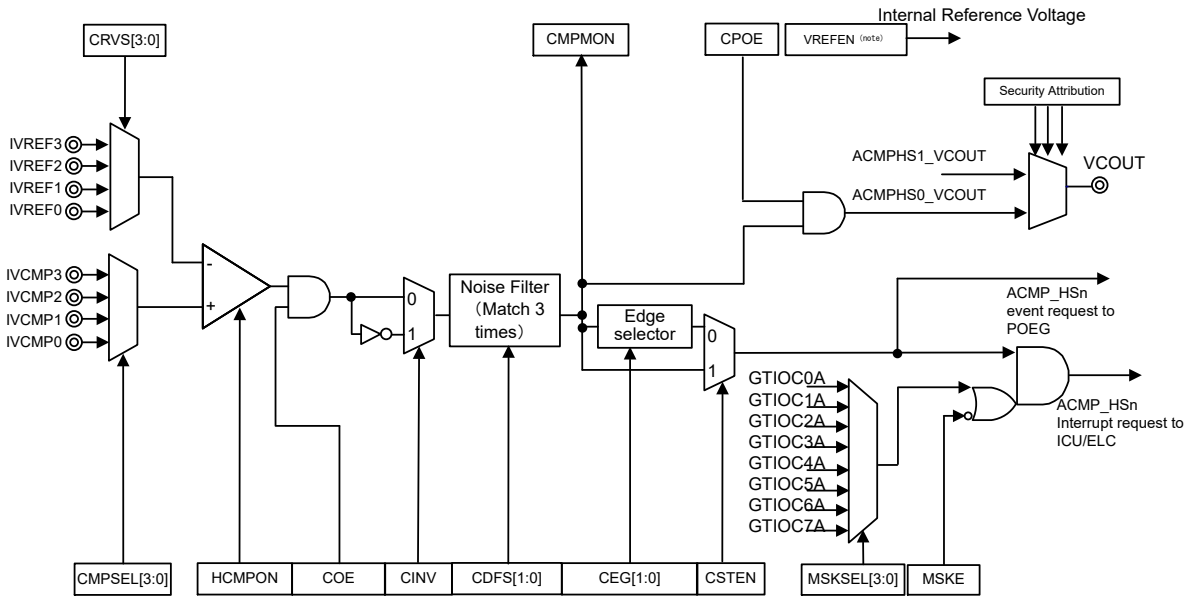
The High-Speed Analog Comparator (ACMPHS) can be used to compare an analog input voltage with a reference voltage and to provide a digital output based on the result of conversion. Both the analog input voltage and the reference voltage can be provided to the ACMPHS from internal sources (D/A converter output or internal reference voltage) and an external source. Such flexibility is useful in applications that require go/no-go comparisons to be performed between analog signals without necessarily requiring A/D conversion.

Table 48.1 lists the ACMPHS specifications, Figure 48.1 shows a block diagram, and Table 48.2 shows the input source configurations.

Table 48.1 ACMPHS specifications

Parameter	Specifications
Number of channels	2 channels: ACMPHSn (n = 0, 1)
Analog input voltage	• Output from internal D/A converter • Input from an external source (compatible with internal A/D converter input pin (one selectable))
Reference voltage	• Internal reference voltage (Vref) • Output from internal D/A converter • Input from an external source (compatible with internal A/D converter input pin (one selectable))
ACMPHS output	• Comparison result output to terminal PIN • Generation of ELC event output • Monitor output from register • Generation of interrupt request output
Interrupt request signal	• Interrupt request generated on valid edge detection from comparison result (when CMPCTL.CSTEN = 0[*1]) • Selectable to rising edge, falling edge, or both edges (when CMPCTL.CSTEN = 0[*1]) • Rising edge only (when CMPCTL.CSTEN = 1[*1])
Digital filter function	• Selectable to one of three sampling frequencies • Not using the filter function is selectable
Module-stop function	Module-stop state can be set for each groups to reduce power consumption.
TrustZone Filter	Security and Privilege attribution can be set for each group

Note 1. Interrupt request signal selection restriction detail refers to section 48.5. ACMPHS Interrupts and section 48.6. ACMPHS Output to the Event Link Controller (ELC).



n = 0～1

CMPSEL[3:0] : bits in ACMPHSn.CMPSEL0 register
CRVS[3:0] : bits in ACMPHSn.CMPSEL1 register
CPOE, VREFEN : bits in ACMPHSn.CPIOC register
CMPMON : bits in ACMPHSn.CMPMON register
HCMPON, COE, CINV, CDFS[1:0], CEG[1:0], CSTEN : bits in ACMPHSn.CMPCTL register
MSKSEL[3:0] : bits in ACMPHSn.CMPSKCTL register
MSKE : bits in ACMPHSn.CMPINTCTL register

Note : VREFEN exists only in ACMPHS0.

Figure 48.1 ACMPHS block diagram

Table 48.2 Input source configuration of the ACMPHS

Comparator	Reference voltage input source				Analog voltage input source				Output pin
	IVREF3	IVREF2	IVREF1	IVREF0	IVCMP3	IVCMP2	IVCMP1	IVCMP0	
ACMPHS0	DA0*3	Vref*1	AN104	AN101	AN002	AN000	DA1*4	AN005	VCOUT*2
ACMPHS1	DA0*3	Vref*1	AN104	AN101	AN102	AN100	DA1*4	AN016/ AN116	

Note 1. Internal voltage reference.
Note 2. ACMPHS0 to ACMPHS1 compare outputs are bundled with the VCOUT pin.
Note 3. When D/A converter 0 output is not used, the signal can be used as AN007 analog input.
Note 4. When D/A converter 1 output is not used, the signal can be used as AN105 analog input.

48.2 Register Descriptions

48.2.1 CMPCTL : Comparator Control Register

Base address: ACMPHSn = 0x4023_6000 + 0x0100 × n (n = 0, 1)
ACMPHSn_NS = 0x5023_6000 + 0x0100 × n (n = 0, 1)

Offset address: 0x000

Bit position:	7	6	5	4	3	2	1	0
Bit field:	HCMPON	CDFS[1:0]		CEG[1:0]		CSTEN	COE	CINV
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CINV	Comparator Output Polarity Selection*1 *2 0: Do not invert comparator output 1: Invert comparator output	R/W
1	COE	Comparator Output Enable 0: Disable comparator output (output signal is low level) 1: Enable comparator output	R/W
2	CSTEN	Interrupt Select*3 0: Output through the edge selector 1: Output directly	R/W
4:3	CEG[1:0]	Selection of Valid Edge (Edge Selector) 0 0: Do not detect edge 0 1: Detect rising edge 1 0: Detect falling edge 1 1: Detect both edges	R/W
6:5	CDFS[1:0]	Noise Filter Selection *1 *2 *3 *4 0 0: Do not use noise filter 0 1: Use noise filter sampling frequency of PCLKB/2^3 1 0: Use noise filter sampling frequency of PCLKB/2^4 1 1: Use noise filter sampling frequency of PCLKB/2^5	R/W
7	HCMPON	Comparator Operation Control*5 0: Stop operation (comparator outputs a low-level signal) 1: Enable operation (enables input to the comparator pins)	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Disable the ACMPHS output (COE= 0) before changing the CDFS[1:0] and CINV bits.
Note 2. If the CDFS[1:0] and CINV bits are changed, an ACMPHS interrupt request and an ELC event might be generated. Before changing these bits, set the ELSRn register to 0 (the ACMPHS output is not linked). After changing these bits, clear the IR flag in the IELSRn register to 0 to clear the interrupt status.
Note 3. Set the CSTEN bit to 1 and the CDFS[1:0] bits to 00b if the ACMPHS interrupt causes release of Software Standby mode. CSTEN is supported only by the ACMPHS0. ACMPHS1.CMPCTL.CSTEN must be set to 0.
Note 4. If the CDFS[1:0] bits are changed from 00b (noise filter not used) to a value other than 00b (noise filter used), perform sampling four times and update the filter output, and then use the ACMPHS interrupt request or the ELC event.
Note 5. A stabilization wait time is required to permit ACMPHS operation after enabling it (HCMPON = 1). The operation stabilization wait time for ACMPHS modules 0 to 1 is 300 ns.

Note: Set this register before setting registers in the POEG when using comparator output as a POEG source.

The CMPCTL register controls the ACMPHS operation, enables or disables the ACMPHS output, selects the noise filter, selects the valid edge of the interrupt signal, and selects the interrupt.

48.2.2 CMPSEL0 : Comparator Input Select Register

Base address: ACMPHSn = 0x4023_6000 + 0x0100 × n (n = 0, 1)
ACMPHSn_NS = 0x5023_6000 + 0x0100 × n (n = 0, 1)

Offset address: 0x004

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CMPSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CMPSEL[3:0]	Comparator Input Selection*1 *2 *3 0x00: Do not input 0x01: Select IVCMP0 0x02: Select IVCMP1 0x04: Select IVCMP2 0x08: Select IVCMP3 Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Use the following procedure to change the CMPSEL[3:0] bits. Writing a value other than 0x00 while the value of the CMPSEL0 register is not 0x00 is invalid. Writing 1 to two or more bits is also invalid. In both cases, the previous value is retained.
To change the CMPSEL[3:0] bits:
1. Set the CMPCTL.COE bit to 0.
2. Set the CMPSEL0 register to 0x00.
3. Set a new value in the CMPSEL[3:0] bits, with 1 set in only one of the bits.
4. Wait for the input switching stabilization wait time (200 ns).
5. Set the CMPCTL.COE bit to 1.
6. Clear the IR flag in the IELSRn register to clear the interrupt status.

Note 2. For details, see Table 48.2.
Note 3. If ACMPHSn level detection signal is used as a POEG source, write access to these bits after the setting of any register in the POEG may generate a POEG source.

48.2.3 CMPSEL1 : Comparator Reference Voltage Select Register

Base address: ACMPHSn = 0x4023_6000 + 0x0100 × n (n = 0, 1)
ACMPHSn_NS = 0x5023_6000 + 0x0100 × n (n = 0, 1)

Offset address: 0x008

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CRVS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CRVS[3:0]	Reference Voltage Selection*1*2*3*4 0x00: Do not input 0x01: Select IVREF0 0x02: Select IVREF1 0x04: Select IVREF2 0x08: Select IVREF3 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Use the following procedure to change the CRVS[3:0] bits. Writing a value other than 0x00 while the value of the CMPSEL1 register is not 0x00 is invalid. Writing 1 to two or more bits is also invalid. In both cases, the previous value is retained.
To change the CRVS[3:0] bits:
1. Set the CMPCTL.COE bit to 0.
2. Set the CMPSEL1 register to 0x00.
3. Set a new value to the CRVS[3:0] bits, with 1 set in only one of the bits.
4. Wait for the input switching stabilization wait time (200 ns)
5. Set the CMPCTL.COE bit to 1.
6. Clear the IR flag in the IELSRn register to clear the interrupt status.

Note 2. For details, see Table 48.2.
Note 3. When the on-chip D/A converter output voltage is used, set the D/A converter to generate comparator C reference voltage before enabling comparator operation (CMPCTL.HCMPON bit = 1). For details on setting the D/A converter, see section 46, 12-Bit D/A Converter (DAC12).
Note 4. If ACMPHSn level detection signal is used as a POEG source, write access to these bits after the setting of any register in the POEG may generate a POEG source.

48.2.4 CMPMON : Comparator Output Monitor Register

Base address: ACMPHSn = 0x4023_6000 + 0x0100 × n (n = 0, 1)
ACMPHSn_NS = 0x5023_6000 + 0x0100 × n (n = 0, 1)

Offset address: 0x00C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	COMPMON
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	COMPMON	Comparator Output Monitor*1 0: Comparator output is low 1: Comparator output is high	R
7:1	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE-3, P-TYPE-3
Note 1. When ACMPHS operation is enabled (CMPCTL.HCMPON = 1 and CMPCTL.COE = 1) but the noise filter is not in use (CDFS[1:0] = 00b), design the software so that the COMPMON bit is read twice and the values are only used after the two consecutive values match.

48.2.5 CPIOC : Comparator Output Control Register

Base address: ACMPHSn = 0x4023_6000 + 0x0100 × n (n = 0, 1)
ACMPHSn_NS = 0x5023_6000 + 0x0100 × n (n = 0, 1)

Offset address: 0x010

Bit position:	7	6	5	4	3	2	1	0
Bit field:	VREFEN	—	—	—	—	—	—	CPOE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPOE	External Pin Output Enable Comparison result by the comparator is output to an external pin. 0: Output to the comparator external pin is disabled (the output signal is fixed to low) 1: Output to the comparator external pin is enabled	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	VREFEN	Internal Vref Enable*1 0: Disable internal Vref 1: Enable internal Vref	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. For ACMPHS modules 0 to 1, VREFEN exists only in ACMPHS0.CPIOC. When using the internal Vref in ACMPHS0 to ACMPHS1, set the VREFEN bit in ACMPHS0.CPIOC to 1. Bit [7] in ACMPHS1.CPIOC should be 0 regardless of whether or not the internal Vref is used.

48.2.6 CPINTCTL : Comparator Interrupt Control Register

Base address: ACMPHSn = 0x4023_6000 + 0x0100 × n (n = 0, 1)
ACMPHSn_NS = 0x5023_6000 + 0x0100 × n (n = 0, 1)

Offset address: 0x040

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MSKE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MSKE	Comparator Interrupt Periodic Mask Enable 0: Disable interrupt masking (Default) 1: Enable interrupt masking by GPT output signal selected by CPMSKCTL.MSKSEL [2:0]	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register controls the interrupt to ICU/ELC, while the event to POEG (Port Output Enable) is not controlled by this register.

48.2.7 CPMSKCTL : Comparator Interrupt Mask Control Register

Base address: ACMPHSn = 0x4023_6000 + 0x0100 × n (n = 0, 1)
ACMPHSn_NS = 0x5023_6000 + 0x0100 × n (n = 0, 1)

Offset address: 0x044

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	MSKSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	MSKSEL[2:0]	Comparator Interrupt Periodic Mask Selection 0 0 0: Enable interrupt masking by GTIOC0A output signal 0 0 1: Enable interrupt masking by GTIOC1A output signal 0 1 0: Enable interrupt masking by GTIOC2A output signal 0 1 1: Enable interrupt masking by GTIOC3A output signal 1 0 0: Enable interrupt masking by GTIOC4A output signal 1 0 1: Enable interrupt masking by GTIOC5A output signal 1 1 0: Enable interrupt masking by GTIOC6A output signal 1 1 1: Enable interrupt masking by GTIOC7A output signal	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register controls the interrupt to ICU/ELC, while the event to POEG (Port Output Enable) is not controlled by this register.

48.3 Operation

The ACMPHS compares a reference voltage to an analog input voltage. Operation is not guaranteed when the values of registers are changed during ACMPHS operation. Table 48.3 shows the procedures for setting the registers associated with ACMPHS.

Table 48.3 Procedure for setting registers associated with ACMPHSn (n = 0, 1) (1 of 2)

Step	Register	Bit	Setting
1	Associated MSTPCRD register	MSTPD28 to MSTPD27	0: Input clock supply.
2	Associated pin function control register (PFS)	ASEL	1: Select the function of pins IVREF and IVCMP.
3	ACMPHSn.CPIOC	VREFEN	1: When using the internal Vref.
4	Associated D/A convertor		When using the D/A convertor, select in the register.
5	CMPSEL0, CMPSEL1	CMPSEL[3:0] CRVS[3:0]	Select the ACMPHSn input, with 1 set in only one of the bits.
6	CMPCTL	CDFS[1:0], CEG[1:0], and CINV	Set up ACMPHSn control.
		HCMPON	1: Enable ACMPHSn operation.
7	CMPSKCTL	MSKCTL[2:0]	Select the interrupt mask source signal (from GPT)

Table 48.3 Procedure for setting registers associated with ACMPHSn (n = 0, 1) (2 of 2)

Step	Register	Bit	Setting
8	CPINT	MSKE	1: Enable interrupt mask function if need
9	Waiting for the ACMPHS stabilization time (minimum 300 ns).		
10	CMPCTL	COE	1: Enable ACMPHSn output.
11	CPIOC	CPOE	Set the VCOUT output
	Associated pin function control register (PFS)	PSEL, PMR	Select the VCOUT port function.
12	IELSRn	IR, IELS[8:0]	When using an interrupt, select the interrupt status flag and the ICU event link.*1
13	ELSRn	ELS[8:0]	When using an ELC, select the event link*2.
14	Operation started		
15	CMPCTL	COE	0: When changing IVREF or IVCMP, to disable ACMPHSn output.
16	CMPSEL1	CRVS[3:0]	Change the CMPSEL1 bits as follows: 1. Set bits CMPSEL1 to 0000 0000b. 2. Set a new value to the CMPSEL1 bits, with 1 set in only one of the bits.
	CMPSEL0	CMPSEL[3:0]	Change the CMPSEL0 bits as follows: 1. Set bits CMPSEL0 to 0000 0000b. 2. Set a new value to the CMPSEL0 bits, with 1 set in only one of the bits.
17	Waiting for the ACMPHS switching stabilization time (minimum 200 ns).		
18	CMPCTL	COE	1: Enable ACMPHSn output.
19	Operation restarted		

Note 1. After ACMPHSn is set, an unnecessary interrupt might occur until operation becomes stable, so initialize the interrupt flag.
Note 2. After ACMPHSn is set, an unnecessary interrupt might occur until operation becomes stable, so initialize the event link select.

Figure 48.2 and Figure 48.3 show an example of ACMPHS operation. The VCOUT output becomes 1 when the analog input voltage is higher than the ACMPHS reference input voltage, and the VCOUT output becomes 0 when the analog input voltage is lower than the reference voltage. When the ACMPHS output changes, an interrupt request and an ELC event are output.

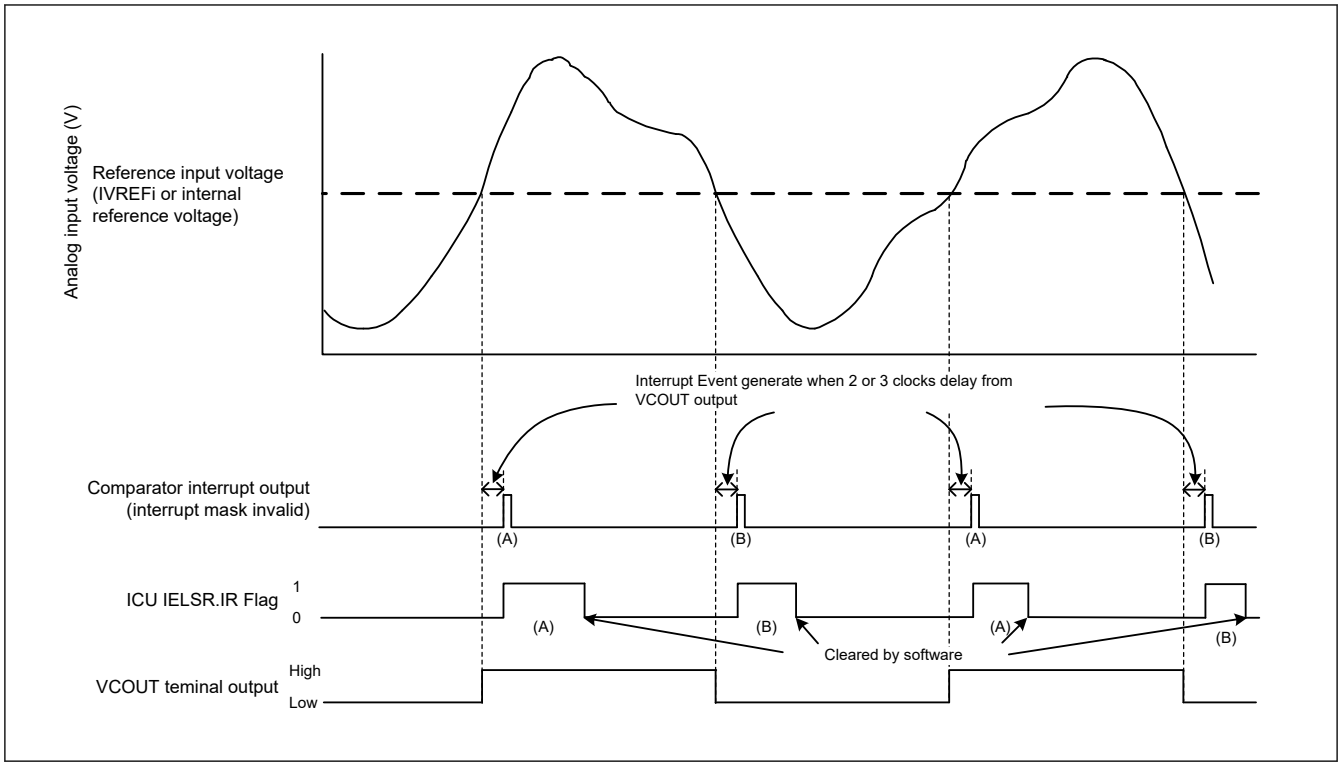


Figure 48.2 ACMPHS operation example (interrupt mask function invalid)

Figure 48.2 applies when CPOE = 1 (pin output enabled), CDFS[1:0] = 00b (filter not used), and CEG[1:0] = 11b (both-edge detection selected). When CINV = 0, CEG[1:0] = 01b (rising-edge detection selected for non-inversion output signal from the ACMPHS), the IELSR.IR flag changes as shown by (A) only. When CINV = 0, CEG[1:0] = 10b (falling-edge detection selected for non-inversion output signal from the ACMPHS), the IR flag changes as shown by (B) only. When CPOE = 1, VCOUT directly outputs.

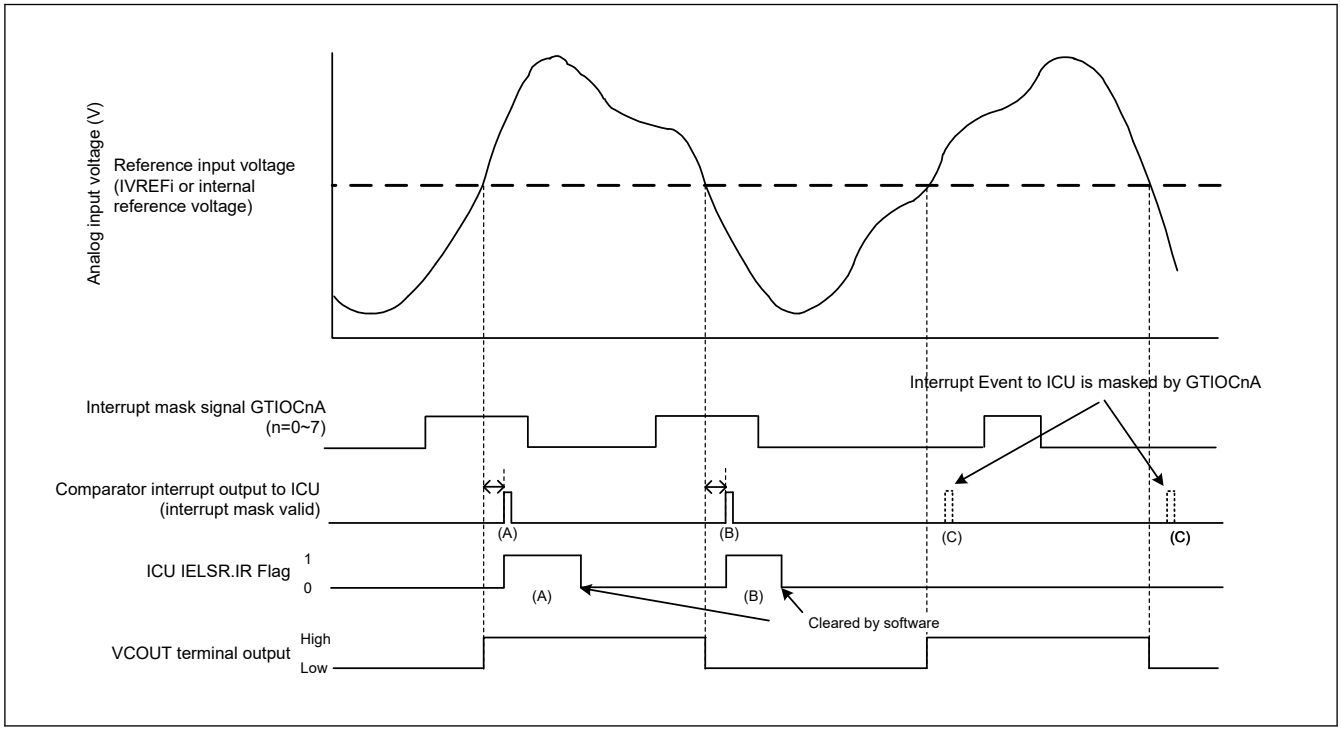


Figure 48.3 ACMPHS operation example (interrupt mask function valid)

Figure 48.3 applies when CPOE = 1 (pin output enabled), CDFS[1:0] = 00b (filter not used), CEG[1:0] = 11b (both-edge detection selected). When CINV = 0, CEG[1:0] = 01b (rising-edge detection selected for non-inversion output signal from

the ACMPHS), the IELSR.IR flag changes as shown by (A) only. When CINV = 0, CEG[1:0] = 10b (falling-edge detection selected for non-inversion output signal from the ACMPHS), the IR flag changes as shown by (B) only.

When MSKE = 1 and MSKSEL[2:0] = 000b, the interrupt signal output to ICU is masked by GTIOC0A when GTIOC0A = low level as shown by (C), while it passes through when GTIOC0A = high level.

When CPOE = 1, VCOUT directly outputs, regardless of the MSKE setting.

48.4 Noise Filter

The ACMPHS contains a noise filter. The sampling clock can be selected in the CMPCTL.CDFS[1:0] bits. The comparator output signal is sampled every sampling clock, and if the same value is sampled three times, the noise filter output at the next sampling clock cycle is used as the ACMPHS output.

Figure 48.4 shows the configuration of the noise filter and edge detector, and Figure 48.5 shows an example of noise filter and interrupt operation.

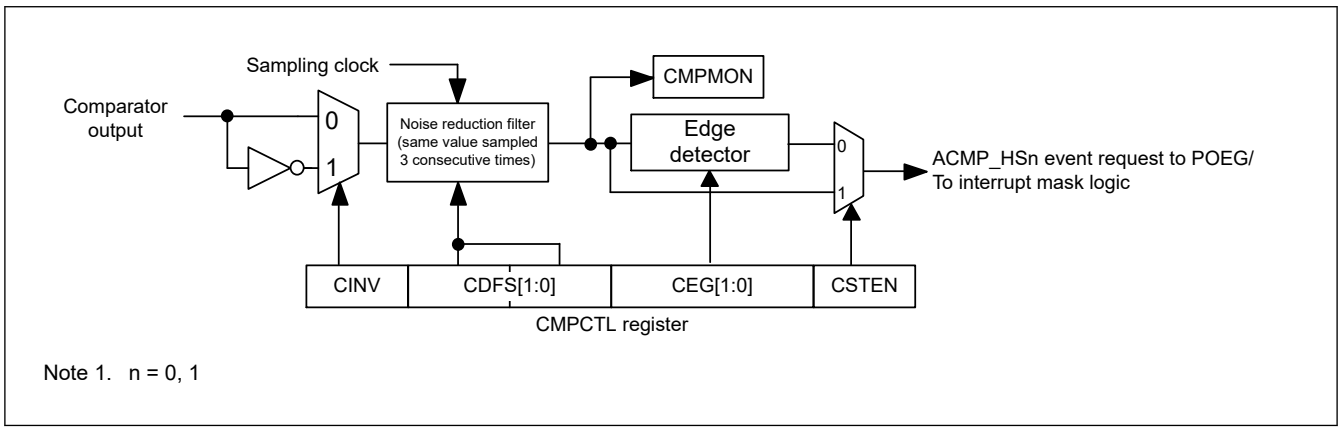


Figure 48.4 Noise filter and edge detection configuration

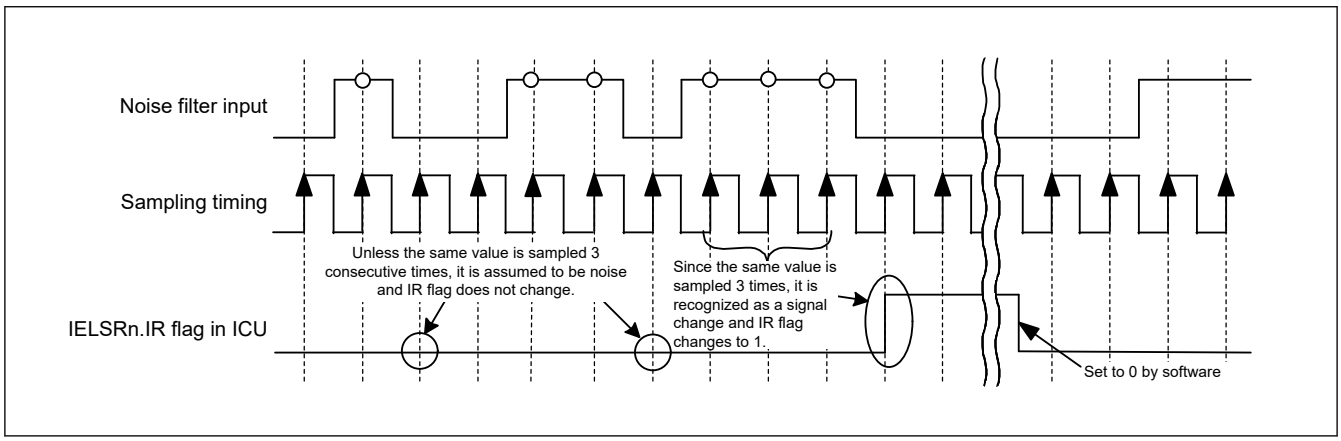


Figure 48.5 Noise filter and interrupt operation example

The operation example in Figure 48.5 applies when the CMPCTL.CDFS[1:0] bits are 01b, 10b, or 11b (noise filter used).

48.5 ACMPHS Interrupts

The ACMPHS generates two interrupt requests from sources ACMPHSn (n = 0, 1). To use an ACMPHS interrupt, select it in the IELSR register in the Interrupt Controller Unit (ICU). Select the interrupt request in the CMPCTL.CSTEN bit either through the edge selector, or not.

Interrupt event to ICU/ELC can be masked by GPT output (GTIOCnA (n = 0 to 7)), it's controlled by CPINTCTL.MSKE bit and CPMSKCTL.MSKSEL[2:0] bits. Detail refer to Figure 48.3.

Event to POEG can't be masked by GPT output (GTIOCnA (n = 0 to 7)), it's not controlled by CPINTCTL.MSKE bit and CPMSKCTL.MSKSEL[2:0] bits.

When using the ACMPHS interrupt through the edge selector, set at least one of the CMPCTL.CEG[1:0] bits to 1 (to a value other than 00b for no edge selection). Set the CMPCTL.CSTEN bit to 0 (output through the edge selector) in Normal mode, CPU Sleep mode, and CPU Deep Sleep mode.

To use the ACMPHS interrupt in Software Standby mode, set the CMPCTL.CSTEN bit to 1 (direct output), set the CMPCTL.CDFS[1:0] bit to 00b (digital noise filter not used), and set CMPCTL.CINV as follows:

- When detecting compare result 0 to 1, set CMPCTL.CINV to 0 (comparator output not inverted)
- When detecting compare result 1 to 0, set CMPCTL.CINV to 1 (comparator output inverted).

An ACMPHS0 interrupt request can be used to release Software Standby mode. ACMPHS1 cannot be used.

ACMPHS0/ACMPHS1 both can not be used in Deep Software Standby mode.

For details on the register settings related to ACMPHS interrupt requests, see section 48.2.1. CMPCTL : Comparator Control Register, section 48.2.6. CPINTCTL : Comparator Interrupt Control Register and section 48.2.7. CPMSKCTL : Comparator Interrupt Mask Control Register.

48.6 ACMPHS Output to the Event Link Controller (ELC)

The ELC uses the ACMPHS interrupt request signal as an ELC event signal, enabling link operation for the preset module. To use the ACMPHS ELC event, select them in the ELSR register in the ELC. When using the ELC event request, set the CMPCTL.CSTEN bit to 0 (output through the edge selector). Also set at least one of the CMPCTL.CEG[1:0] bits to 1 (to a value other than 00b for no edge selection).

Interrupt event to ELC can be masked by GPT output (GTIOCnA (n = 0 to 7)), it's controlled by CPINTCTL.MSKE bit and CPMSKCTL.MSKSEL[2:0] bits.

48.7 ACMPHS Pin Output

The comparison result from the ACMPHS can be output to external pins. Use the CMPCTL.CINV and CPIOC.CPOE bits to set the output polarity (non-inverted or inverted output) and enable or disable output. To output the ACMPHS comparison result to the VCOUT output pin, set the associated port mn pin function control register (PmnPFS) in the I/O register.

Only when the security attribution of Port and ACMPHS match, the ACMPHS comparison result is output to the VCOUT pin as shown in the Table 48.4.

Table 48.4 The output condition of VCOUT pin by the security setting

PORT Security Attribution (PmSAR (m = 0 to 9, A, B))	ACMPHS0 Security Attribution PSARD	ACMPHS1 Security Attribution PSARD	VCOUT pin output
0 (secure)	0 (secure)	0 (secure)	both ACMPHS0/1 output enable
1 (non-secure)	0 (secure)	0 (secure)	both ACMPHS0/1 output disable
0 (secure)	0 (secure)	1 (non-secure)	ACMPHS0_VCOUT output enable ACMPHS1_VCOUT output disable
1 (non-secure)	0 (secure)	1 (non-secure)	ACMPHS0_VCOUT output disable ACMPHS1_VCOUT output enable
0 (secure)	1 (non-secure)	0 (secure)	ACMPHS0_VCOUT output disable ACMPHS1_VCOUT output enable
1 (non-secure)	1 (non-secure)	0 (secure)	ACMPHS0_VCOUT output enable ACMPHS1_VCOUT output disable
0 (secure)	1 (non-secure)	1 (non-secure)	both ACMPHS0/1 output disable
1 (non-secure)	1 (non-secure)	1 (non-secure)	both ACMPHS0/1 output enable

48.8 Usage Notes

48.8.1 Settings for the Module-Stop Function

ACMPHS operation can be disabled or enabled using the Module Stop Control Register. The ACMPHS is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

48.8.2 Settings for the DAC12

ACMPHS is connected to the internal module output of the D/A converter. For details, see section 46, 12-Bit D/A Converter (DAC12).

48.8.3 Relationship with the ADC12

Constraints apply on the simultaneous use of ACMPHS analog input and ADC12 analog input. For details, see section 45.6.14. Relationship between ADC12 Units 0 and 1 and the ACMPHS.

48.8.4 ACMPHS Operation in Module-Stop State

When the module-stop state is entered while ACMPHS is operating, analog circuits in the ACMPHS are not stopped and the analog power supply current is the same as that when ACMPHS is being used. If the analog power supply current needs to be reduced in the module-stop state, set the CMPCTL.HCMPON bit to 0 to stop the ACMPHS.

48.8.5 ACMPHS Operation in Software Standby Mode

When Software Standby mode is entered while ACMPHS is operating, analog circuits in the ACMPHS are not stopped and the analog power supply current is the same as that when ACMPHS is being used. If the analog power supply current needs to be reduced in Software Standby mode, set the CMPCTL.HCMPON bit to 0 to stop the ACMPHS.

48.8.6 Setting the D/A Converter for Generating Reference Voltage

Set the D/A converter to generate reference voltage and wait for the D/A converter output settling time before enabling the comparator. Similarly, before making any changes to the settings of the D/A converter, stop the comparator temporarily. After the changes are made, wait for the D/A converter output settling time before enabling the comparator.

49. Data Operation Circuit (DOC)

This is the DOC_B version of the DOC peripheral module.

DOC_B is referred to as DOC in this chapter.

49.1 Overview

The data operation circuit (DOC) is used to compare, add, and subtract 16 or 32-bit data. An interrupt can be generated when the following conditions apply.

- When the 16 or 32-bit compared values match the detection condition
- When the result of 16 or 32-bit data addition overflows
- When the result of 16 or 32-bit data subtraction underflows

Table 49.1 lists the data operation circuit specifications and Figure 49.1 shows a block diagram of the data operation circuit.

Table 49.1 DOC specifications

Item	Description
Data operation function	• 16 or 32-bit data comparison, comparison to detect data above or below thresholds, and window comparison • 16 or 32-bit data addition, and subtraction
Module-stop function	The module-stop state can be set to reduce power consumption.
Interrupts	• The compared values match the detection condition • The result of data addition is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF_FFFF (DOCR.DOBW = 1) • The result of data subtraction is less than 0x0000 (DOCR.DOBW = 0) or 0x0000_0000 (DOCR.DOBW = 1)
Event link function (output)	• The result of data comparison is consistent with detection condition • The result of data addition is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF_FFFF (DOCR.DOBW = 1) • The result of data subtraction is less than 0x0000 (DOCR.DOBW = 0) or 0x0000_0000 (DOCR.DOBW = 1)
TrustZone Filter	Security and Privilege attribution can be set

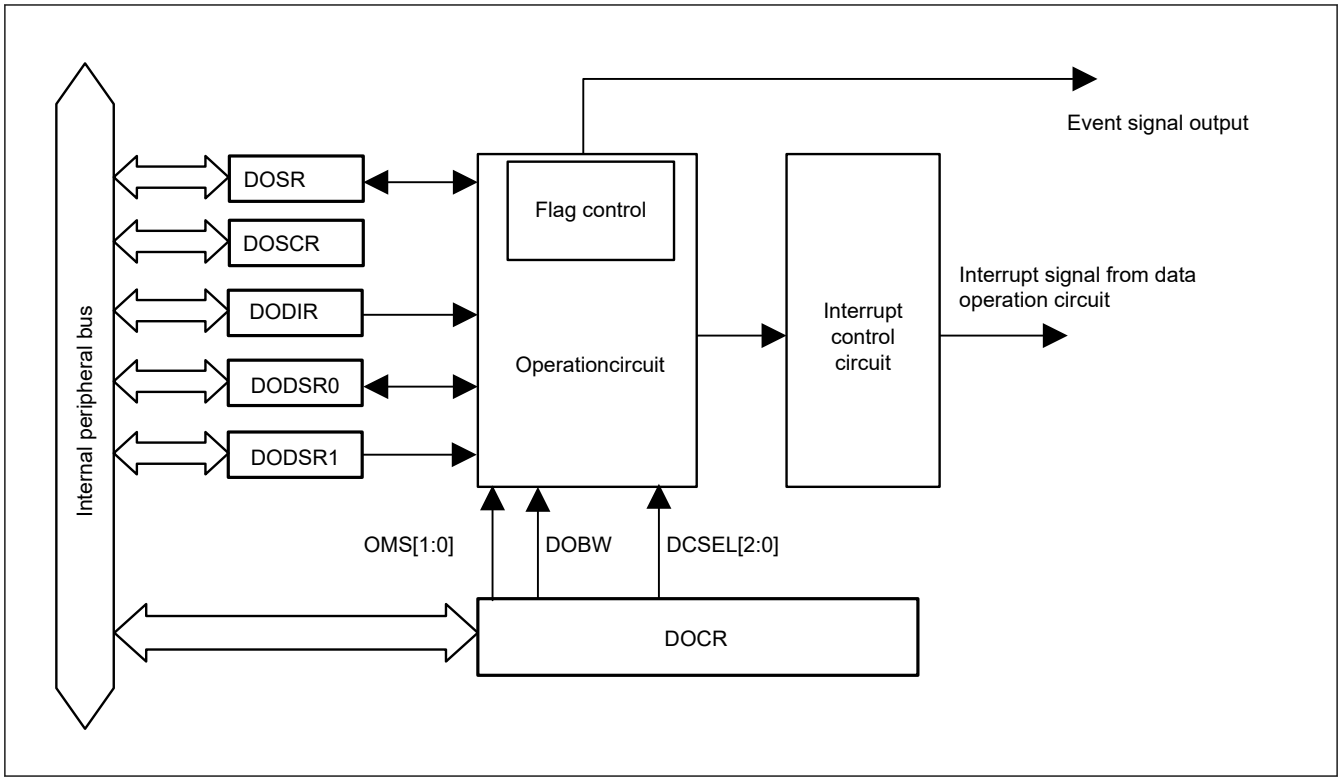


Figure 49.1 DOC block diagram

49.2 DOC Register Descriptions

49.2.1 DOCR : DOC Control Register

Base address: DOC_B = 0x4031_1000
DOC_B_NS = 0x5031_1000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	DCSEL[2:0]			DOBW	—	OMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	OMS[1:0]	Operating Mode Select 0 0: Data comparison mode 0 1: Data addition mode 1 0: Data subtraction mode 1 1: Setting prohibited	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	DOBW	Data Operation Bit Width Select 0: 16-bit 1: 32-bit	R/W
6:4	DCSEL[2:0]*1	Detection Condition Select 0 0 0: Mismatch (DODSR0 ≠ DODIR) 0 0 1: Match (DODSR0 = DODIR) 0 1 0: Lower (DODSR0 > DODIR) 0 1 1: Upper (DODSR0 < DODIR) 1 0 0: Inside window (DODSR0 < DODIR < DODSR1) 1 0 1: Outside window (DODIR < DODSR0, DODSR1 < DODIR) Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. Valid only when data comparison mode is selected.

The DOCR is a register which can set the operation mode of data operation circuit and interrupt enable/disable.

OMS[1:0] bits (Operating Mode Select)

These bits select the operating mode of the data operation circuit.

DOBW bit (Data Operation Bit Width Select)

This bit selects the bit width of data operation.

DCSEL[2:0] bits (Detection Condition Select)

These bits are valid only when data comparison mode is selected.

These bits select the condition for detection in data comparison mode.

49.2.2 DOSR : DOC Flag Status Register

Base address: DOC_B = 0x4031_1000
DOC_B_NS = 0x5031_1000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	DOPCF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DOPCF	Data Operation Circuit Flag Indicates the result of an operation.	R
7:1	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3

The DOSR register indicates the status of the data operation.

DOPCF flag (Data Operation Circuit Flag)

[Setting conditions]

- DOCR.OMS[1:0] bits = 00b (Data comparison mode): The compared value matches the detection condition selected by DOCR.DCSEL[2:0] bits
- DOCR.OMS[1:0] bits = 01b (Data addition mode): The result of data addition is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF_FFFF (DOCR.DOBW = 1)
- DOCR.OMS[1:0] bits = 10b (Data subtraction mode): The result of data subtraction is less than 0x0000 (DOCR.DOBW = 0) or 0x0000_0000 (DOCR.DOBW = 1)

[Clearing condition]

- Writing 1 to the DOSCR.DOPCFCL bit

49.2.3 DOSCR : DOC Flag Status Clear Register

Base address: DOC_B = 0x4031_1000
DOC_B_NS = 0x5031_1000

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	DOPCFCL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DOPCFCL	DOPCF Clear 0: Maintains the DOPCF flag state. 1: Clears the DOPCF flag.	W
7:1	—	The write value should be 0.	W

Note: S-TYPE3, P-TYPE3

The DOSCR is a register which can clear the status of data operation. This register is read as 0x00.

DOPCFCL bit (DOPCF Clear)

Setting this bit to 1 clears the DOPCF flag.

49.2.4 DODIR : DOC Data Input Register

Base address: DOC_B = 0x4031_1000
DOC_B_NS = 0x5031_1000

Offset address: 0x0C

Bit position:	31	0
Bit field:		
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	n/a	It stores data used in the operations. Access the DODIR with the bit width of data operation selected by the DOCR.DOBW bit.	R/W

Note: S-TYPE3, P-TYPE3

49.2.5 DODSR0 : DOC Data Setting Register 0

Base address: DOC_B = 0x4031_1000
DOC_B_NS = 0x5031_1000

Offset address: 0x10

Bit position:	31	0
Bit field:		
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	n/a	Access the DODSR0 with the bit width of data operation selected by the DOCR.DOBW bit. This register stores data for use as a reference in data comparison mode. When selecting window comparison (DOCR.DCSEL[2:0] = 100b, 101b), set a value less than DODSR1 (DODSR1 > DODSR0). This register also stores the results of operations in data addition and data subtraction modes.	R/W

Note: S-TYPE3, P-TYPE3

49.2.6 DODSR1 : DOC Data Setting Register 1

Base address: DOC_B = 0x4031_1000
DOC_B_NS = 0x5031_1000

Offset address: 0x14

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Access the DODSR1 with the bit width of data operation selected by the DOCR.DOBW bit. This register stores data for use as a reference in data comparison mode. When selecting window comparison (DOCR.DCSEL[2:0] = 100b, 101b), set a value greater than DODSR0 (DODSR1 > DODSR0). This register is only used for window comparisons.	R/W

Note: S-TYPE3, P-TYPE3

49.3 Operation

49.3.1 Data Comparison Mode

Figure 49.2 to Figure 49.7 shows an example of the steps involved in data comparison mode operation by the data operation circuit.

The following is an example of operation when the bit width of data operation is 32-bit.

1. Writing 00b to the DOCR.OMS[1:0] bits selects data comparison mode, and setting the DOCR.DCSEL[2:0] to selects detection condition.
2. The 32-bit reference data is set in DODSR0 and DODSR1.*1
3. 32-bit data for comparison is written to DODIR.
4. If a value written to DODIR match the detection condition set by DOCR.DCSEL[2:0], the DOCR.DOPCF flag is set to 1 and an ELC event and a data operation circuit interrupt are generated.

Note: The comparison operation is executed only by writing to the DODIR

Note 1. The DODSR1 register setting is required only when window comparison is selected. Set a value greater than DODSR0 (DODSR1 > DODSR0).

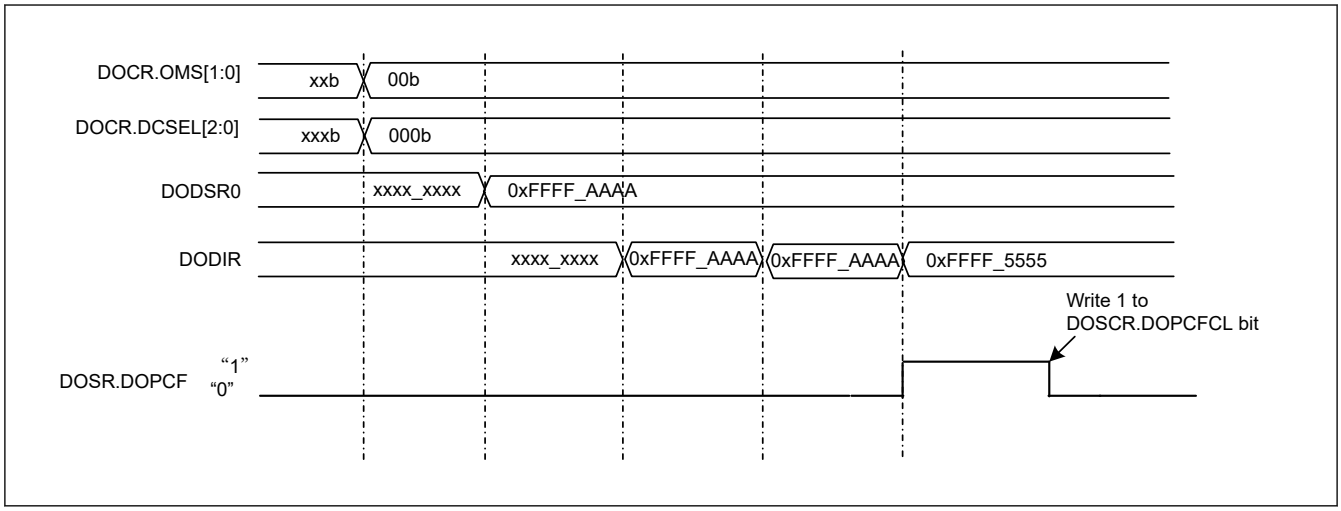


Figure 49.2 Example of Operation in Data Comparison Mode (Detection condition: Mismatch)

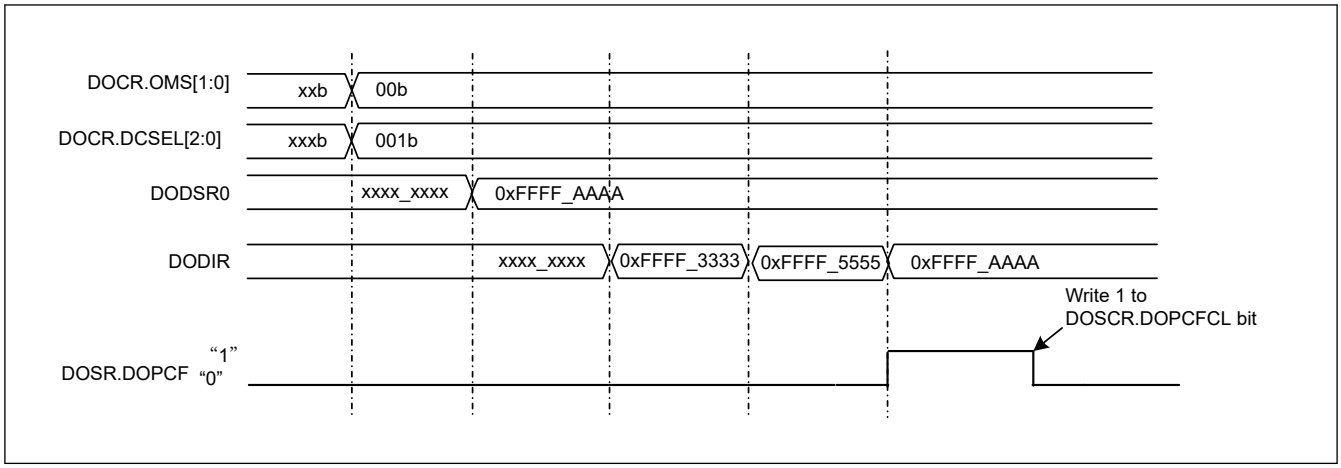


Figure 49.3 Example of Operation in Data Comparison Mode (Detection condition: Match)

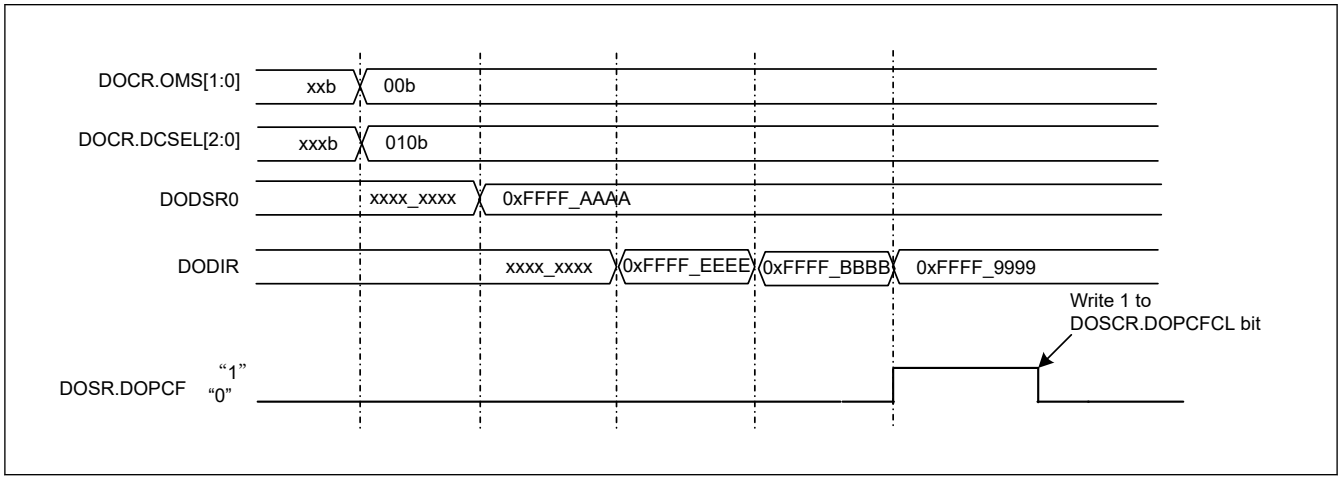


Figure 49.4 Example of Operation in Data Comparison Mode (Detection condition: Lower)

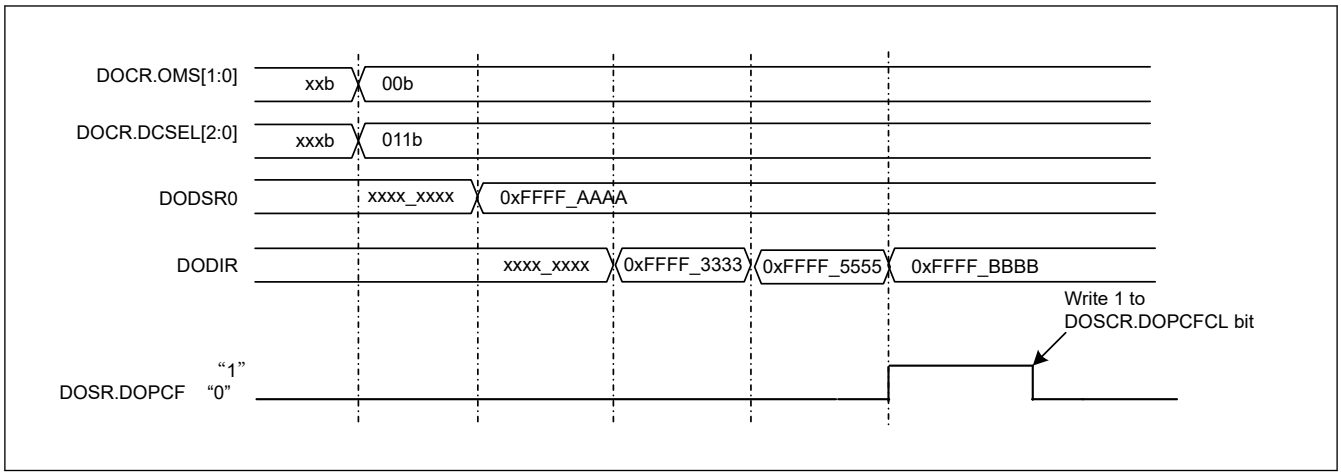


Figure 49.5 Example of Operation in Data Comparison Mode (Detection condition: Upper)

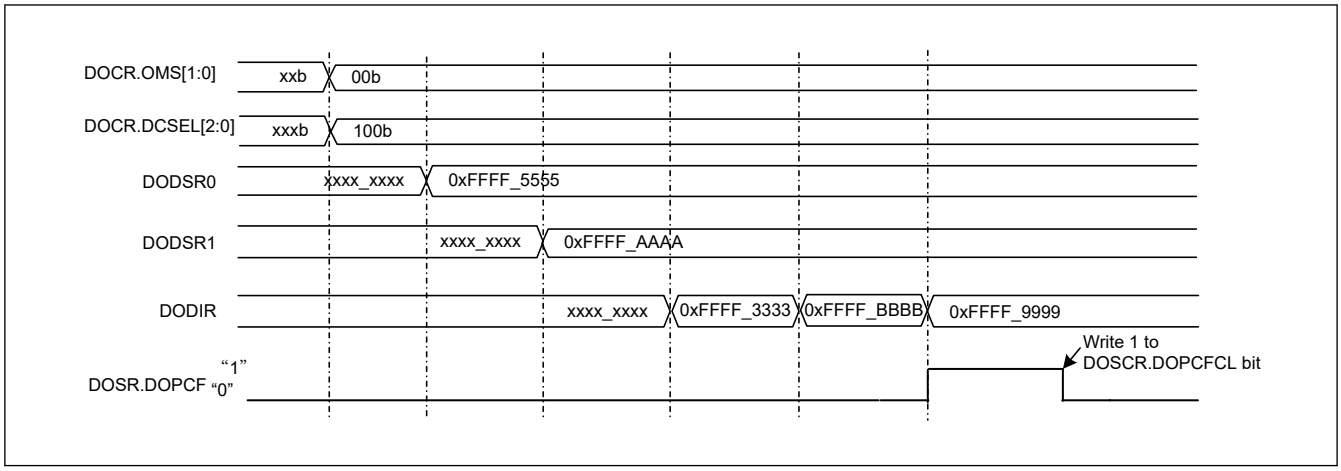


Figure 49.6 Example of Operation in Data Comparison Mode (Detection condition: Inside window)

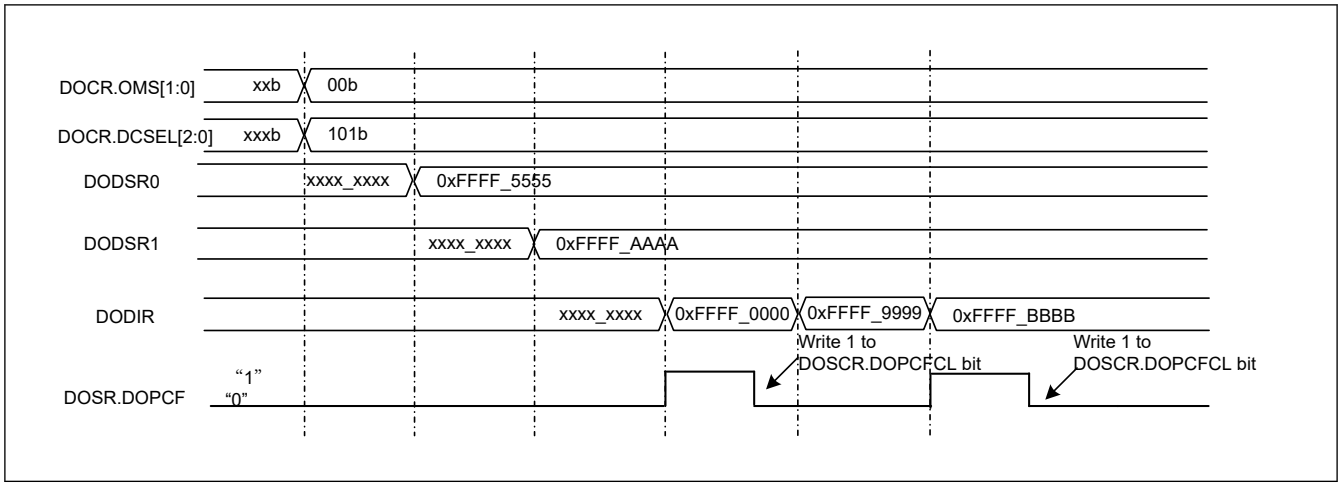


Figure 49.7 Example of Operation in Data Comparison Mode (Detection condition: Outside window)

49.3.2 Data Addition Mode

Figure 49.8 shows an example of the steps involved in data addition mode [*1] operation by the data operation circuit.

The following is an example of operation when the bit width of data operation is 32-bit.

1. Writing 01b to the DOCR.OMS[1:0] bits selects data addition mode.
2. 32-bit data is set in the DODSR0 register as the initial value.
3. 32-bit data to be added is written to DODIR. The result of the operation is stored in DODSR0.
4. Writing of 32-bit data continues until all data for addition have been written to DODIR.
5. If the result of an operation is greater than 0xFFFF_FFFF, the DOSR.DOPCF flag is set to 1 and an ELC event and a data operation circuit interrupt are generated.

Note 1. Addition is executed only by writing to the DODIR.

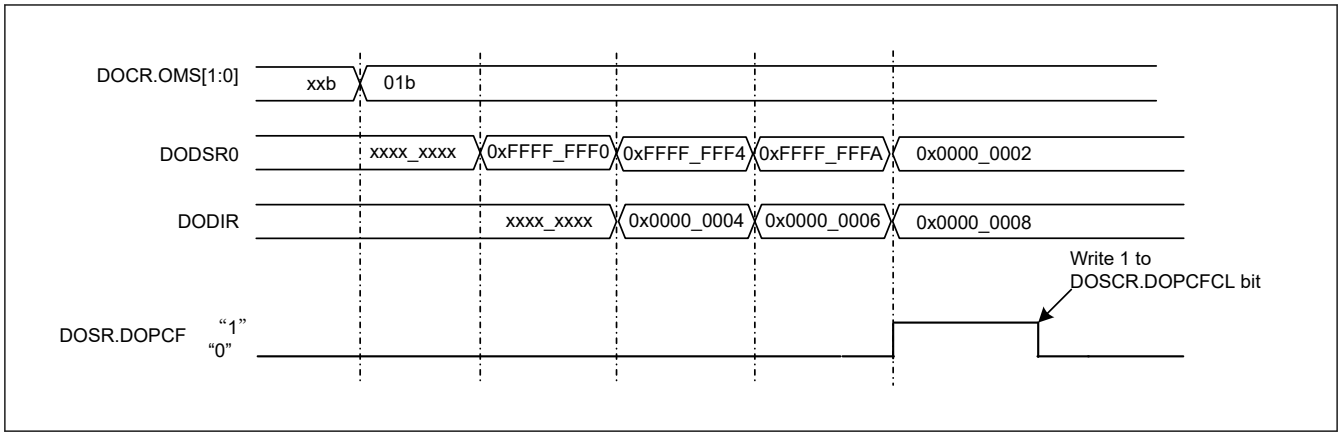


Figure 49.8 Example of Operation in Data Addition Mode

49.3.3 Data Subtraction Mode

Figure 49.9 shows an example of the steps involved in data subtraction mode *1 operation by the data operation circuit.

The following is an example of operation when the bit width of data operation is 32-bit.

1. Writing 10b to the DOCR.OMS[1:0] bits selects data subtraction mode.
2. 32-bit data is set in the DODSR0 register as the initial value.
3. 32-bit data to be subtracted is written to DODIR. The result of the operation is stored in DODSR0.
4. Writing of 32-bit data continues until all data for subtraction have been written to DODIR.
5. If the result of an operation is less than 0x0000_0000, the DOSR.DOPCF flag is set to 1 and an ELC event and a data operation circuit interrupt are generated.

Note 1. Subtraction is executed only by writing to the DODIR.

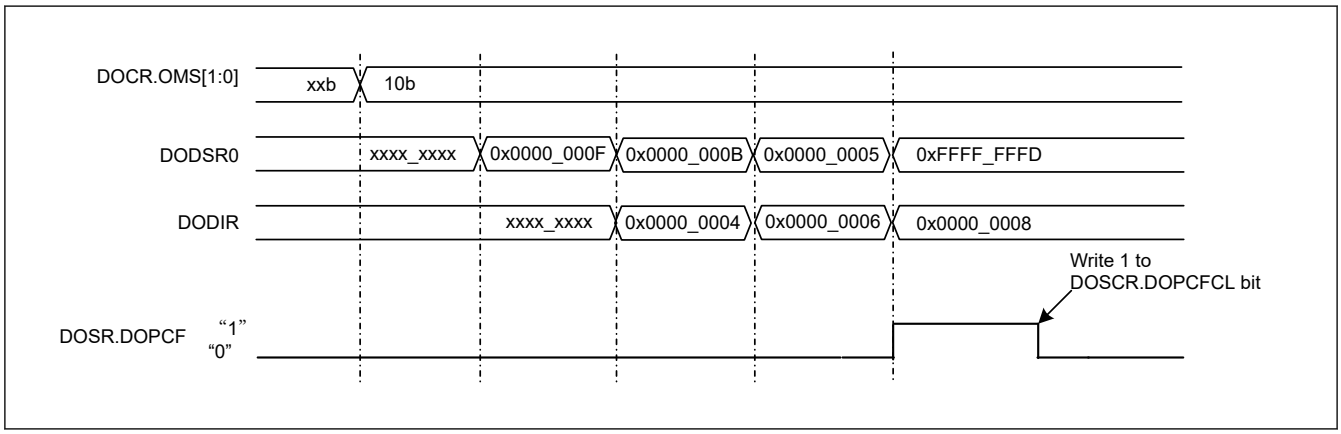


Figure 49.9 Example of Operation in Data Subtraction Mode

49.4 Interrupt Source

The data operation circuit generates the data operation circuit interrupt (DOC_DOPCI) as an interrupt request. When an interrupt source is generated, the data operation circuit flag corresponding to the interrupt is set to 1, and then interrupt request signal is generated. Table 49.2 describes the interrupt request.

Table 49.2 Interrupt request from DOC

Interrupt request	Status flag	Interrupt source
DOC interrupt	DOPCF	• The compared values match the detection condition. • The result of data addition is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF_FFFF (DOCR.DOBW = 1). • The result of data subtraction is less than 0x0000 (DOCR.DOBW = 0) or 0x0000_0000 (DOCR.DOBW = 1).

49.5 Event Link Output

The DOC outputs event signals for the event link controller (ELC) under the following conditions, and these can be used to initiate operations by other modules selected in advance.

- The compared values match the detection condition
- The data addition result is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF_FFFF (DOCR.DOBW = 1)
- The data subtraction result is less than 0x0000 (DOCR.DOBW = 0) or 0x0000_0000 (DOCR.DOBW = 1)

49.6 Interrupt Handling and Event Linking

The DOC has a bit to enable or disable interrupts. An interrupt request signal is output for the CPU when an interrupt source is generated while the corresponding enable bit is enabled.

In contrast, an event link output signal is sent to other modules as an event signal via the ELC when an interrupt source is generated, regardless of the setting of the corresponding interrupt enable bit.

49.7 Usage Notes

49.7.1 Settings for the Module-Stop State

The module Stop Control Register C (MSTPCRC) can enable or disable DOC operation. The DOC is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

50. SRAM

50.1 Overview

The MCU provides an on-chip, high-density SRAM module with either parity-bit checking or Error Correction Code (ECC). SRAM0 is ECC. SRAM1 is Parity check.

Table 50.1 lists the SRAM specifications.

Table 50.1 SRAM specifications

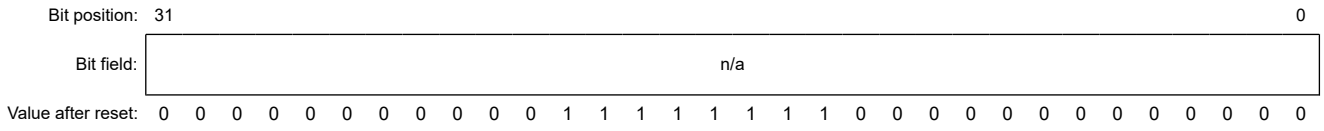
Parameter	SRAM0	SRAM1
SRAM capacity	384 KB	512 KB
SRAM address	0x2200_0000 to 0x2205_FFFF (Secure alias), 0x3200_0000 to 0x3205_FFFF (Non-secure alias)	0x2206_0000 to 0x220D_FFFF (Secure alias), 0x3206_0000 to 0x320D_FFFF (Non-secure alias)
Access	Wait states are inserted into the read cycle by default. If the ICLK frequency is higher than 120 MHz, a wait state is required. If the ICLK frequency is 120 MHz or less, a wait state is not required.	
Data retention function	Not available in Deep Software Standby mode	
Module-stop function	Module-stop state can be set to reduce power consumption	
Error checking	SEC-DED (Single-Error Correction and Double-Error Detection Code)	Even-parity (Data: 8 bits, Parity: 1 bit)
Security	TrustZone Filter is integrated for memory access and SFR access. Access to the memory space is controlled by setting the memory Security Attribution (SA). And access to I/O space (SFR) space is controlled by setting the register SA. See section 50.3.6. TrustZone Filter function.	

50.2 Register Descriptions

50.2.1 SRAMSABARn : SRAM Security Attribute Boundary Address Register (n = 0, 1)

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x400 + 0x04 × n



Bit	Symbol	Function	R/W
31:0	n/a	Boundary address between Secure and Non-secure. (Start address of Non-secure region)	R/W

Note: S-TYPE-1, P-TYPE-1
Note: This register is write-protected by PRCR_S.PRC4 register.

SRAMSABARn specifies the boundary address between the Secure and Non-secure regions of each SRAM. Write the absolute address to the SRAMSABARn. When writing to SRAMSABARn, writing from b31 to b21 is ignored and the value written from b12 to b0 should be 0.

The region lower than the boundary address is marked as Secure, and higher than or equal to the boundary address is marked as Non-secure.

The boundary address is as follow.

0x2200_0000 + SRAMSABARn (Secure alias)

0x3200_0000 + SRAMSABARn (Non-secure alias)

For SRAMSABAR0

When the boundary address is 0x00000000, whole SRAM0 is marked as Non-secure.

When the boundary address is 0x00060000 or large, whole SRAM0 is marked as Secure.

For SRAMSABAR1

When the boundary address is 0x00060000 or less, whole SRAM1 is marked as Non-secure.

When the boundary address is 0x000E0000 or large, whole SRAM1 is marked as Secure.

50.2.2 SRAMSAR : SRAM Security Attribution Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SRAMWTSA	STBRAMSA	—	—	—	—	—	SRAMSA1	SRAMSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SRAMSA0	SRAM0 Register Security Attribution 0: Secure 1: Non-secure	R/W
1	SRAMSA1	SRAM1 Register Security Attribution 0: Secure 1: Non-secure	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	STBRAMSA	Standby SRAM Register Security Attribution 0: Secure 1: Non-secure	R/W
8	SRAMWTSA	SRAMWTSC Security Attribution 0: Secure 1: Non-secure	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1
Note: Only Secure access can write to this register. Both Secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by PRCR_S.PRC4 register.

SRAMSA0 bit (SRAM0 Register Security Attribution)

The SRAMSA0 bit specifies security attributes of SRAMCR0, SRAMECCRGN0, SRAMESCLR.CLR00, and SRAMESCLR.CLR01.

SRAMSA1 bit (SRAM1 Register Security Attribution)

The SRAMSA1 bit specifies security attributes of SRAMCR1 and SRAMESCLR.CLR1.

STBRAMSA bit (Standby SRAM Register Security Attribution)

The STBRAMSA bit specifies security attributes of STBRAMCR and SRAMESCLR.CLRS.

SRAMWTSA bit (SRAMWTSC Security Attribution)

The SRAMWTSA bit specifies security attributes of SRAMWTSC.

50.2.3 SRAMPRCR_S : SRAM Protection Control Register for Secure

Base address: SRAM = 0x4000_2000

Offset address: 0x00

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KW[7:0]								—	—	—	—	—	—	—	PR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PR	Register Write Control 0: Writing to registers are disabled 1: Writing to registers are enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KW[7:0]	Write Key Code Key code protection to the PR bit	R/W

Note: S-TYPE-6, P-TYPE-2

Note: It is necessary to write by half word access.
Byte write access is prohibited. When byte write access is executed, operation is not guaranteed.

PR bit (Register Write Control)

The PR bit controls the write mode of the SRAMWTSC, SRAMCR0, SRAMCR1, SRAMECCRGN0, and STBRAMCR registers that are marked as Secure by SRAMSAR register. When this bit is set to 1, writing to the SRAMWTSC, SRAMCR0, SRAMCR1, SRAMECCRGN0, and STBRAMCR marked as Secure is enabled.

While writing to this register it is necessary to write 0xA5 to the KW[7:0] bits simultaneously.

KW[7:0] bits (Write Key Code)

The KW[7:0] bits enable or disable writing to the PR bit. When you write to the PR bit, write 0xA5 to the KW[7:0] bits simultaneously. When a value other than 0xA5 is written to KW[7:0], the PR bit is not updated. The KW[7:0] bits are always read as 0x00.

50.2.4 SRAMPRCR_NS : SRAM Protection Control Register for Non-secure

Base address: SRAM_NS = 0x5000_2000

Offset address: 0x04

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KW[7:0]								—	—	—	—	—	—	—	PR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PR	Register Write Control 0: Writing to registers are disabled 1: Writing to registers are enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KW[7:0]	Write Key Code Key code protection to the PR bit	R/W

Note: S-TYPE-7, P-TYPE-2

Note: It is necessary to write by half word access.
Byte write access is prohibited. When byte write access is executed, operation is not guaranteed.

PR bit (Register Write Control)

The PR bit controls the write mode of the SRAMWTSC, SRAMCR0, SRAMCR1, SRAMECCRGN0, and STBRAMCR registers that are marked as Non-secure by SRAMSAR register. When this bit is set to 1, writing to the SRAMWTSC, SRAMCR0, SRAMCR1, SRAMECCRGN0, and STBRAMCR marked as Non-secure is enabled.

While writing to this register it is necessary to write 0xA5 to the KW[7:0] bits simultaneously.

KW[7:0] bits (Write Key Code)

The KW[7:0] bits enable or disable writing to the PR bit. When you write to the PR bit, write 0xA5 to the KW[7:0] bits simultaneously. When a value other than 0xA5 is written to KW[7:0], the PR bit is not updated. The KW[7:0] bits are always read as 0x00.

50.2.5 SRAMWTSC : SRAM Wait State Control Register

Base address: SRAM = 0x4000_2000
SRAM_NS = 0x5000_2000

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	WTEN
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	WTEN	SRAM Wait Enable 0: No wait 1: Add wait state in read access cycle to SRAMs	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3 (SRAMSAR.SRAMWTSA), P-TYPE-2

This register can be written only when the PR bit in the SRAMPRCR_S or SRAMPRCR_NS register is 1.

WTEN bit (SRAM Wait Enable)

The WTEN bit sets the wait cycle insertion to the access cycle of SRAM. When 1 is set to the WTEN bit, 1 wait cycle is inserted to the read access cycle of SRAM. Also when the WTEN bit is 1, the access to the same region*1 and the same mat*2 is continuous and 1 wait cycle is inserted to the second access cycle. When the access frequency is more than 120 MHz, it is necessary to set 1 wait cycle to the WTEN bit.

Note 1. The region of SRAM is divided into 128 KB units.
Note 2. The mat is divided into 0x0 to 0x7 and 0x8 to 0xF in the lower 4 bits of the address.

For example, the same region and the same mat is 6 areas of SRAM0, and 8 areas of SRAM1 in the figure below. If 1 area is accessed continuously, 1 wait cycle is inserted. If ECC function is enabled or bypass is enabled, ECC region is accessed, and so if access to the same mat continues, 1 wait cycle is inserted.

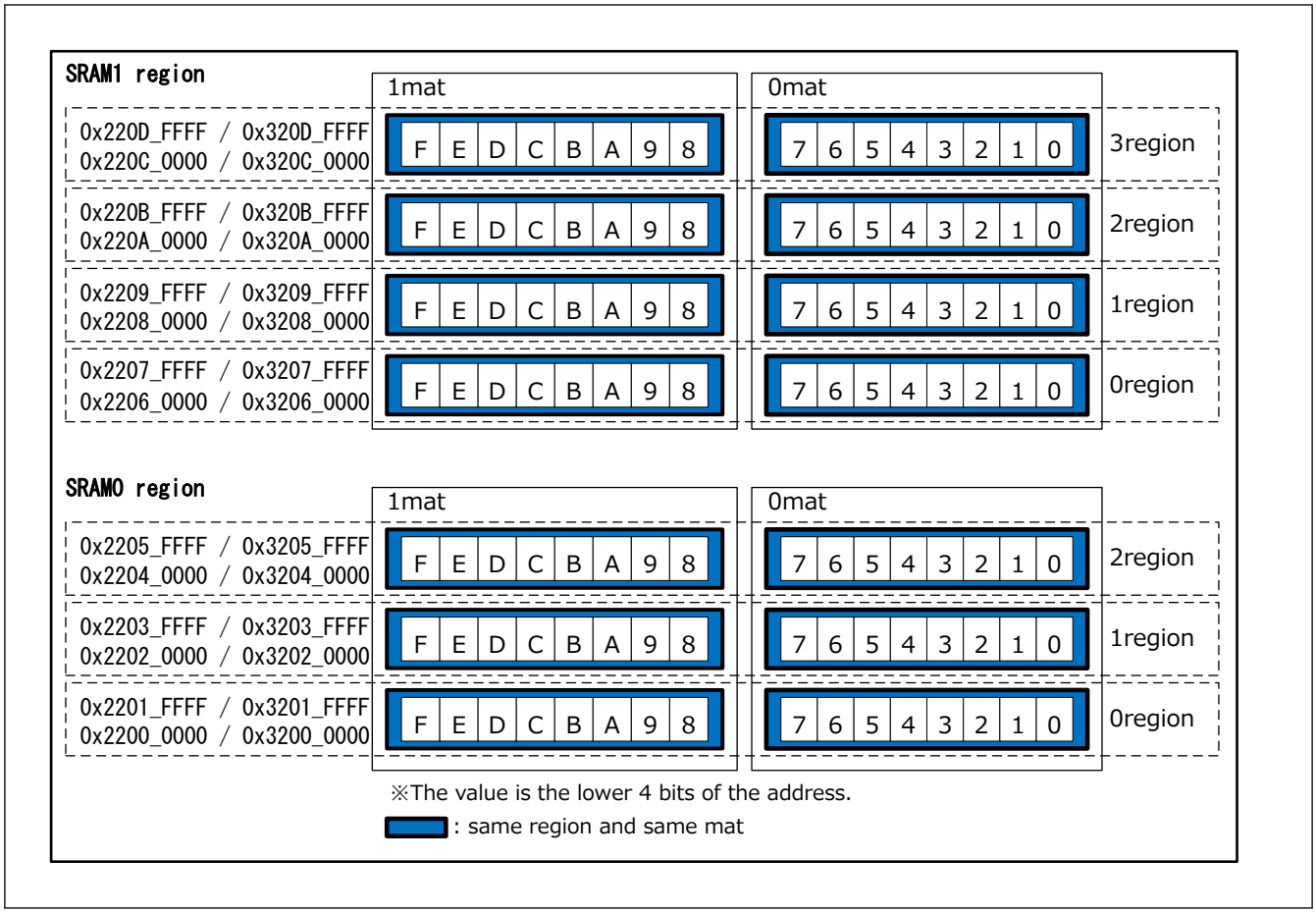


Figure 50.1 Mat configuration of each SRAM

50.2.6 SRAMCR0 : SRAM Control Register 0

Base address: SRAM = 0x4000_2000
SRAM_NS = 0x5000_2000

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TSTBYP	—	—	E1STSEN	ECCMOD[1:0]		—	OAD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OAD	Operation after Detection for ECC Error Detection 0: Non-maskable interrupt 1: Reset.	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
3:2	ECCMOD[1:0]	ECC Operating Mode Select 0 0: Disable ECC function 0 1: Setting prohibited 1 0: Enable ECC function without error checking 1 1: Enable ECC function with error checking	R/W
4	E1STSEN	ECC 1-Bit Error Information Update Enable 0: Disable updating of 1-bit ECC error information 1: Enable updating of 1-bit ECC error information	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	TSTBYP	ECC Bypass Select 0: Disable ECC bypass 1: Enable ECC bypass	R/W

Note: S-TYPE-3 (SRAMSAR.SRAMSA0), P-TYPE-2

This register can be written only when the PR bit in the SRAMPRCR_S or SRAMPRCR_NS register is 1.

Do not write to this register while access to SRAM is in progress. See section 50.4.2. Note of write to SRAMCR0, SRAMCR1 and SRAMECCRGN0 registers for writing to this register.

OAD bit (Operation after Detection for ECC Error Detection)

The OAD bit selects a reset or non-maskable interrupt when a ECC error is detected.

ECCMOD[1:0] bits (ECC Operating Mode Select)

The ECCMOD[1:0] bits set the access mode to the ECC area.

E1STSEN bit (ECC 1-Bit Error Information Update Enable)

The E1STSEN bit enables or disables updating of the ECC 1-bit error status bit in the SRAMESR in response to a 1-bit error. This bit also functions as an interrupt or a reset mask.

TSTBYP bit (ECC Bypass Select)

The TSTBYP bit enables direct access to ECC code by bypassing the ECC function. The ECC Bypass function is used along with setting the ECCMOD[1:0] bits in the same register as 00b. The ECC must be accessed the same address. The ECC code is assigned to the lower 8 bits. When writing the ECC code, the bits above the ECC bits are ignored. When reading the ECC code, the bits above the ECC bits are undefined.

For details of ECC test, see section 50.3.4. ECC Decoder Testing.

50.2.7 SRAMCR1 : SRAM Control Register 1

Base address: SRAM = 0x4000_2000
SRAM_NS = 0x5000_2000

Offset address: 0x14

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OAD	Operation after Detection for Parity Error Detection 0: Non-maskable interrupt 1: Reset.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3 (SRAMSAR.SRAMSA1), P-TYPE-2

This register can be written only when the PR bit in the SRAMPRCR_S or SRAMPRCR_NS register is 1.

Do not write to this register while access to SRAM is in progress. See section 50.4.2. Note of write to SRAMCR0, SRAMCR1 and SRAMECCRGN0 registers for writing to this register.

OAD bit (Operation after Detection for Parity Error Detection)

The OAD bit selects a reset or non-maskable interrupt when a parity error is detected.

50.2.8 SRAMECCRGN0 : SRAM0 ECC Region Control Register

Base address: SRAM = 0x4000_2000
SRAM_NS = 0x5000_2000

Offset address: 0x30

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	ECCRGN[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
1:0	ECCRGN[1:0]	ECC Region Control ECC region selection 0 0: No ECC Region 0 1: 0x2200_0000 – 0x2201_FFFF / 0x3200_0000 – 0x3201_FFFF (128 KB) 1 0: 0x2200_0000 – 0x2203_FFFF / 0x3200_0000 – 0x3203_FFFF (256 KB) 1 1: 0x2200_0000 – 0x2205_FFFF / 0x3200_0000 – 0x3205_FFFF (384 KB) The ECC area selected by this register can be controlled by SRAMCR0.	R/W
7:2	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-3 (SRAMSAR.SRAMSA0), P-TYPE-2

This register can be written only when the PR bit in the SRAMPRCR_S or SRAMPRCR_NS register is 1.

Do not write to this register while access to SRAM is in progress. See section 50.4.2. Note of write to SRAMCR0, SRAMCR1 and SRAMECCRGN0 registers for writing to this register.

ECCRGN[1:0] bits (ECC Region Control)

The ECCRGN[1:0] bits select ECC area in SRAM0.

50.2.9 SRAMESR : SRAM Error Status Register

Base address: SRAM = 0x4000_2000
SRAM_NS = 0x5000_2000

Offset address: 0x40

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ERRS	—	—	—	—	—	—	—	—	—	—	—	ERR1	ERR01	ERR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ERR00	SRAM0 1-bit ECC Error Status 0: 1-bit ECC error has not occurred. 1: 1-bit ECC error has occurred.	R
1	ERR01	SRAM0 2-bit ECC Error Status 0: 2-bit ECC error has not occurred. 1: 2-bit ECC error has occurred.	R
2	ERR1	SRAM1 Parity Error Status 0: Parity error has not occurred. 1: Parity error has occurred.	R
13:3	—	These bits are read as 0.	R
14	ERRS	Standby SRAM Parity Error status 0: Parity error has not occurred. 1: Parity error has occurred.	R
15	—	This bit is read as 0.	R

Note: S-TYPE-5, P-TYPE-2

This register is cleared by the corresponding bit in the SRAMESCLR register or resets other than Bus Error Reset and Memory Error Reset. If error and clear occur at the same time, clear has priority. Also updating register stops during accessing from debugger.

ERR00 bit (SRAM0 1-bit ECC Error Status)

The ERR00 bit shows whether there is a 1-bit ECC error in SRAM0.

When ECC operations are enabled, error correction is selected and updating of the 1-bit error information is enabled. The ERR00 bit is set to 1 if there is a 1-bit error.

When the ERR00 bit is set to 1, a reset or non-maskable interrupt request is generated according to the SRAMCR0.OAD.

ERR01 bit (SRAM0 2-bit ECC Error Status)

The ERR01 bit shows whether there is a 2-bit ECC error in SRAM0.

When ECC operations are enabled, error correction is selected. The ERR01 bit is set to 1 if there is a 2-bit error.

When the ERR01 bit is set to 1, a reset or non-maskable interrupt request is generated according to the SRAMCR0.OAD.

ERR1 bit (SRAM1 Parity Error Status)

The ERR1 bit shows whether there is a parity error or not in the SRAM1.

The ERR1 bit is set if a parity error is detected.

When the ERR1 bit is set to 1, a reset or non-maskable interrupt request is generated according to the SRAMCR1.OAD.

ERRS bit (Standby SRAM Parity Error status)

The ERRS bit shows whether there is a parity error or not in the Standby SRAM.

The ERRS bit is set to 1 if there is an error.

When the ERRS bit is set to 1, a reset or non-maskable interrupt request is generated according to the STBRAMCR.OAD.

50.2.10 SRAMESCLR : SRAM Error Status Clear Register

Base address: SRAM = 0x4000_2000
SRAM_NS = 0x5000_2000

Offset address: 0x48

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CLRS	—	—	—	—	—	—	—	—	—	—	—	CLR1	CLR01	CLR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLR00	SRAM0 1-bit ECC Error Status Clear Writing to the CLR00 clears SRAMESR.ERR00. 1: Clear 1-bit ECC error. Read value is always 0.	R/W*1
1	CLR01	SRAM0 2-bit ECC Error Status Clear Writing to the CLR01 clears SRAMESR.ERR01. 1: Clear 2-bit ECC error. Read value is always 0.	R/W*1
2	CLR1	SRAM1 Parity Error Status Clear Writing to the CLR1 clears SRAMESR.ERR1. 1: Clear Parity error. Read value is always 0.	R/W*1
13:3	—	These bits are read as 0. The write value should be 0.	R/W
14	CLRS	Standby SRAM Parity Error Status Clear Writing to the CLRS clears SRAMESR. ERRS. 1: Clear Parity error. Read value is always 0.	R/W*1

Bit	Symbol	Function	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-4 (SRAMSAR.SRAMSAn,STBRAMSA), P-TYPE-2
Note 1. For each bit, only 1 can be written to clear the error status.

CLR00 bit (SRAM0 1-bit ECC Error Status Clear)

The CLR00 bit can clear the SRAM0 1-bit ECC error status bit in the SRAMESR. If error and clear occur at the same time, clear has priority.

The SRAMESR.ERR00 bit can be cleared when writing 1 to CLR00.

CLR01 bit (SRAM0 2-bit ECC Error Status Clear)

The CLR01 bit can clear the SRAM0 2-bit ECC error status bit in the SRAMESR. If error and clear occur at the same time, clear has priority.

The SRAMESR.ERR01 bit can be cleared when writing 1 to CLR01.

CLR1 bit (SRAM1 Parity Error Status Clear)

The CLR1 bit can clear the SRAM1 parity error status bit in the SRAMESR. If error and clear occur at the same time, clear has priority.

The SRAMESR.ERR1 bit can be cleared when writing 1 to CLR1.

CLRS bits (Standby SRAM Parity Error Status Clear)

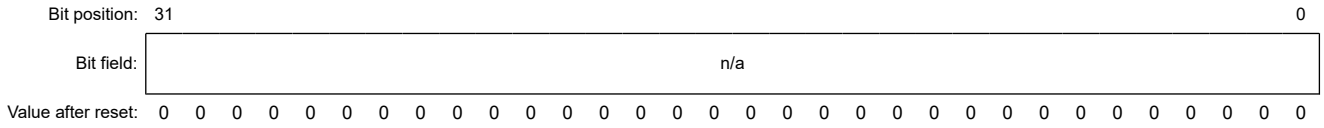
The CLRS bits can clear the Standby SRAM Parity error bit in the SRAMESR. If error and clear occur at the same time, clear has priority.

The SRAMESR.ERRS bit can be cleared when writing 1 to CLRS.

50.2.11 SRAMEARn : SRAM Error Address Register (n = 0 to 2)

Base address: SRAM = 0x4000_2000
SRAM_NS = 0x5000_2000

Offset address: 0x50 + 0x04 × n



Bit	Symbol	Function	R/W
31:0	n/a	When an SRAM error occurs, it stores an error address.	R

Note: S-TYPE-5, P-TYPE-2

This register is cleared by the corresponding bit in the SRAMESCLR register or resets other than Bus Error Reset and Memory Error Reset. If error and clear occur at the same time, clear has priority. Also updating register stops during accessing from debugger.

The error address is as follow.

0x2200_0000 + SRAMEARn (Secure alias)

0x3200_0000 + SRAMEARn (Non-secure alias)

For SRAMEAR0

This register stores the error address where 1-bit ECC error is detected. These bits hold the error address that occurred first. These bits are cleared by clearing 1-bit ECC error from SRAMESCLR.

For SRAMEAR1

This register stores the error address where 2-bit ECC error is detected. These bits hold the error address that occurred first. These bits are cleared by clearing 2-bit ECC error from SRAMESCLR.

For SRAMEAR2

This register stores the error address where parity error is detected. These bits hold the error address that occurred first. These bits are cleared by clearing parity error from SRAMESCLR.

50.3 Operation

50.3.1 Module Stop Function

Power consumption can be reduced by setting module stop control register A (MSTPCRA) to stop supply of the clock signal to SRAM.

SRAMn (n = 0 ,1) is controlled by MSTPAn (n = 0, 1) bit in MSTPCRA register and, in the case of 1, SRAMn becomes the clock stop state.

The SRAM is thus placed in the module-stop state by stopping supply of the clock signals. The SRAM operates after a reset.

SRAM is not accessible if it is in the module-stop state. A transition to the module-stop state should not be made while access to SRAM is in progress.

Access to the SRAM in the module-stop state is prohibited. If access is attempted, correct operation is not guaranteed.

For details on the MSTPCRA register, see section 10, Low Power Modes.

50.3.2 Correction of ECC errors

Enabling and disabling of ECC error correction can be selected through the setting by ECCMOD[1:0] bits of SRAMCR0 register. In the initial state, ECC error correction is disabled. The ECC check type is SEC-DED (Single-Error Correction and Double-Error Detection Code).

When ECC function is enabled, 8-bit check bits are appended to 64-bit data for writing. For reading, 72-bit (data: 64 bits, check bits: 8 bits) data is read out from the SRAM (ECC area) .

When ECC function and error checking are enabled, error correction is done if a 1-bit error occurs and the ERR00 bit in the SRAMESR register is set to 1 if the E1STSEN bit in the SRAMCR0 register is 1. If a 2-bit error occurs, error detection is done and the ERR01 bit in the SRAMESR register is set to 1, though error correction is not performed.

When ECC function is enabled and the error checking is disable, error correction is done if a 1-bit error occurs but ERR00 bit in the SRAMESR register is not updated although E1STSEN bit in the SRAMCR0 register is 1. If a 2-bit error occurs, this error is detected but the ERR01 bit in the SRAMESR register is not updated, and error correction is not performed.

When ECC function is disable, neither error correction nor error detection is done although 1-bit or 2-bit error occur.

So ERR00 and ERR01 bits are not updated.

When updating all the data after the occurrence of an error, the only support of 64 bit data writing.

Since the SRAM data is undefined after power on and release from Deep Software Standby mode, accessing the SRAM when ECC function is enabled and error checking is selected causes an ECC error to occur. Therefore, before using ECC function, initial writing with 64-bit data size, or initial writing with ECC function is enabled and the error checking is disabled, to the area to be used in the SRAM should be done.

When a read access is executed in a row after a write access, read access may be executed with priority. Therefore, during initialization, please do not perform the read access in a row after the write access.

50.3.3 ECC Error Interrupt Function

When ECC function is enabled, error checking is possible in the SRAM0. In the case of a 2-bit error, the ERR01 bit in the SRAMESR register set to 1. In the case of a 1-bit error, the ERR00 bit in the SRAMESR register set to 1 when SRAMCR0.E1STSEN bit is 1.

When the ECC 1-bit error is to be masked, it is necessary to set the SRAMCR0.E1STSEN bit to 0 to disable updating of the ERR00 bit to 1. An ECC error will not be generated while ECC function is disabled or when ECC function is enabled but error checking is not selected using the SRAMCR0.ECCMOD[1:0] bits.

ECC error may result in non-maskable interrupt or reset. When the OAD bit in the SRAMCR0 register is 1, ECC error is outputted to the Reset function. When it is 0, ECC Error interrupt is outputted to the ICU as a non-maskable interrupt.

ECC Error interrupt occurs when one flag in the SRAMESR register is set. ECC Error interrupt continues to occur until the SRAMESR register flag is cleared.

50.3.4 ECC Decoder Testing

ECC decoder testing should be done in the following order.

1. Executes the DMB instruction.
2. Write 0xA501 to the SRAMPRCR_S or SRAMPRCR_NS and enable writes to the SRAM-related registers.
3. Write 0x08 to the SRAMCR0 and enable the ECC function without error checking and disable the ECC bypass.
4. Executes the DMB instruction.
5. Write 8-bytes data to the target address. The 8-bits ECC code is automatically updated.
6. Executes the DMB instruction.
7. Write 0x80 to the SRAMCR0 and enable the ECC bypass.
8. Executes the DMB instruction.
9. Read target address to get 8-bits-ECC code.
10. To generate 1-bit/2-bit ECC error, reverse 1-bit/2-bit of the data read in the previous process and write the data back to the target address.
11. Executes the DMB instruction.
12. Write 0x1C to the SRAMCR0, disable the ECC bypass and enable the updating of 1-bit ECC error information and enable the ECC function with error checking.
13. Executes the DMB instruction.
14. Read the target address.
15. Executes the DMB instruction.
16. Confirm the generation of the ECC error by the SRAMESR.

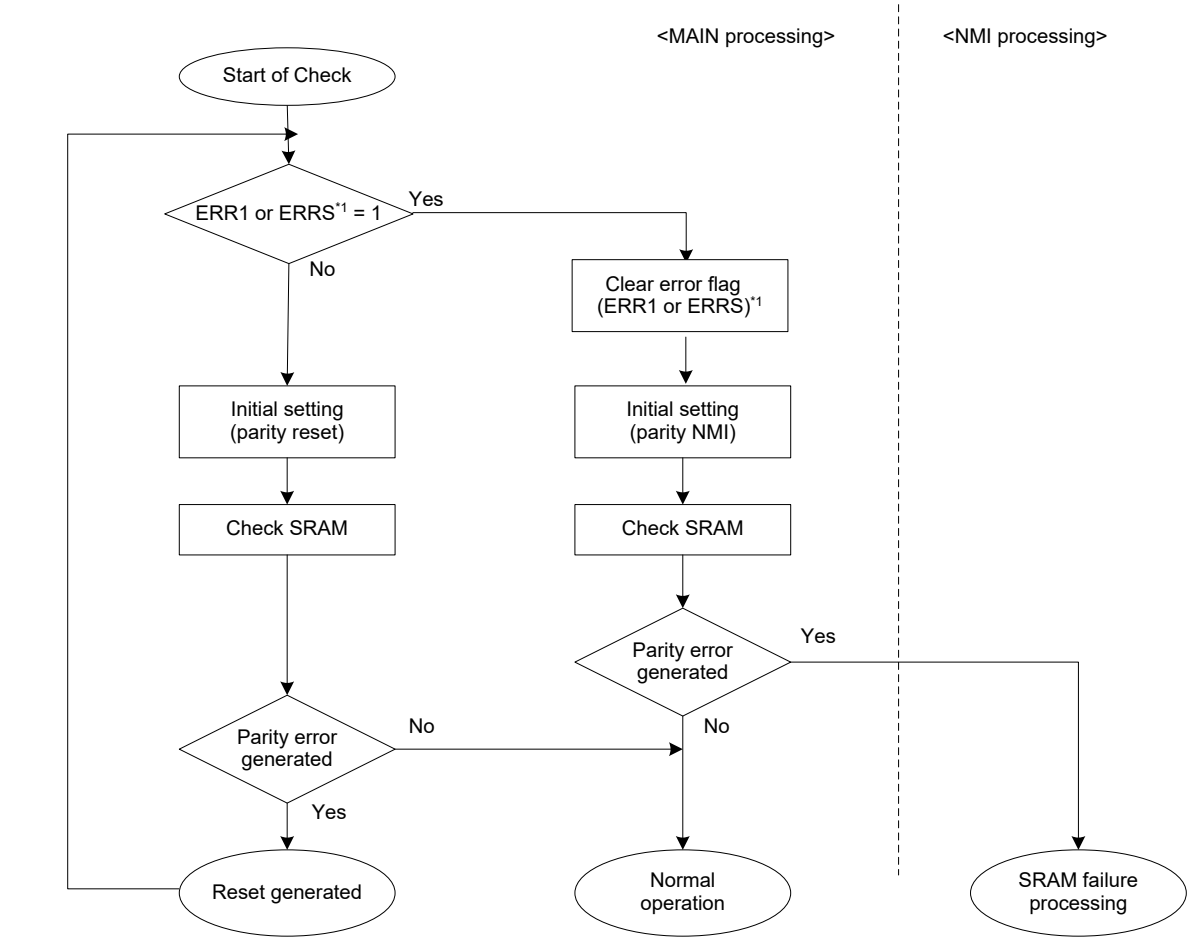
50.3.5 Parity Calculation Function

The IEC60730 standard requires the checking of SRAM data. When data is written, a parity bit is added to every 8-bit data in the SRAM, and when data is read, the parity is checked. When a parity error occurs, a parity-error notification is generated. This function can also be used to trigger a reset.

The parity-error notification can be specified as a non-maskable interrupt or a reset in the OAD bit of the SRAMCR1 register. When the OAD bit is set to 1, a parity error is output to the reset function. When the OAD bit is set to 0, a parity error is output to the ICU as a non-maskable interrupt.

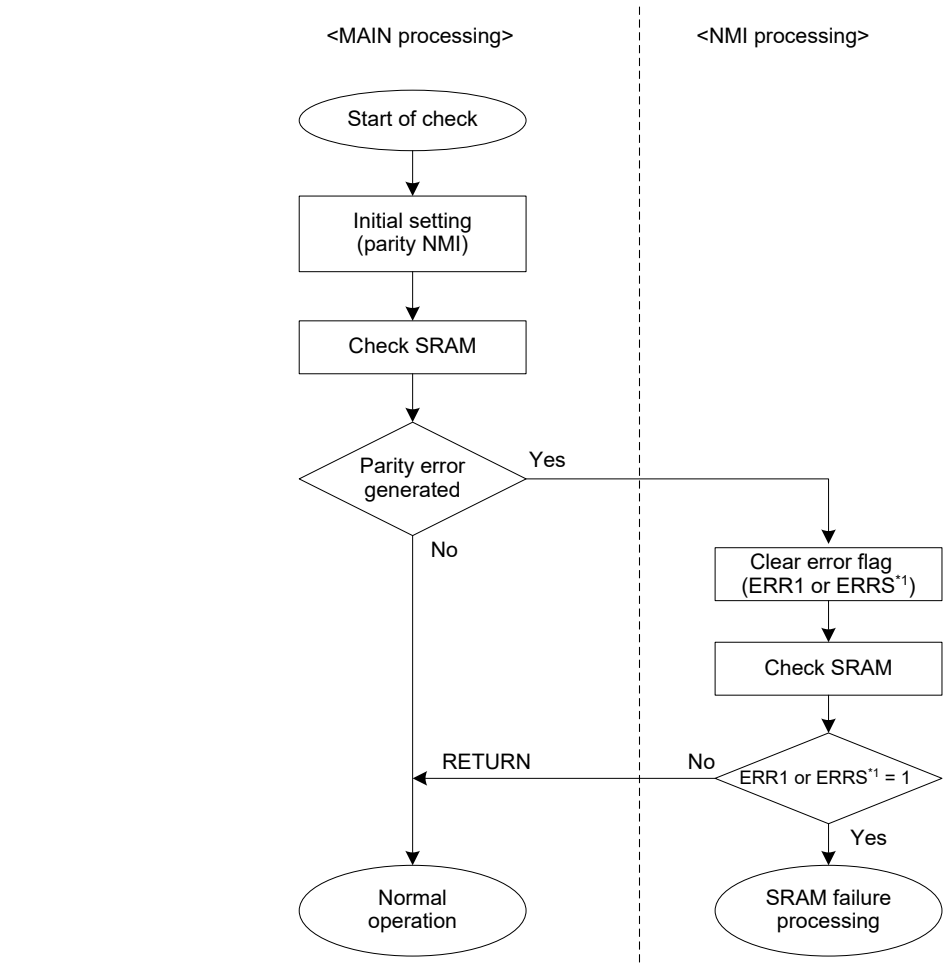
Parity Error often occurs due to noise. It is possible to confirm whether the parity error is caused by noise or destruction of data by following the parity operation flowchart. Figure 50.2 and Figure 50.3 shows flowcharts of SRAM parity check operation..

When a read access is executed in a row after a write access, read access is executed with priority. Therefore, during initialization, do not perform the read access in a row after the write access.



Note 1. ERR1 or ERRS: SRAM error status (SRAMESR.ERR1, ERRS bit)

Figure 50.2 Flow of SRAM parity check when SRAM parity reset is enabled



Note 1. ERR1 or ERRS: SRAM error status (SRAMESR.ERR1, ERRS bit)

Figure 50.3 Flow of SRAM parity check when SRAM parity interrupt is enabled

50.3.6 TrustZone Filter function

50.3.6.1 TrustZone Filter for SRAM Registers

SRAM registers can be protected with a Security Attribution (SA) from Non-secure or Secure transaction. When SA indicates that SRAM registers are Secure status, Non-secure transaction cannot access them because TrustZone Filter detects an error and protects the access. When SA indicates that SRAM registers are Non-secure status, Secure transaction cannot access them because Trust Zone Filter detects an error and protects the access. SA for SRAM registers is just one to be used commonly among SRAM registers.

In the case of a failed access due to TrustZone error, error response is generated.

Table 50.2 Register protection (1 of 2)

SA	Transaction	Write access	Read access
Secure	Secure	Permitted	Permitted
	Non-secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)

Table 50.2 Register protection (2 of 2)

SA	Transaction	Write access	Read access
Non-secure	Secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
	Non-secure	Permitted	Permitted

50.3.6.2 TrustZone Filter for SRAM Memory Regions

SRAM0, SRAM1 regions can be divided into Secure and Non-secure independently. The access permissions for these regions are as below.

Table 50.3 Memory protection

SA	Transaction	Write access	Read access
Secure	Secure	Permitted	Permit
	Non-secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
Non-secure	Secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
	Non-secure	Permitted	Permitted

TrustZone filter error for SRAM memory generates an error notification.

50.3.7 Interrupt Source

The SRAM interrupt source includes an ECC error, Parity error and TrustZone filter error. ECC error and Parity error can choose non-maskable interrupt or reset by SRAMCR0.OAD or SRAMCR1.OAD bit. The SRAM interrupt occurs when one error status in the SRAMESR register is set to 1, the SRAM interrupt continues to occur until the SRAMESR register flag is cleared. When Common memory error occur (NMISR.CMST=1 or RSTSR1.CMSR=1), please read SRAMESR and check SRAM interrupt source. When the access from debugger, the error is detected and corrected, but no error flag is set, reset and non-maskable interrupt are maskable. For details on the debug mode, see section 2, CPU.

Table 50.4 SRAM Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
ECCERR	ECC error (SRAMs with ECC)	Not possible	Not possible
PARITYERR	Parity error (SRAMs with parity)	Not possible	Not possible
TZFLT	TrustZone filter error	Not possible	Not possible

50.3.8 Wait state

When ICLK frequencies is higher than 120 MHz, do not set 0x00 in wait enable bit for the SRAMWTSC register, in order to insert a wait cycle. When the wait is not inserted, the operation is not guaranteed.

Depending on the operating frequency of ICLK, the WAIT setting for SRAM access has the following conditions.

[ICLK frequency]:

- 240 MHz ≥ ICLK > 120 MHz = 1 wait
- 120 MHz ≥ ICLK = No-wait

50.4 Usage Note

50.4.1 Instruction fetch from SRAM area

When using SRAM to operate a program, initialize the SRAM area so that the CPU can correctly prefetch data. A parity or ECC error might occur if the CPU prefetches from an area that is not initialized. Initialize the additional 12-byte area from the end address of the program with the 8-byte boundary. Renesas recommends using the NOP instruction for data initialization.

50.4.2 Note of write to SRAMCR0, SRAMCR1 and SRAMECCRGN0 registers

This chapter provides examples of how to configure SRAMCR0, SRAMCR1 and SRAMECCRGN0.

All of the followings have to be satisfied:

1. CPU does not execute program stored in SRAM
2. All bus masters other than CPU does not access SRAM
3. SRAMCR0, SRAMCR1 and SRAMECCRGN0 are Device-nGnRnE set by MPU
4. SRAMCR0, SRAMCR1 and SRAMECCRGN0 have to be configured with the following procedure:
   1. DMB instruction
   2. Write to SRAMCR0, SRAMCR1 and SRAMECCRGN0 registers
   3. DMB instruction

50.4.3 Notes on Using Error Checking of SRAM

Data in SRAM are undefined when the power is turned on. Therefore, parity check errors or ECC errors occur if the data are read before initialization. The SRAM are read in 8-byte (64-bit) units. Initialize them on 8-byte boundaries.

51. Standby SRAM

51.1 Overview

An on-chip SRAM is provided to retain data in Deep Software Standby mode. Table 51.1 lists the Standby SRAM specifications.

Table 51.1 Standby SRAM specifications

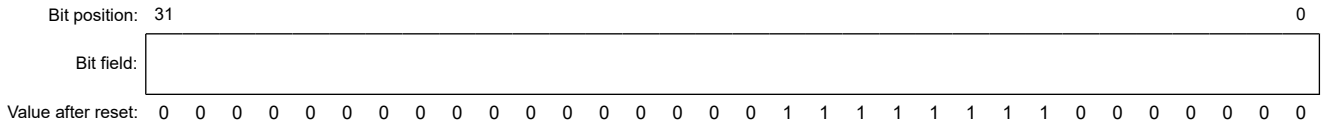
Item	Description
SRAM capacity	1 KB
SRAM address	0x2600_0000 to 0x2600_03FF (Secure alias), 0x3600_0000 to 0x3600_03FF (Non-secure alias)
Access	Wait states are inserted into the access cycle by default. If the ICLK frequency is higher than 120 MHz, a wait state is required. If the ICLK frequency is 120 MHz or less, a wait state is not required.
Data retention function	Data can be retained in Deep Software Standby mode 1. In Deep Software Standby mode 2 and 3, data cannot be retained. See section 51.3.1. Data Retention for details.
parity	Even parity (data: 8 bits, parity: 1 bit)
Module-stop function	Module-stop state can be set to reduce power consumption. See section 51.3.2. Module-stop Function for details.
Security	Permits the read and write operations to Standby RAM following TrustZone Filter function. See section 51.3.4. TrustZone Filter function for details.

51.2 Register Descriptions

51.2.1 STBRAMSABAR : Standby SRAM Security Attribute Boundary Address Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x420



Bit	Symbol	Function	R/W
31:0	n/a	Boundary address between Secure and Non-secure.	R/W

Note: S-TYPE-1, P-TYPE-1
Note: This register is write-protected by PRCR_S.PRC4 register.

STBRAMSABAR specifies the boundary address between the Secure and Non-secure regions of Standby SRAM. Write the absolute address to the STBRAMSABAR. When writing to STBRAMSABAR, writing from b31 to b15 is ignored and the value written from b6 to b0 should be 0.

The region lower than the boundary address is marked as Secure, and higher than or equal to the boundary address is marked as Non-secure. Therefore, when the boundary address is 0x00000000, whole Standby SRAM is marked as Non-secure, and when the boundary address is 0x00000400 or large, whole Standby SRAM is marked as Secure.

The boundary address is as follow.

- 0x2600_0000 + STBRAMSABAR (Secure alias)
- 0x3600_0000 + STBRAMSABAR (Non-secure alias)

51.2.2 STBRAMPABAR_NS : Standby SRAM Privilege Attribute Boundary Address Register for Non-secure

Base address: CPSCU_NS = 0x5000_8000

Offset address: 0x490

Bit position:	31																												0
Bit field:	n/a																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Boundary address between Privileged and Unprivileged.	R/W

Note: S-TYPE-7, P-TYPE-2
Note: This register is write-protected by PRCR_NS.PRC4 register.

STBRAMPABAR_NS specifies the boundary address between the Privileged and Unprivileged regions of Standby SRAM Non-secure regions set by STBRAMSABAR. Write the absolute address to the STBRAMPABAR_NS. When writing to STBRAMPABAR_NS, writing from b31 to b15 is ignored and the value written from b6 to b0 should be 0.

The region lower than the boundary address is marked as Privileged, and higher than or equal to the boundary address is marked as Unprivileged. Therefore, when the boundary address is 0x00000000, or less than or equal to start address of Standby SRAM Non-secure regions set by STBRAMSABAR, whole Standby SRAM is marked as Unprivileged, and when the boundary address is 0x00000400 or large, or higher than or equal to end address of Standby SRAM Non-secure regions set by STBRAMSABAR, whole Standby SRAM is marked as Privileged.

The boundary address is as follow.

- 0x3600_0000 + STBRAMPABAR_NS (Non-secure alias)

51.2.3 STBRAMPABAR_S : Standby SRAM Privilege Attribute Boundary Address Register for Secure

Base address: CPSCU = 0x4000_8000

Offset address: 0x494

Bit position:	31																												0
Bit field:	n/a																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Boundary address between Privileged and Unprivileged.	R/W

Note: S-TYPE-6, P-TYPE-2
Note: This register is write-protected by PRCR_S.PRC4 register.

STBRAMPABAR_S specifies the boundary address between the Privileged and Unprivileged regions of Standby SRAM Secure regions set by STBRAMSABAR. Write the absolute address to the STBRAMPABAR_S. When writing to STBRAMPABAR_S, writing from b31 to b15 is ignored and the value written from b6 to b0 should be 0.

The region lower than the boundary address is marked as Privileged, and higher than or equal to the boundary address is marked as Unprivileged. Therefore, when the boundary address is 0x00000000, or less than or equal to start address of Standby SRAM Secure regions set by STBRAMSABAR, whole Standby SRAM is marked as Unprivileged, and when the boundary address is 0x00000400 or large, or higher than or equal to end address of Standby SRAM Secure regions set by STBRAMSABAR, whole Standby SRAM is marked as Privileged.

The boundary address is as follow.

- 0x2600_0000 + STBRAMPABAR_S (Secure alias)

51.2.4 STBRAMCR : Standby SRAM Control Register

Base address: SRAM = 0x4000_2000
SRAM_NS = 0x5000_2000

Offset address: 0x110

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OAD	Operation after detection 0: Non-maskable interrupt. 1: Reset.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3 (SRAMSAR.STBRAMSA), P-TYPE-2

This register can be written only when the PR bit in the SRAMPRCR_S or SRAMPRCR_NS register is 1.

Do not write to this register while access to Standby SRAM is in progress.

OAD bit (Operation after detection)

This bit is specified to generate either reset or non-maskable interrupt when parity error is detected.

51.2.5 STBRAMEAR : Standby SRAM Error Address Register

Base address: SRAM = 0x4000_2000
SRAM_NS = 0x5000_2000

Offset address: 0x150

Bit position:	31	0
Bit field:	n/a	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	n/a	When an SRAM error occurs, it stores an error address.	R

Note: S-TYPE-5, P-TYPE-2

This register is cleared by the corresponding bit in SRAMESCLR register, or resets other than Bus Error Reset and Memory Error Reset. If error and clear occur at the same time, clear has priority. Also during accessing from debugger it is to stop updating register.

The error address is as follow.

- 0x2600_0000 + STBRAMEAR (Secure alias)
- 0x3600_0000 + STBRAMEAR (Non-secure alias)

This register stores the error address where parity error is detected. These bits hold the error address that occurred first. These bits are cleared by clearing parity error from SRAMESCLR.

51.3 Operation

51.3.1 Data Retention

The power supply to the Standby SRAM in Deep Software Standby mode is enabled by the DPSBYCR.SRKEEP bit. If the DPSBYCR.SRKEEP bit are set to 1b, data in the Standby SRAM is retained in Deep Software Standby mode 1. For details on the DPSBYCR.SRKEEP bit, see section 10, Low Power Modes.

51.3.2 Module-stop Function

Power consumption can be reduced by setting module stop control register A (MSTPCRA) to stop supply of the clock signal to SRAM.

If the Standby SRAM bit in MSTPCRA is set to 1, supply of the clock signal to the Standby SRAM is stopped.

The Standby SRAM is thus placed in the module-stop state by stopping supply of the clock signals. The Standby SRAM operates after a reset.

The Standby SRAM is not accessible if it is in the module-stop state. A transition to the module-stop state should not be made while access to the standby SRAM is in progress.

Access to the Standby SRAM in the module-stop state is prohibited. If access is attempted, correct operation is not guaranteed.

For details on the MSTPCRA register, see section 10, Low Power Modes.

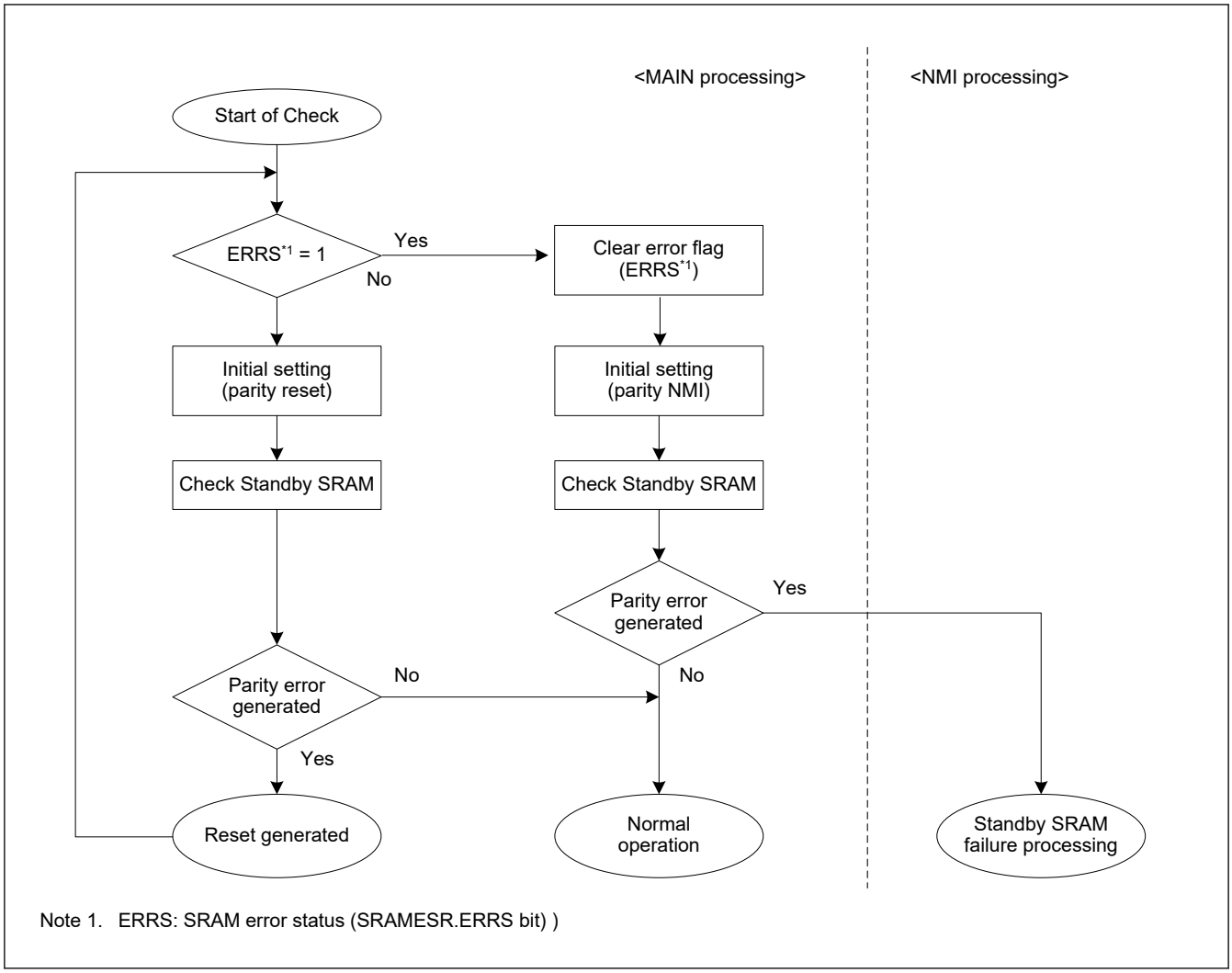
51.3.3 Parity Calculation Function

The IEC60730 standard requires the checking of STBRAMCR data. When data is written, a parity bit is added to every 8-bit data in the Standby SRAM which has 32-bit data width, and when data is read, the parity is checked. When a parity error occurs, a parity error notification is generated. This function can also be used to trigger a reset.

The parity-error notification can be specified as a non-maskable interrupt or a reset in the OAD bit of the STBRAMCR register. When the OAD bit is set to 1, a parity error is output to the reset function. When the OAD bit is set to 0, a parity error is output to the ICU as a non-maskable interrupt.

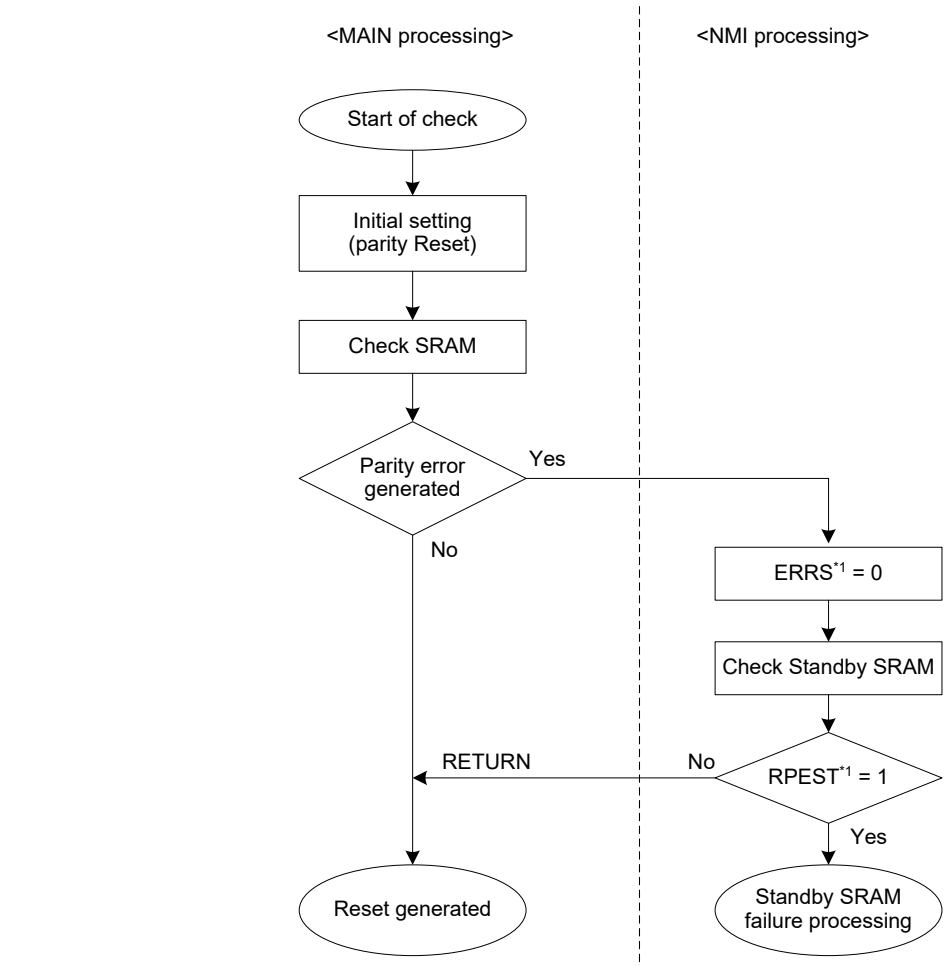
Parity errors often occur because of noise. To confirm whether the cause of the parity error is noise or corruption, follow the parity check flows shown in Figure 51.1 and Figure 51.2.

When a read access is executed in a row after a write access, read access is executed with priority. Therefore, during initialization, do not perform the read access in a row after the write access.



Note 1. ERRS: SRAM error status (SRAMESR.ERRS bit))

Figure 51.1 Flow of Standby SRAM parity check when SRAM parity reset is enabled



Note 1. ERRS: SRAM error status (SRAMESR.ERRS bit)

Figure 51.2 Flow of Standby SRAM parity check when SRAM parity interrupt is enabled

51.3.4 TrustZone Filter function

51.3.4.1 TrustZone Filter for Standby SRAM registers

Standby SRAM registers can be protected with a Security Attribution (SA) from Non-secure or Secure transaction. When SA indicates that Standby SRAM registers are Secure status, Non-secure transaction can not access them because TrustZone Filter detects finds an error and protects the access. When SA indicates that SRAM registers are Non-secure status, Secure transaction can not access them because Trust Zone Filter detects finds an error and protects the access. SA for Standby SRAM registers is just one to be used commonly among Standby SRAM registers.

In the case of a failed access due to TrustZone error, error response is generated.

Table 51.2 Standby SRAM register protection

SA	Transaction	Write access	Read access
Secure	Secure	Permitted	Permitted
	Non-secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
Non-secure	Secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
	Non-secure	Permitted	Permitted

51.3.4.2 TrustZone Filter for Standby SRAM memory regions

Standby SRAM regions can be divided into Secure and Non-secure. The access permissions for these regions are as below.

Table 51.3 Standby SRAM memory protection by Security Attribution

SA	Transaction	Write access	Read access
Secure	Secure	Permitted	Permitted
	Non-secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
Non-secure	Secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
	Non-secure	Permitted	Permitted

Standby SRAM regions can be divided into Privileged and Unprivileged. The access permissions for these regions are as below.

Table 51.4 Standby SRAM memory protection by Privilege Attribution

PA	Transaction	Write access	Read access
Privileged	Privileged	Permitted	Permitted
	Unprivileged	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
Unprivileged	Privileged	Permitted	Permitted
	Unprivileged	Permitted	Permitted

TrustZone Filter error for Standby SRAM memory generates an error notification.

For details, see section 43, Security Features .

51.3.5 Interrupt Source

The Standby SRAM interrupt source includes a Parity error and TrustZone filter error.

Parity error can choose non-maskable interrupt or reset by STBRAMCR.OAD bit.

The Standby SRAM interrupt occurs when the SRAMESR.ERRS is set to 1. the Standby SRAM interrupt continues to occur until the SRAMESR.ERRS is cleared.

When Common memory error occur(NMISR.CMST=1 or RSTSR1.CMSR=1), please read SRAMESR and check SRAM interrupt source. When the access from debugger, no error flag is set, reset and non-maskable interrupt are maskable.

For more detail on the debug mode, see section 2, CPU.

Table 51.5 Standby SRAM Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
PARITYERR	Parity error	Not possible	Not possible
TZFLT	TrustZone filter error	Not possible	Not possible

51.3.6 Wait state

When ICLK frequencies is higher than 120 MHz, do not set 0x00 in wait enable bit for the SRAMWTSC register, in order to insert a wait cycle. When the wait is not inserted, the operation is not guaranteed.

Depending on the operating frequency of ICLK, the WAIT setting for Standby SRAM access has the following conditions.

[ICLK frequency]:

- 240 MHz ≥ ICLK > 120 MHz = 1 wait
- 120 MHz ≥ ICLK = No-wait

51.4 Usage Notes

51.4.1 Instruction Fetch from the Standby SRAM Area

When using Standby SRAM to operate a program, initialize the Standby SRAM area so that the CPU can correctly prefetch data. A parity error might occur if the CPU prefetches from an area that is not initialized. Initialize the additional 12-byte area from the end address of the program with the 4-byte boundary. Renesas recommends using the NOP instruction for data initialization.

51.4.2 Notes on Self-Diagnosis of the Standby SRAM

A write buffer is mounted for the Standby Standby SRAM. When the same address is read after a write operation, data in the write buffer, rather than in the memory cell of the Standby SRAM may be read. When the Standby SRAM is self-diagnosed, confirm that the data have been written by following the procedure below so that data will not be read from the write buffer.

1. Write data to the address targeted for diagnosis.
2. Write data to an address which is at least 4 addresses away from the that in step1.
3. Read the data from the address in step1.

52. Flash Memory

This MCU incorporates code flash memory, data flash memory, and option-setting memory. The code flash memory stores instructions and operands, and the data flash memory stores data. For option-setting memory, see section 6, Option-Setting Memory.

52.1 Overview

Table 52.1 lists the specifications of the flash memory, and Figure 52.1 is block diagrams of the flash memory related modules.

The I/O pins used in boot mode, see Table 52.37.

The FCU (flash control unit) controls programming and erasure of the flash memory. The FACI (flash application command interface) controls the FCU according to the specified FACI commands.

Regarding the configuration of the code flash memory, see Figure 52.2 to Figure 52.3, and for the configuration of the data flash memory, see Figure 52.4.

Table 52.1 Specifications of flash memory (1 of 2)

Item	Code flash memory	Data flash memory
Memory capacity	User area: 2 Mbytes max	Data area: 12 Kbytes
Value after erasure	0xFF	Undefined
Programming/erasing method	• Programming and erasing the code flash memory and data flash memory, and programming the option-setting memory are handled by the FACI commands specified in the FACI command issuing area (Secure: 0x4010_0000, Non-secure: 0x5010_0000) (self-programming). • Programming/erasure through transfer by a serial-programmer via a serial interface (serial programming)	
Protection	Protects against erroneous rewriting of the flash memory	
Dual bank function	The dual-bank structure makes a safe update possible in cases where programming is suspended. • Linear mode: the code flash memory is used as one area. • Dual mode: the code flash memory is divided into two areas.	Not available
Block swap function	The block swap structure makes a safe update for a part of Non-secure application possible in case where programming is suspended.	Not available
Background operations (BGOs)	• The code flash memory can be read while the code flash memory is being programmed or erased. • The data flash memory can be read while the code flash memory is being programmed or erased. • The code flash memory can be read while the data flash memory is being programmed or erased.	
Units of programming and erasure	• Units of programming for the user area: 128 bytes • Units of erasure for the user area: Block units	• Unit of programming for the data area:4/8/16 bytes • Unit of erasure for the data area:64/128/256 bytes
Other functions	Interrupts can be accepted during self-programming. In the initial settings of this MCU, an expansion area of the option-setting memory can be set.	
On-board programming (four types)	Programming/erasure in boot mode (for the SCI interface) • The asynchronous serial interface (SCI9) is used. • The transfer rate is adjusted automatically. Programming/erasure in boot mode (for the USB interface) • USBFS is used. • Dedicated hardware is not required, so direct connection to a PC is possible. Programming/erasure in On-chip debug mode • JTAG or SWD interface is used Programming and erasure by self-programming • This allows code flash memory programming/erasure without resetting the system.	

Table 52.1 Specifications of flash memory (2 of 2)

Item	Code flash memory	Data flash memory
Unique ID	A 16-byte ID code provided for each MCU	
FACI command	Program : 128 bytes Block erase: 1 block (8 KB or 32 KB) P/E suspend P/E resume Forced Stop Status Clear Configuration set (16 bytes)	Program: 4/8/16 bytes Block Erase: 1 block (64 bytes) Multi Block Erase: 64/128/256 bytes P/E suspend P/E resume Forced Stop Blank Check: 4 bytes to data flash memory capacity Status Clear Configuration set (4/16 Bytes) Increment Counter: 1bit Refresh Counter Read Counter: 8 Bytes
Security function	Protects against illicit tampering with or reading out of data in flash memory Startup area select setting protection ● BTFLG and FSUACR registers are protected by the FSPR bit. Permanent block protect setting protection ● Code flash memory is permanently protected from programming/erasure operation by the permanent block protect function. Flash memory protection for TrustZone ● Protection for flash memory area (P/E) ● Protection for flash memory area (read) ● Protection for register ● Protection during FACI command operation. ● Code flash P/E mode entry protection ● Data flash configuration area protection ● Anti-rollback Counter	
Safety function	Software protection ● FACI command protection by FENTRYR register. ● Flash memory is protected by FWEPROR register ● The user area is protected by the block protect setting Error protection ● Error is detected when unintended commands or prohibited settings occur. The FACI command is not accepted after an error detection. Boot area protection ● The start-up area select function allows customer to safely update the boot firmware. The size of the start-up area is 8 KB.	
Interrupt request	● FRDYI (flash sequencer ready (processing end)) : Enabled by FRDYIE bit. ● FIFERR (flash sequencer error) : Enabled by CFAEIE/CMDLKIE/DFAEIE bits	
Address conversion	Start-up area select function is supported in linear mode Dual mode and Linear mode ● Bank swap function is supported in dual mode ● Block swap function is supported in linear mode	

Figure 52.1 shows how modules related to flash memory can be configured. The flash sequencer is configured with the FCU and FACI. The FCU executes basic control for rewriting of the flash memory. The FACI receives FACI commands using peripheral bus, and controls FCU operations accordingly.

In response to a reset, the FACI transfers data from the flash memory to the option byte storage registers.

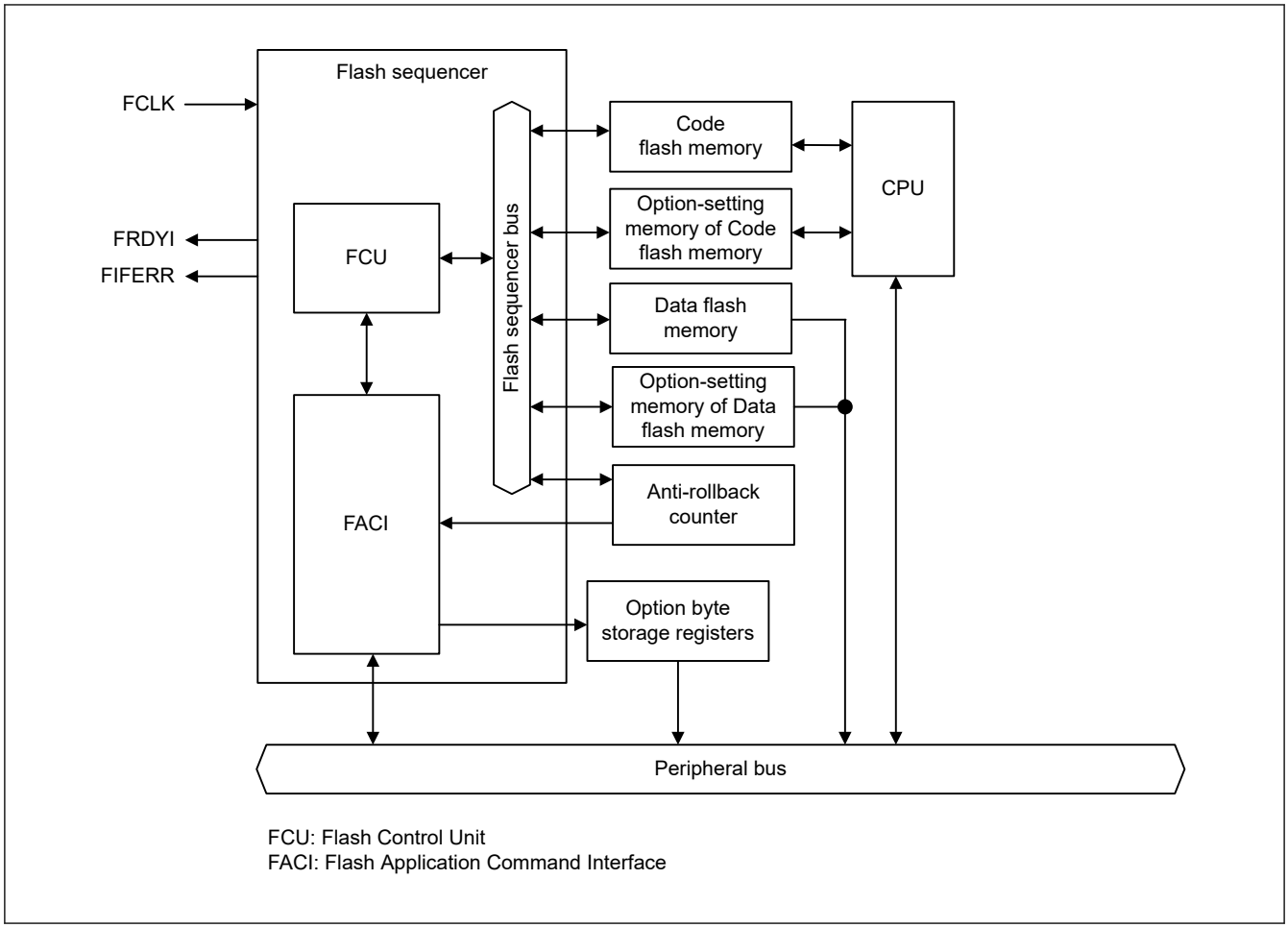


Figure 52.1 Block diagram of flash memory-related modules

52.2 Structure of Memory

Figure 52.2 shows the memory map of code flash memory in linear mode. Figure 52.3 shows the memory map of code flash memory in dual mode. This MCU can use the code flash memory as 2 bank areas by using the dual bank function. This dual-bank structure allows a safe update of a program while a user program is running.

The user area of the code flash memory in this MCU is divided into 8 KB and 32 KB blocks, which serve as the units of erasure.

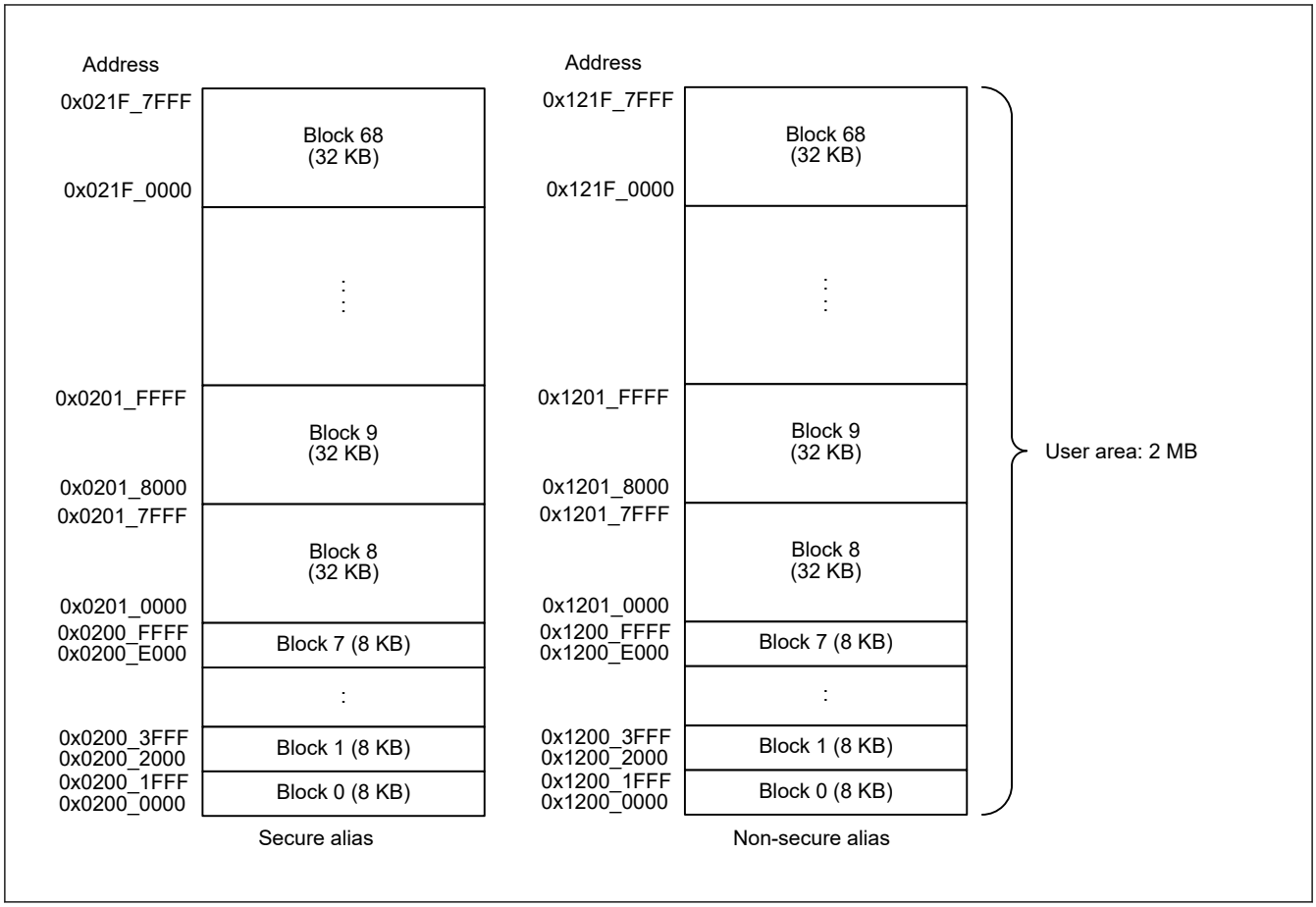


Figure 52.2 Map of the Code Flash Memory in Linear Mode

Table 52.2 Read and programming/erasure address by product for the code flash memory in Linear Mode

Product	Address	Number of blocks
2 MB product	Secure alias : 0x0200_0000 to 0x021F_7FFF Non-secure alias : 0x1200_0000 to 0x121F_7FFF	0 to 68
1 MB product	Secure alias : 0x0200_0000 to 0x020F_FFFF Non-secure alias : 0x1200_0000 to 0x120F_FFFF	0 to 37

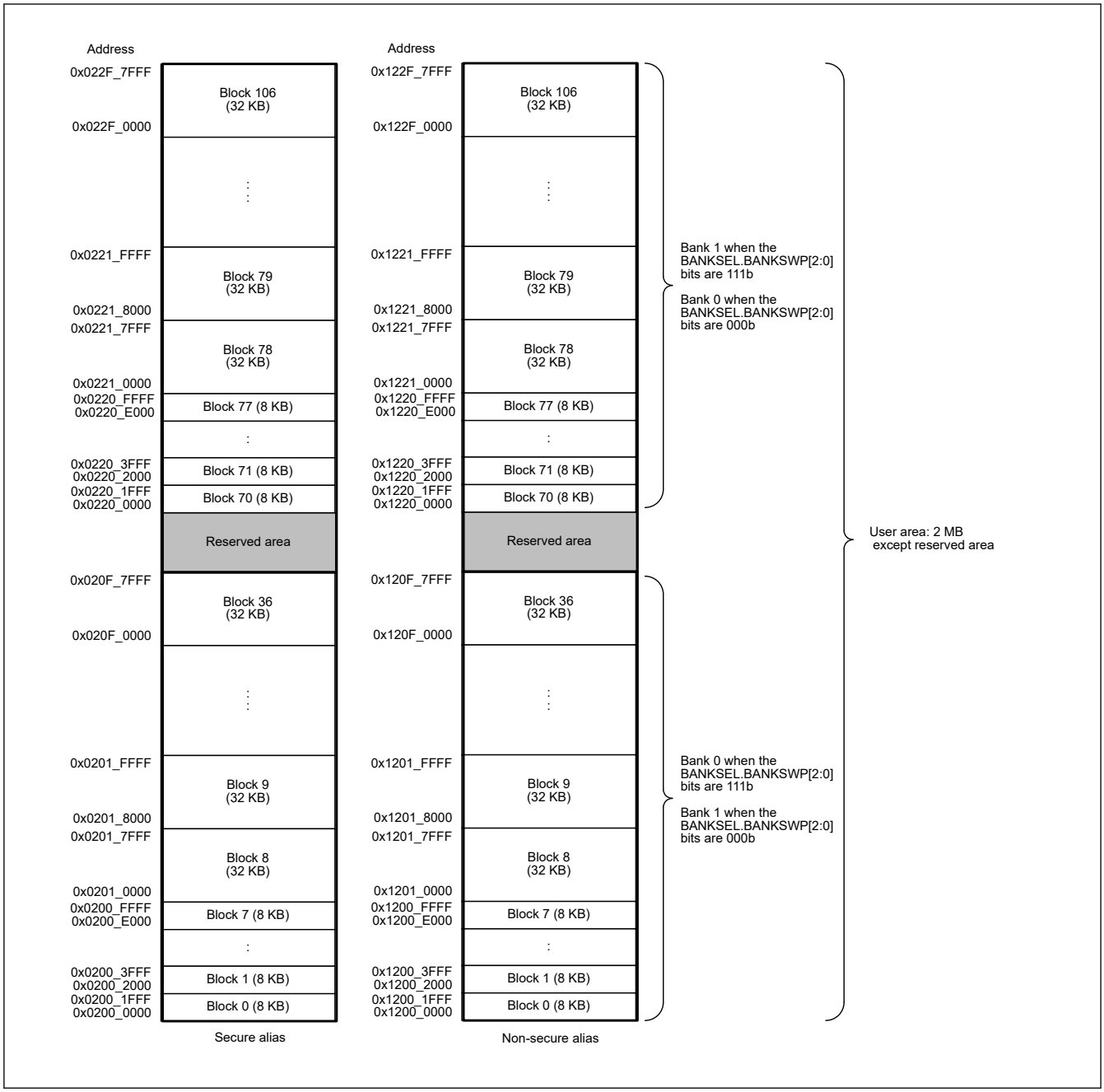


Figure 52.3 Map of the Code Flash Memory in Dual Mode

Table 52.3 Read and programming/erasure address by product for the code flash memory in Dual Mode

Product	Address	Number of blocks
2 MB product, lower side bank	Secure alias : 0x0200_0000 to 0x020F_7FFF Non-secure alias : 0x1200_0000 to 0x120F_7FFF	0 to 36
2 MB product, upper side bank	Secure alias : 0x0220_0000 to 0x022F_7FFF Non-secure alias : 0x1220_0000 to 0x122F_7FFF	70 to 106
1 MB product, lower side bank	Secure alias : 0x0200_0000 to 0x0207_FFFF Non-secure alias : 0x1200_0000 to 0x1207_FFFF	0 to 21
1 MB product, upper side bank	Secure alias : 0x0220_0000 to 0x0227_FFFF Non-secure alias : 0x1220_0000 to 0x1227_FFFF	70 to 91

The data area of the data flash memory in this MCU is divided into 64 byte blocks, with each being a unit for erasure. Figure 52.4 shows the mapping of the data flash memory.

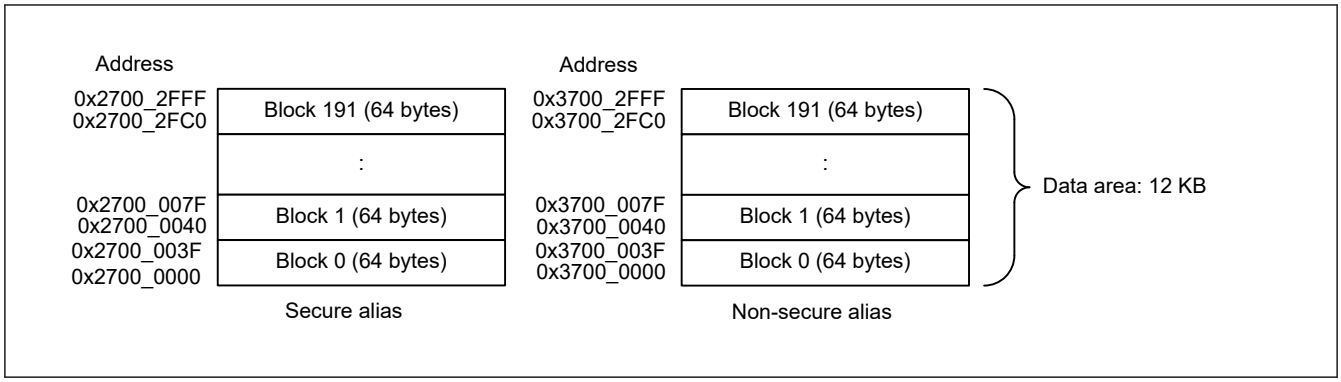


Figure 52.4 Map of the Data Flash Memory

52.3 Address Space

Using the hardware interface with flash memory requires access to all registers of the hardware, which is for issuing FACI commands. Table 52.4 provides information about the hardware interface.

Table 52.4 Information on the hardware interface area

Area	Address	Capacity
Area containing various registers of the hardware	See section 52.4. Register Descriptions.	See section 52.4. Register Descriptions.
FACI command-issuing area	Secure alias : 0x4010_0000 Non-secure alias : 0x5010_0000	4 bytes

For the address information of the flash memory, see Figure 52.2.

52.4 Register Descriptions

52.4.1 FCACHEE : Flash Cache Enable Register

Base address: FCACHE = 0x4001_C100
FCACHE_NS = 0x5001_C100

Offset address: 0x000

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FCACHEEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FCACHEEN	Flash Cache Enable 0: FCACHE is disabled 1: FCACHE is enabled	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Associated SA: FSAR register. See section 52.4.4. FSAR : Flash Security Attribution Register.

FCACHEEN bit (Flash Cache Enable)

FCACHEE.FCACHEEN bit enable and disables the function of Flash Cache of FCACHE1, FCACHE2 and FLPF.

FCACHEE.FCACHEEN bit dose not influence for FCACHEIV.FCACHEIV.

When FCACHE is enabled, it works for accesses marked as cacheable.

It is prohibited to disable FCACHE after enabling.

52.4.2 FCACHEIV : Flash Cache Invalidate Register

Base address: FCACHE = 0x4001_C100
FCACHE_NS = 0x5001_C100

Offset address: 0x004

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FCACHEIV
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FCACHEIV	Flash Cache Invalidate 0: Read: Do not invalidate. Write: The setting is ignored. 1: Invalidate FCACHE is invalidated.	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5, P-TYPE-2

This register is not controlled by any security attribute register.

FCACHEIV bit (Flash Cache Invalidate)

When 1 is written to FCACHEIV.FCACHEIV bit, the Flash cache data of FCACHE1, FCACHE2 and FLPF is invalidated.

Invalidate FCACHE with keeping FCACHE enabled after programming or erasing the code flash or the option setting memory.

52.4.3 FLWT : Flash Wait Cycle Register

Base address: FCACHE = 0x4001_C100
FCACHE_NS = 0x5001_C100

Offset address: 0x01C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	FLWT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	FLWT[2:0]	Flash Wait Cycle 0 0 0: 0 wait (ICLK ≤ 48 MHz) 0 0 1: 1 wait (48 MHz < ICLK ≤ 96 MHz) 0 1 0: 2 wait (96 MHz < ICLK ≤144 MHz) 0 1 1: 3 wait (144 MHz <ICLK ≤ 192 MHz) 1 0 0: 4 wait (192 MHz < ICLK ≤ 240 MHz) Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Associated SA: FSAR register. See section 52.4.4. FSAR : Flash Security Attribution Register.

FLWT[2:0] bits (Flash Wait Cycle)

The Flash Wait Cycle Register (FLWT) sets the access wait count for the flash memory.

For faster clock frequencies, set FLWT.FLWT before changing the clock frequency. For slower clock frequencies, set FLWT.FLWT after changing the clock frequency.

For information on the frequency setting, see section 8, Clock Generation Circuit.

52.4.4 FSAR : Flash Security Attribution Register

Base address: FCACHE = 0x4001_C100
FCACHE_NS = 0x5001_C100

Offset address: 0x040

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	FACICMRSA	FACICMISA	FCKMHZSA	—	—	—	—	—	—	FCACHESA	FLWTSA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FLWTSA	FLWT Security Attribution Target register : FLWT 0: Secure 1: Non-secure	R/W
1	FCACHESA	FCHACHEEN Security Attribution Target register : FCHACHEEN 0: Secure 1: Non-secure	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	FCKMHZSA	FCKMHZ Security Attribution Target register : FCKMHZ 0: Secure 1: Non-secure	R/W
9	FACICMISA	FACI command Issuing Security Attribution Target area : Flash access area 0: Secure 1: Non-secure	R/W
10	FACICMRSA	FACI command Registers Security Attribution Target registers : FASTAT, FAEINT, FRDYIE, FSADDR, FEADDR, FBPROT1, FSTATR, FENTRYR, FSUINITR, FCMDR, FBCCNT, FBCSTAT, FPSADDR, FSUASMON, FCPSR, FPCKAR, FWEPROR 0: Secure 1: Non-secure	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1
Note: Set the PRCR.PRC4 bit to 1 (write enabled) before rewriting this register.

FLWTSA bit (FLWT Security Attribution)

This bit sets the security attribute of FLWT.

FCACHESA bit (FCHACHEEN Security Attribution)

This bit sets the security attribute of FCHACHEEN.

FCKMHZSA bit (FCKMHZ Security Attribution)

This bit sets the security attribute of Flash access area.

FACICMISA bit (FACI command Issuing Security Attribution)

This bit sets the security attribute of Flash access area.

FACICMRSA bit (FACI command Registers Security Attribution)

This bit sets the security attribute of FACI command security registers.

52.4.5 UIDRn : Unique ID Registers n (n = 0 to 3)

Address: 0x0300_8190 + n × 4 (Secure)
0x1300_8190 + n × 4 (Non-secure)

Bit position:	31	0
Bit field:	UID	
Value after reset:	Unique value for each chip	

Bit	Symbol	Function	R/W
31:0	UID	Unique ID	R

Note: S-TYPE-5, P-TYPE-5

The UIDRn is a read-only register that stores a 16-byte ID code (unique ID) for identifying the individual MCU. The UIDRn register should be read in 32-bit units. When reading by the signature request command of the serial programming interface, the data is read in order from the data with the large address. That is, the data in 0x0300_819F is read first, and in 0x0300_8190 is read last.

52.4.6 PNRn : Part Numbering Register n (n = 0 to 3)

Address: 0x0300_80F0 + n × 4 (Secure)
0x1300_80F0 + n × 4 (Non-secure)

Bit position:	31	0
Bit field:	PNR	
Value after reset:	Unique value for each chip	

Bit	Symbol	Function	R/W
31:0	PNR	Part Number	R

Note: S-TYPE-5, P-TYPE-5

The PNRn is a read-only register that stores a 16-byte part numbering. The PNRn register should be read in 32-bit units. Each byte corresponds to the ASCII code representation of the product part number as detailed in Table 1.13. The first character ("R", 0x52 in ASCII code) of the part number is stored in the byte with the smallest address (0x0300_80F0). When reading by the signature request command of the serial programming interface, the data is read in order from the data with the small address. That is, the data in 0x0300_80F0 is read first, and in 0x0300_80FF is read last.

52.4.7 MCUVER : MCU Version Register

Address: 0x0300_81B0 (Secure)
0x1300_81B0 (Non-secure)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MCUVE							
Value after reset:	Value depend on the chip							

Bit	Symbol	Function	R/W
7:0	MCUVE	MCU Version	R

Note: S-TYPE-5, P-TYPE-5

The MCUVER is a read-only register that stores a MCU version. The MCUVER register should be read in 8-bit units.

52.4.8 FWEPROR : Flash P/E Protect Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA54

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	FLWE[1:0]	
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
1:0	FLWE[1:0]	Flash Programming and Erasure 0 0: Prohibits Program, Block Erase, Multi Block Erase, Blank Check, and Configuration set command processing. 0 1: Permits Program, Block Erase, Multi Block Erase, Blank Check, and Configuration set command processing. 1 0: Prohibits Program, Block Erase, Multi Block Erase, Blank Check, and Configuration set command processing. 1 1: Prohibits Program, Block Erase, Multi Block Erase, Blank Check, and Configuration set command processing.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

It is possible that Program, Block Erase, Multi Block Erase, Blank Check, and Configuration set command processing are prohibited by software.

The FWEPROR register is initialized by a reset from the following:

- All reset source except for VBATT_POR reset
- Transition to Deep Software Standby mode
- Transition to Software Standby mode.

FLWE[1:0] bits (Flash Programming and Erasure)

The FLWE[1:0] bits are used to set the flash P/E protection. The value after reset is 10b.

If these bits are set to other than 01b that does not allow programming and erasure of the flash memory, the following commands cannot be executed. Issuing any of the following commands leads to setting of the FLWEERR bit in the FSTATR register to 1.

Program / Block Erase / Multi Block Erase / Blank Check / Configuration set command

52.4.9 FASTAT : Flash Access Status Register

Base address: FACI = 0x4011_E000
FACI_NS = 0x5011_E000

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CFAE	—	—	CMDLK	DFAE	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	DFAE	Data Flash Memory Access Violation Flag 0: No data flash memory access violation has occurred 1: A data flash memory access violation has occurred.	R/W*1

Bit	Symbol	Function	R/W
4	CMDLK	Command Lock Flag 0: The flash sequencer is not in the command-locked state 1: The flash sequencer is in the command-locked state.	R
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	CFAE	Code Flash Memory Access Violation Flag 0: No code flash memory access violation has occurred 1: A code flash memory access violation has occurred.	R/W*1

Note: S-TYPE-3, P-TYPE-2
Note: Associated SA: FSAR register. See section 52.4.4. FSAR : Flash Security Attribution Register
Note 1. Only 0 can be written to clear the flag after 1 is read.

The FASTAT register indicates whether a code flash or data flash memory access violation has occurred. If any of the CFAE, CMDLK, and DFAE bits is set to 1, the flash sequencer enters the command-locked state (see section 52.11.2. Error Protection). To release it from the command-locked state, issue a status clear command or Forced Stop command to the flash sequencer.

DFAE bit (Data Flash Memory Access Violation Flag)

The DFAE bit indicates whether a data flash memory access violation occurred. When this bit is set to 1, the ILGLERR bit in the FSTATR register is set to 1, placing the flash sequencer in the command-locked state.

[Setting conditions]

FACI commands issued in the data flash P/E mode are as follows:

- The setting of the FSADDR or FEADDR register is the reserved portion of the data area
- The Configuration set command is issued while the setting of the FSADDR register is out of 0x2703_0050 to 0x2703_03FF.
- FACI command is issued to data flash memory with the wrong security attribution.

[Clearing conditions]

- When this bit is written as 0 after it is set to 1
- When the flash sequencer starts to process the Status Clear or Forced Stop command.

CMDLK bit (Command Lock Flag)

The CMDLK bit indicates that the flash sequencer is in the command-locked state.

[Setting conditions]

- The flash sequencer detects an error and enters the command-locked state.

[Clearing conditions]

- When the flash sequencer starts to process the Status Clear or Forced Stop command.

CFAE bit (Code Flash Memory Access Violation Flag)

The CFAE bit indicates whether a code flash memory access violation has occurred. When this bit is set to 1, the ILGLERR bit in the FSTATR register is set to 1, placing the flash sequencer in the command-locked state.

[Setting conditions]

FACI commands issued in the code flash P/E mode are as follows:

- The setting of the FSADDR register is the reserved portion of the user area
- The Configuration set command is issued while the setting of the FSADDR register is from 0x0200_A100 to 0x0200_A2F0 in self-programming mode
- FACI command is issued to code flash memory with the wrong security attribution.

[Clearing conditions]

- When this bit is written as 0 after it is set to 1

- When the flash sequencer starts to process the Status Clear or Forced Stop command.

52.4.10 FAEINT : Flash Access Error Interrupt Enable Register

Base address: FACI = 0x4011_E000
FACI_NS = 0x5011_E000

Offset address: 0x14

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CFAEIE	—	—	CMDLKIE	DFAEIE	—	—	—
Value after reset:	1	0	0	1	1	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	DFAEIE	Data Flash Memory Access Violation Interrupt Enable 0: Generation of an FIFERR interrupt request is disabled when FASTAT.DFAE is set to 1 1: Generation of an FIFERR interrupt request is enabled when FASTAT.DFAE is set to 1.	R/W
4	CMDLKIE	Command Lock Interrupt Enable 0: Generation of an FIFERR interrupt request is disabled when FASTAT.CMDLK is set to 1 1: Generation of an FIFERR interrupt request is enabled when FASTAT.CMDLK is set to 1.	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	CFAEIE	Code Flash Memory Access Violation Interrupt Enable 0: Generation of an FIFERR interrupt request is disabled when FASTAT.CFAE is set to 1 1: Generation of an FIFERR interrupt request is enabled when FASTAT.CFAE is set to 1.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Associated SA: FSAR register. See section 52.4.4. FSAR : Flash Security Attribution Register.

The FAEINT register enables or disables generation of a flash access error (FIFERR) interrupt request.

DFAEIE bit (Data Flash Memory Access Violation Interrupt Enable)

The DFAEIE bit enables or disables generation of an FIFERR interrupt request when a data flash memory access violation occurs, setting the DFAE bit in the FASTAT register to 1.

CMDLKIE bit (Command Lock Interrupt Enable)

The CMDLKIE bit enables or disables generation of an FIFERR interrupt request when the flash sequencer enters the command-locked state, setting the CMDLK bit in the FASTAT register to 1.

CFAEIE bit (Code Flash Memory Access Violation Interrupt Enable)

The CFAEIE bit enables or disables generation of an FIFERR interrupt request when a code flash memory access violation occur, setting the CFAE bit in the FASTAT register to 1.

52.4.11 FRDYIE : Flash Ready Interrupt Enable Register

Base address: FACI = 0x4011_E000
FACI_NS = 0x5011_E000

Offset address: 0x18

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	FRDYIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FRDYIE	Flash Ready Interrupt Enable 0: Generation of an FRDY interrupt request is disabled 1: Generation of an FRDY interrupt request is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Associated SA: FSAR register. See section 52.4.4. FSAR : Flash Security Attribution Register

The FRDYIE register enables or disables generation of a flash ready (FRDY) interrupt request.

FRDYIE bit (Flash Ready Interrupt Enable)

The FRDYIE bit enables or disables generation of an FRDY interrupt request when the FRDY bit in the FSTATR register is changed from 0 to 1 on completion of processing by the flash sequencer of the Program, Block Erase, Multi Block Erase, Blank Check, and Configuration set command.

52.4.12 FSADDR : FACI Command Start Address Register

Base address: FACI = 0x4011_E000
FACI_NS = 0x5011_E000

Offset address: 0x30

Bit position:	31																												0
Bit field:	FSADDR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FSADDR[31:0]	Start Address for FACI Command Processing	R/W*1

Note: S-TYPE-3, P-TYPE-2
Note: Associated SA: FSAR register. See section 52.4.4. FSAR : Flash Security Attribution Register
Note 1. These bits can be written when the FRDY bit in the FSTATR register is 1. Writing to these bits are ignored when the FRDY bit is 0. Note that b0 and b1 are read-only.

Table 52.5 FACI command address boundary

Command	Address Boundary
Program (code flash memory)	128 byte
Program (data flash memory)	4, 8, 16 byte
Block Erase (code flash memory)	8 KB or 32 KB
Block Erase (data flash memory)	64 byte
Multi Block Erase (data flash memory)	64 byte
Blank Check (data flash memory)	4 byte
For code flash configuration set	16 byte
For data flash configuration set	4, 16 byte

The FSADDR register specifies the address where the target area for command processing starts when the FACI command for Program, Block Erase, Multi Block Erase, Blank Check, or Configuration set is issued.

The FSADDR value is initialized when the SUINIT bit in the FSUINITR register is set to 1. It is also initialized by a reset.

FSADDR[31:0] bits (Start Address for FACI Command Processing)

The FSADDR[31:0] bits specify the start address for FACI command processing. Bits [31:29] and [27:24] are ignored in FACI command processing for the code flash memory. Bits [31:29] and [27:13] are ignored in FACI command processing for the data flash memory. Bit [28] is used the security alias for flash sequencer command. When bit [28] is "0", flash sequencer can update secure region with secure access. When bit [28] is "1", flash sequencer can update non-security region with secure and non-secure access. Bits associated with the address bits of lower order than the address boundary listed in Table 52.5 are also ignored.

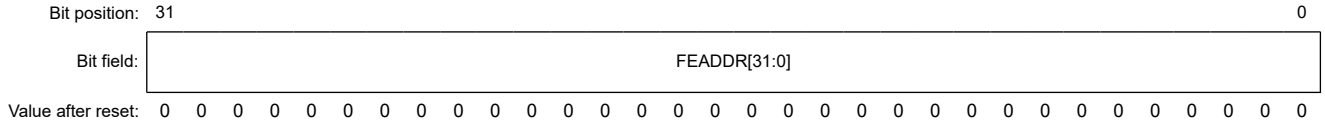
For information on the addresses of the code flash memory and the data flash memory, see section 52.2. Structure of Memory.

For information on the addresses of the configuration setting, see section 52.9.3.15. Configuration Set Command.

52.4.13 FEADDR : FACI Command End Address Register

Base address: FACI = 0x4011_E000
FACI_NS = 0x5011_E000

Offset address: 0x34



Bit	Symbol	Function	R/W
31:0	FEADDR[31:0]	End Address for FACI Command Processing	R/W*1

Note: S-TYPE-3, P-TYPE-2
Note: Associated SA: FSAR register. See section 52.4.4. FSAR : Flash Security Attribution Register
Note 1. These bits can be written when the FRDY bit in the FSTATR register is 1. Writing to these bits are ignored when the FRDY bit is 0. Note that bit [0] and bit [1] are read-only.

The FEADDR register specifies the end address of the target area for Multi Block Erase and Blank Check command processing. When incremental mode is selected as the addressing mode for Blank Checking (when FBCCNT.BCDIR = 0), the address specified in the FSADDR register should be equal to or smaller than the address in the FEADDR register. Conversely, the address in the FSADDR register should be equal to or larger than the address in the FEADDR register when decremental mode is selected as the addressing mode for Blank Check (i.e. when FBCCNT.BCDIR = 1). If the BCDIR, FSADDR, and FEADDR bit settings are inconsistent with the specified rules, the flash sequencer enters the command-locked state (see section 52.11.2. Error Protection).

The FEADDR value is initialized when the SUINIT bit in the FSUINITR register is set to 1. It is also initialized by a reset.

FEADDR[31:0] bits (End Address for FACI Command Processing)

The FEADDR[31:0] bits specify the end address for Multi Block Erase and Blank Check command processing. In command processing, bits 31 to 17 and any bits that do not reach the address boundaries listed in the section 52.4.12. FSADDR : FACI Command Start Address Register are ignored.

For information on the addresses of the flash memory, see section 52.2. Structure of Memory.

52.4.14 FMEPROT : Flash P/E Mode Entry Protection Register

Base address: FACI = 0x4011_E000

Offset address: 0x44

Bit	Symbol	Function	R/W
0	CEPROT	Code Flash P/E Mode Entry Protection 0: FENTRYC bit is not protected 1: FENTRYC bit is protected.	R/W*1 *2 *4
1	—	The read value is undefined. The write value should be 0.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	W*3

Note: S-TYPE-6, P-TYPE-2
Note 1. Writing to this bit is only possible when the FRDY bit in the FSTATR register is 1. Writing to this bit while the FRDY bit = 0 is ignored.

Note 2. Writing to this bit is only possible when 16 bits are written and the value written to the KEY bits is 0xD9.
Note 3. Written values are not retained by these bits (always read as 0x00).
Note 4. Only Secure access can write to this register. Both Secure access and Non-secure read access are allowed. Non-secure write access is denied, but TrustZone access error is not generated.

CEPROT bit (Code Flash P/E Mode Entry Protection)

The CEPROT bit specifies the protection setting of the FRNTRYC bit in the FENTRYR register.

[Setting condition]

- 1 being written to the CEPROT bit while writing to FMEPROT is enabled.

[Clearing condition]

- 0 being written to the CEPROT bit while writing to FMEPROT is enabled.

52.4.15 FCNTSELR : Flash Counter Select Register

Base address: FACI = 0x4011_E000

Offset address: 0x048

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CNTSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CNTSEL[2:0]	Counter Select 0 0 0: Anti-rollback counter is not selected 0 0 1: ARC_SEC is selected. Counter size is 64 bits. 0 1 0: ARC_OEMBL is selected. Counter size is 64 bits. 0 1 1: Anti-rollback counter is not selected. 1 x x: ARC_NSEC is selected. Counter configuration is 64 or 256 bits. See ,Table 52.6 Table 52.7 and Table 52.8 in details counter select condition. Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6, P-TYPE-2
Note: If the security attribution is configured as Secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

CNTSEL[2:0] bits (Counter Select)

FCNTSELR specifies 3 types of anti-rollback counter (ARC_SEC, ARC_NSEC, or ARC_OEMBL) in "Increment Counter", "Refresh Counter", or "Read Counter" command.

FCNTSELR value is initialized when SUINIT bit in FSUINITR is set to "1"

Note: This bit can be written when the FRDY bit in the FSTATR register is 1. Writing to this bit is ignored when the FRDY bit is 0.

Before using the ARC_NSEC counter, it is necessary to set whether the counter is to be used with 4line x 64 bits or 1line x 256 bits using the Data Flash configuration set command. The bits to be set are CNF_ARCNS0 and CNF_ARCNS1. If the CNF_ARCNS1 bit is set to 1, the ARC_NSEC counter cannot be used. Once the counter type is decided, it cannot be changed.

Table 52.6 Select condition of ARC_NSEC (64 bits) in Increment/Refresh/Read Counter Command

FCNTSELR[2:0]	Target Counter	Counter Configuration
100	ARC_NSEC[63:0]	Counter number is 4. Counter size is 64bit.
101	ARC_NSEC[127:64]	
110	ARC_NSEC[191:128]	
111	ARC_NSEC[255:192]	

Table 52.7 Select condition of ARC_NSEC (256 bits) in Increment/Refresh Counter Command

FCNTSELR[2:0]	Target Counter	Counter Configuration
100	ARC_NSEC[255:0]	Counter number is 1. Counter size is 256bit.
101		
110		
111		

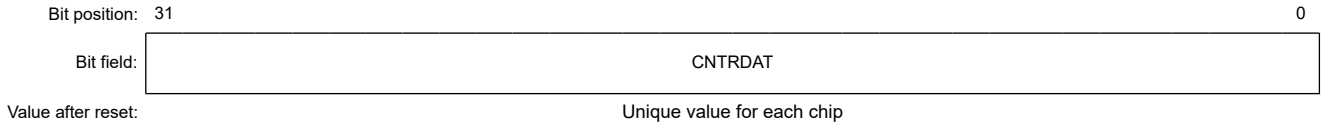
Table 52.8 Select condition of ARC_NSEC (256 bits) in Read Counter Command

FCNTSELR[2:0]	Target Counter	Counter Configuration
100	ARC_NSEC[63:0]	Counter number is 1. Counter size is 256bit.
101	ARC_NSEC[127:64]	
110	ARC_NSEC[191:128]	
111	ARC_NSEC[255:192]	

52.4.16 FCNTDATARn : Flash Counter Data Register n (n = 0, 1)

Base address: FACI = 0x4011_E000

Offset address: 0x04C + 0x004 × n (n = 0 ,1)



Bit	Symbol	Function	R/W
31:0	CNTRDAT	Counter Read Data Reading data of 32 bits (LSB side of counter values) is output to FCNTDATAR0. Reading data of 32 bits (MSB side of counter values) is output to FCNTDATAR1. .	R

Note: S-TYPE-6, P-TYPE-2

FCNTDATAR0 and FCNTDATAR1 indicate anti-rollback counter values (64 bits) that read by "Read Counter" command.

CNTRDAT bit (Counter Read Data)

[Setting condition]

- FACI has accepted "Read Counter" command.

[Clearing condition]

- FACI has accepted flash sequencer command except for "Read Counter" command.
- Flash sequencer enters "Command Lock" state

52.4.17 FBPROT0 : Flash Block Protection Register

Base address: FACI = 0x4011_E000
FACI_NS = 0x5011_E000

Offset address: 0x78

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	BPCN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BPCN0	Block Protection for Non-secure Cancel 0: Block protection is enabled 1: Block protection is disabled.	R/W*1 *2
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	W*3

Note: S-TYPE-3, P-TYPE-2
Note: Associated SA: FSAR register. See section 52.4.4. FSAR : Flash Security Attribution Register
Note 1. This bit can be written when the FRDY bit in the FSTATR register is 1. Writing to this bit is ignored when the FRDY bit is 0.
Note 2. Writing to this bit is only possible when 16 bits are written and the value written to the KEY[7:0] bits is 0x78.
Note 3. Written values are not retained by these bits (always read as 0x00).

The FBPROT0 register is used to disable the block protect function for Non-secure. When the block protect setting is locked by the permanent block setting, it cannot be disabled by this register.

The FBPROT0 value is initialized when the SUINIT bit in the FSUINITR is set to 1, because the FENTRYR value is initialized to 0x0000. It is also initialized by a reset.

BPCN0 bit (Block Protection for Non-secure Cancel)

The BPCN0 bit disables the block protect setting for Non-secure function.

[Setting condition]

- When the write-enabling conditions are satisfied and the FENTRYR is not 0x0000, write 1 to this bit.

[Clearing conditions]

- 8 bits being written to FBPROT0 while the FRDY bit is 1.
- A value other than 0x78 specified in the KEY bits and 16 bits are written to FBPROT0 while the FRDY bit is 1.
- 0 being written to the BPCN0 bit while writing to FBPROT0 is enabled.
- The FENTRYR register value is 0x0000.

52.4.18 FBPROT1 : Flash Block Protection for Secure Register

Base address: FACI = 0x4011_E000

Offset address: 0x7C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	BPCN1
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BPCN1	Block Protection for Secure Cancel 0: Block protection is enabled 1: Block protection is disabled.	R/W*1 *2
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15:8	KEY[7:0]	Key Code	W*3

Note: S-TYPE-6, P-TYPE-2
Note: Associated SA: FSAR register. See section 52.4.4. FSAR : Flash Security Attribution Register
Note 1. Writing to this bit is only possible when the FRDY bit in the FSTATR register is 1. Writing to this bit while FRDY bit = 0 is ignored.
Note 2. Writing to this bit is only possible when 16 bits are written and the value written to the KEY[7:0] bits is 0xB1.
Note 3. Written values are not retained by these bits (always read as 0x00).

The FBPROT1 register is used to disable the block protect function for Secure developer. When the block protect setting is locked by the permanent block setting, it cannot be disabled by this register.

The FBPROT1 value is initialized when the SUINIT bit in the FSUINITR is set to 1, because the FENTRYR value is initialized to 0x0000. It is also initialized by a reset.

BPCN1 bit (Block Protection for Secure Cancel)

The BPCN1 bit disables the block protect setting for Secure function.

[Setting condition]

- When the write-enabling conditions are satisfied and the FENTRYR is not 0x0000, write 1 to BPCN1.

[Clearing conditions]

- 8 bits being written to FBPROT1 while the FRDY bit is 1.
- A value other than 0xB1 specified in the KEY bits and 16 bits are written to FBPROT1 while the FRDY bit is 1.
- 0 being written to the BPCN1 bit while writing to FBPROT1 is enabled.
- The FENTRYR register value is 0x0000.

52.4.19 FSTATR : Flash Status Register

Base address: FACI = 0x4011_E000
FACI_NS = 0x5011_E000

Offset address: 0x80

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	ILGCOMERR	FESETERR	SECERR	OTERR	TZFERR	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FRDY	ILGLERR	ERSERR	PRGERR	SUSRDY	DBFULL	ERSSPD	PRGSPD	—	FLWEERR	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0.	R
6	FLWEERR	Flash Write/Erase Protect Error Flag 0: An error has not occurred 1: An error has occurred.	R
7	—	This bit is read as 0.	R
8	PRGSPD	Programming Suspend Status Flag 0: The flash sequencer is not in the programming suspension processing state or programming suspended state 1: The flash sequencer is in the programming suspension processing state or programming suspended state.	R
9	ERSSPD	Erasure Suspend Status Flag 0: The flash sequencer is not in the erasure suspension processing state or the erasure suspended state 1: The flash sequencer is in the erasure suspension processing state or the erasure suspended state.	R

Bit	Symbol	Function	R/W
10	DBFULL	Data Buffer Full Flag 0: The data buffer is empty 1: The data buffer is full.	R
11	SUSRDY	Suspend Ready Flag 0: The flash sequencer cannot receive P/E suspend commands 1: The flash sequencer can receive P/E suspend commands.	R
12	PRGERR	Programming Error Flag 0: Programming has completed successfully 1: An error has occurred during programming.	R
13	ERSERR	Erasure Error Flag 0: Erasure has completed successfully 1: An error has occurred during erasure.	R
14	ILGLERR	Illegal Command Error Flag 0: The flash sequencer has not detected an illegal FACI command or illegal flash memory access 1: The flash sequencer has detected an illegal FACI command or illegal flash memory access.	R
15	FRDY	Flash Ready Flag 0: Program, Block Erase, Multi Block Erase, P/E suspend, P/E resume, Forced Stop, Blank Check, Configuration set, Increment counter, Refresh counter, or Read counter command processing is in progress. 1: None of the above is in progress.	R
18:16	—	These bits are read as 0.	R
19	TZFERR	TrustZone Filter Error 0: A TrustZone filter error has not been detected. 1: A TrustZone filter error has been detected.	R
20	OTERR	Other Error 0: An error has not been detected. 1: An error has been detected.	R
21	SECERR	Security Error 0: A write protection error against MSUASMON.FSPR bit has not been detected. 1: A write protection error against MSUASMON.FSPR bit has been detected.	R
22	FESETERR	FENTRY Setting Error 0: A setting error in the FENTRYR register has not been detected. 1: A setting error in the FENTRYR register has been detected.	R
23	ILGCOMERR	Illegal Command Error 0: An illegal FACI command error has not been detected. 1: An illegal FACI command error has been detected.	R
31:24	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-2
Note: Associated SA: FSAR register. See section 52.4.4. FSAR : Flash Security Attribution Register

The FSTATR register indicates the state of the flash sequencer.

FLWEERR flag (Flash Write/Erase Protect Error Flag)

The FLWEERR flag indicates a violation of the flash memory overwrite protection setting in the FWEPROR register. When this flag is 1, the flash sequencer is in the command-locked state.

[Setting condition]

- An error has occurred.

[Clearing condition]

- The flash sequencer starts processing the Forced Stop command.

PRGSPD flag (Programming Suspend Status Flag)

The PRGSPD flag indicates that the flash sequencer is in the programming suspension processing state or programming suspended state.

[Setting condition]

- The flash sequencer starts processing in response to the programming suspend command.

[Clearing conditions]

- Reception of the P/E resume command by the flash sequencer (after write access to the FACI command-issuing area is complete)
- The flash sequencer starts processing the Forced Stop command.

ERSSPD flag (Erasure Suspend Status Flag)

The ERSSPD flag indicates that the flash sequencer is in the erasure suspension processing state or erasure suspended state.

[Setting condition]

- The flash sequencer starts processing in response to an erasure suspend command.

[Clearing condition]

- Reception of the P/E resume command by the flash sequencer (after write access to the FACI command-issuing area is complete)
- The flash sequencer starts processing of the Forced Stop command.

DBFULL flag (Data Buffer Full Flag)

The DBFULL flag indicates the state of the data buffer when the program command is issued. The flash sequencer incorporates a buffer for write data (data buffer). When data for writing to the flash memory are written to the FACI command-issuing area while the data buffer is full, the flash sequencer inserts a wait cycle in the peripheral bus.

[Setting condition]

- The data buffer becomes full while program commands are issued.

[Clearing condition]

- The data buffer becomes empty.

SUSRDY flag (Suspend Ready Flag)

The SUSRDY flag indicates whether the flash sequencer can receive a P/E suspend command.

[Setting condition]

- After starting programming/erasure processing, the flash sequencer enters a state in which P/E suspend commands can be received.

[Clearing conditions]

- Reception of the P/E suspend command or Forced Stop command by the flash sequencer (after write access to the FACI command-issuing area is complete)
- During programming or erasure, the flash sequencer enters the command-locked state
- Programming or erasure has completed.

PRGERR flag (Programming Error Flag)

The PRGERR flag indicates the result of programming of the flash memory. When this flag is 1, the flash sequencer is in the command-locked state.

[Setting condition]

- An error has occurred during programming.

[Clearing condition]

- The flash sequencer starts processing of the Status Clear or Forced Stop command.

ERSERR flag (Erasure Error Flag)

The ERSERR flag indicates the result of erasure of the flash memory. When this flag is 1, the flash sequencer is in the command-locked state.

[Setting condition]

- An error has occurred during erasure.

[Clearing condition]

- The flash sequencer starts processing of the Status Clear or Forced Stop command.

ILGLERR flag (Illegal Command Error Flag)

The ILGLERR flag indicates that the flash sequencer has detected an illegal FACI command or flash memory access. If this flag is 1, the flash sequencer is in the command-locked state.

[Setting conditions]

- See section 52.11.2. Error Protection.

[Clearing condition]

- The flash sequencer starts processing of the Status Clear or Forced Stop command.

FRDY flag (Flash Ready Flag)

The FRDY flag indicates the command processing state of the flash sequencer.

[Setting conditions]

- The flash sequencer completes command processing
- The flash sequencer receives a P/E suspend command and suspends programming of the flash memory
- The flash sequencer received the Forced Stop command and ended command processing.

Note: In the case of program command processing, the FRDY flag might be set to 1 even if the flash sequencer does not complete command processing. For details, see section 52.9.3.7. Program Command.

[Clearing conditions]

- The flash sequencer received an FACI command
- For Program and Configuration setting, the first write access to the FACI command-issuing area
- For other commands, the last write access to the FACI command-issuing area.

TZFERR flag (TrustZone Filter Error)

This flag indicates the TrustZone protection error in the FACI command.
See Table 52.30. When this bit is "1", flash sequencer enters "Command Lock" state.

[Setting conditions]

- An error has been detected.

[Clearing conditions]

- Status Clear or Forced Stop command processing is completed.

OTERR flag (Other Error)

This flag indicates that an FACI command has been issued when the condition of accepting commands is not satisfied.
See Table 52.30. When this flag is 1, the flash sequencer is in the command-lock state.

[Setting condition]

- An error has occurred.

[Clearing condition]

- The status clear or forced stop command processing is complete.

SECERR flag (Security Error)

This flag indicates that write protection by the MSUASMON.FSPR bit is violated.
See Table 52.30. When this flag is 1, the flash sequencer is in the command-lock state.

[Setting condition]

- An error has occurred.

[Clearing condition]

- The status clear or forced stop command processing is complete.

FESETERR flag (FENTRY Setting Error)

This flag indicates that a value of 0xAA81 is written in the FENTRYR register or the value in the FENTRYR register differs when P/E is suspended and resumed.
See Table 52.30. When this flag is 1, the flash sequencer is in the command-lock state.

[Setting condition]

- An error has occurred.

[Clearing condition]

- The status clear or forced stop command processing is complete.

ILGCOMERR flag (Illegal Command Error)

This flag indicates that the flash sequencer has detected an illegal FACI command.
See Table 52.30. When this flag is 1, the flash sequencer is in the command-lock state.

[Setting condition]

- An error has occurred.

[Clearing condition]

- The status clear or forced stop command processing is complete.

52.4.20 FENTRYR : Flash P/E Mode Entry Register

Base address: FACI = 0x4011_E000
FACI_NS = 0x5011_E000

Offset address: 0x84

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								FENTRYD	—	—	—	—	—	—	FENTRYC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FENTRYC	Code Flash P/E Mode Entry 0: Code flash is in read mode 1: Code flash is in P/E mode.	R/W*1*2
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	FENTRYD	Data Flash P/E Mode Entry 0: Data flash is in read mode 1: Data flash is in P/E mode.	R/W*1*2
15:8	KEY[7:0]	Key Code	W*3

Note: S-TYPE-3, P-TYPE-2
Note: Associated SA: FSAR register. See section 52.4.4. FSAR : Flash Security Attribution Register
Note 1. These bits can be written when the FRDY bit in the FSTATR register is 1. Writing to these bits are ignored when the FRDY bit is 0.
Note 2. Writing to these bits is only possible when 16 bits are written and the value written to the KEY[7:0] bits is 0xAA.
Note 3. Written values are not retained by these bits (always read 0x00).

FENTRYR is used to specify code flash P/E mode or data flash P/E mode. To specify the code flash P/E mode or data flash P/E mode so that the flash sequencer can receive FACI commands, set either the FENTRYD or FENTRYC bit to 1 to place the flash sequencer in P/E mode.

FENTRYR is initialized when the SUINIT bit in FSUINITR is set to 1. It is also initialized by a reset.

Note: Writing a value of 0XAA81 to this register causes the ILGLERR bit in the FSTATR register to be set to 1, resulting in the flash sequencer being placed in the command-locked state.

FENTRYC bit (Code Flash P/E Mode Entry)

The FENTRYC bit specifies P/E mode for the code flash memory.

[Setting condition]

- Write 1 to the FENTRYC bit while writing to FENTRYR is enabled and FENTRYR is 0x0000.

[Clearing conditions]

- Write 8 bits to FENTRYR while the FRDY bit is 1
- A value other than 0xAA is specified in the KEY[7:0] bits and 16 bits are written to FENTRYR while the FRDY bit is 1
- Write 0 to the FENTRYC bit while writing to FENTRYR is enabled
- Write to FENTRYR while writing is enabled and its value is other than 0x0000
- The protection of FMEPROT register is enabled.

FENTRYD bit (Data Flash P/E Mode Entry)

The FENTRYD bit specifies P/E mode for the data flash memory.

[Setting condition]

- Write 1 to the FENTRYD bit while writing to FENTRYR is enabled and FENTRYR is 0x0000.

[Clearing conditions]

- Write 8 bits to FENTRYR while the FRDY bit is 1
- Writing of 16 bits to FENTRYR with a value other than 0xAA specified for the KEY[7:0] bits while the FRDY bit is 1
- Write 0 to the FENTRYD bit while writing to FENTRYR is enabled
- Write to FENTRYR while writing is enabled and its value is other than 0x0000.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits control writing permission to the FENTRYD or FENTRYC bits.

52.4.21 FSUINITR : Flash Sequencer Setup Initialization Register

Base address: FACI = 0x4011_E000
FACI_NS = 0x5011_E000

Offset address: 0x8C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	SUINIT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SUINIT	Set-Up Initialization 0: The FSADDR, FEADDR, FBPROT0, FBPROT1, FENTRYR, FBCCNT, FCPSR, and FCNTSELR flash sequencer setup registers keep their current values 1: The FSADDR, FEADDR, FBPROT0, FBRPOT1, FENTRYR, FBCCNT, FCPSR, and FCNTSELR flash sequencer setup registers are initialized.	R/W*1*2
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15:8	KEY[7:0]	Key Code	W*3

Note: S-TYPE-3, P-TYPE-2
Note: Associated SA: FSAR register. See section 52.4.4. FSAR : Flash Security Attribution Register
Note 1. This bit can be written when the FRDY bit in the FSTATR register is 1. Writing to this bit is ignored when the FRDY bit is 0.
Note 2. Writing to these bits is only possible when 16 bits are written and the value written to the KEY[7:0] bits is 0x2D.
Note 3. Written values are not retained by these bits (always read 0x00).

FSUINITR is used for initialization of the flash sequencer setup.

SUINIT bit (Set-Up Initialization)

The SUINIT bit initializes the following flash sequencer setup registers:

- FSADDR
- FEADDR
- FBPROT0
- FBPROT1
- FENTRYR
- FBCCNT
- FCPSR
- FCNTSELR

KEY[7:0] bits (Key Code)

The KEY[7:0] bits control writing permission to the SUINIT bit.

52.4.22 FCMDR : FACI Command Register

Base address: FACI = 0x4011_E000
FACI_NS = 0x5011_E000

Offset address: 0xA0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMDR[7:0]								PCMDR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PCMDR[7:0]	Pre-command Flag The command just before the latest command is stored.	R
15:8	CMDR[7:0]	Command Flag The latest command is stored.	R

Note: S-TYPE-3, P-TYPE-2
Note: Associated SA: FSAR register. See section 52.4.4. FSAR : Flash Security Attribution Register

FCMDR records the two most recent commands accepted by the flash sequencer.

PCMDR[7:0] bits (Pre-command Flag)

The PCMDR[7:0] bits indicate the command received immediately before the latest command received by the flash sequencer.

CMDR[7:0] bits (Command Flag)

The CMDR[7:0] bits indicate the latest command received by the flash sequencer.

Table 52.9 States of FCMDR after receiving commands

Command	CMDR	PCMDR
Program	0xE8	Previous command
Block erase	0xD0	0x20
Multi block erase	0xD0	0x21
P/E suspend	0xB0	Previous command
P/E resume	0xD0	Previous command
Status Clear	0x50	Previous command
Forced Stop	0xB3	Previous command
Blank Check	0xD0	0x71
Configuration set	0x40	Previous command
Increment Counter	0xD0	0x35
Refresh Counter	0xD0	0x37
Read Counter	0xD0	0x39

52.4.23 FBCCNT : Blank Check Control Register

Base address: FACI = 0x4011_E000
FACI_NS = 0x5011_E000

Offset address: 0xD0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	BCDIR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BCDIR	Blank Check Direction 0: Blank checking is executed from the lower addresses to the higher addresses (incremental mode) 1: Blank checking is executed from the higher addresses to the lower addresses (decremental mode).	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Associated SA: FSAR register. See section 52.4.4. FSAR : Flash Security Attribution Register

FBCCNT specifies the addressing mode in processing the Blank Check command. FBCCNT is initialized when the SUINIT bit in FSUINITR is set to 1. It is also initialized by a reset.

BCDIR bit (Blank Check Direction)

The BCDIR bit specifies the addressing mode for Blank Check.

52.4.24 FBCSTAT : Blank Check Status Register

Base address: FACI = 0x4011_E000
FACI_NS = 0x5011_E000

Offset address: 0xD4

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	BCST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BCST	Blank Check Status Flag 0: The target area is in the non-programmed state, that is, the area has been erased but has not yet been reprogrammed 1: The target area has been programmed with 0s or 1s.	R
7:1	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-2
Note: Associated SA: FSAR register. See section 52.4.4. FSAR : Flash Security Attribution Register

FBCSTAT stores the results of checking in response to the Blank Check command.

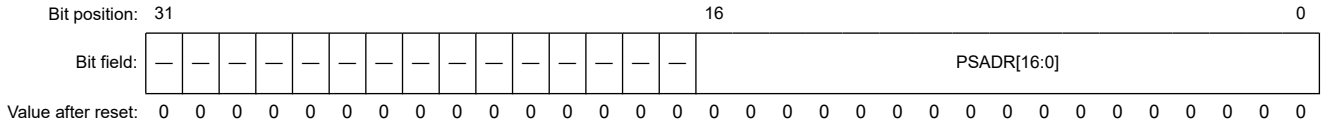
BCST flag (Blank Check Status Flag)

The BCST flag indicates the results of checking in response to the Blank Check command.

52.4.25 FPSADDR : Data Flash Programming Start Address Register

Base address: FACI = 0x4011_E000
FACI_NS = 0x5011_E000

Offset address: 0xD8



Bit	Symbol	Function	R/W
16:0	PSADR[16:0]	Programmed Area Start Address The address of the first programmed area	R
31:17	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE-3, P-TYPE-2
Note: Associated SA: FSAR register. See section 52.4.4. FSAR : Flash Security Attribution Register

FPSADDR indicates the address of the first programmed area found in processing of the Blank Check command.

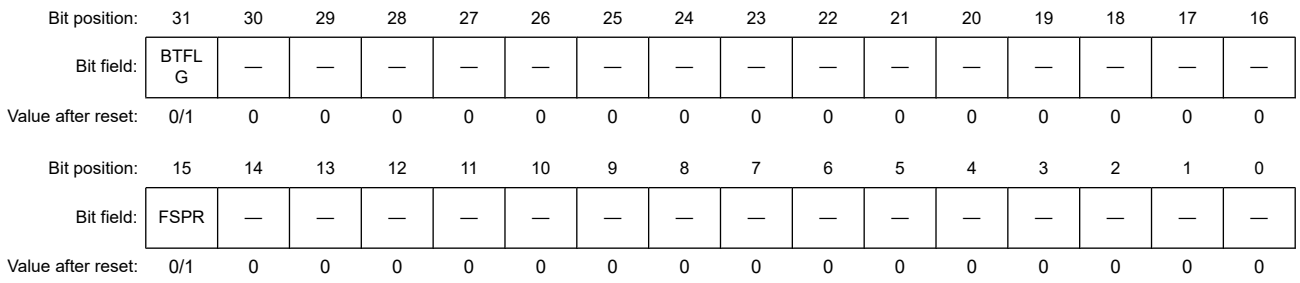
PSADR[16:0] bits (Programmed Area Start Address)

The PSADR[16:0] bits indicate the address of the first programmed area found in processing of the Blank Check command. The address is an offset from the address where the data flash memory starts. The setting of these bits is only valid if the BCST bit in the FBCSTAT register is 1 and while the FRDY bit in the FSTATR register is 1. When the BCST bit in the FBCSTAT register is 0, the PSADR[16:0] bits hold the address produced by the previous check.

52.4.26 FSUASMON : Flash Startup Area Select Monitor Register

Base address: FACI = 0x4011_E000
FACI_NS = 0x5011_E000

Offset address: 0xDC



Bit	Symbol	Function	R/W
14:0	—	These bits are read as 0. The write value should be 0.	R
15	FSPR	Protection Programming Flag to set Boot Flag and Startup Area Control 0: Protected state 1: Non-protected state.	R
30:16	—	These bits are read as 0. The write value should be 0.	R
31	BTFLG	Flag of Startup Area Select for Boot Swap 0: The startup area is the alternate block (block 1) 1: The startup area is the default block (block 0).	R

Note: S-TYPE-3, P-TYPE-2
Note: Associated SA: FSAR register. See section 52.4.4. FSAR : Flash Security Attribution Register.

FSPR bit (Protection Programming Flag to set Boot Flag and Startup Area Control)

The FSPR bit indicates the protection state against the configuration set command for the BTFLG bit, and FSUACR Register.

In response to a reset or configuration set command, the FACI transfers data from flash memory to this register.

BTFLG bit (Flag of Startup Area Select for Boot Swap)

The BTFLG bit indicates whether the address of the startup area is exchanged for the boot swap function or not.

In response to a reset or configuration set command, the FACI transfers data from flash memory to this register.

52.4.27 FCPSR : Flash Sequencer Processing Switching Register

Base address: FACI = 0x4011_E000
FACI_NS = 0x5011_E000

Offset address: 0xE0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ESUSPMD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ESUSPMD	Erasure Suspend Mode 0: Suspension priority mode 1: Erasure priority mode.	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Associated SA: FSAR register. See section 52.4.4. FSAR : Flash Security Attribution Register

FCPSR selects the erasure suspension mode. FCPSR is initialized when the SUINIT bit in FSUINITR is set to 1. It is also initialized by a reset.

ESUSPMD bit (Erasure Suspend Mode)

The ESUSPMD bit selects the erasure suspension mode when a P/E suspend command is issued while the flash sequencer is executing erasure processing (see section 52.9.3.10. P/E Suspend Command). This bit should be set before issuing Block Erase or Multi Block Erase command.

52.4.28 FPCKAR : Flash Sequencer Processing Clock Notification Register

Base address: FACI = 0x4011_E000
FACI_NS = 0x5011_E000

Offset address: 0xE4

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								PCKA[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0

Bit	Symbol	Function	R/W
7:0	PCKA[7:0]	Flash Sequencer Operating Clock Notification These bits are used to set the operating frequency of the flash sequencer while processing FACI commands.	R/W*1*2
15:8	KEY[7:0]	Key Code	W*3

Note: S-TYPE-3, P-TYPE-2
Note: Associated SA: FSAR register. See section 52.4.4. FSAR : Flash Security Attribution Register
Note 1. This bit can be written when the FRDY bit in the FSTATR register is 1. Writing to this bit is ignored when the FRDY bit is 0.
Note 2. Writing to these bits is only possible when 16 bits are written and the value written to the KEY[7:0] bits is 0x1E.
Note 3. Written values are not retained by these bits (always read 0x00).

FPCKAR specifies the operating frequency of the flash sequencer while processing FACI commands. The highest operating frequency for the given product is set as the initial value.

PCKA[7:0] bits (Flash Sequencer Operating Clock Notification)

The PCKA[7:0] bits specify the operating frequency of the flash sequencer while processing FACI commands. Set the desired frequency for these bits before issuing an FACI command. Specifically, convert the frequency in MHz to a binary number and set it for these bits.

Example:

Frequency is 35.9 MHz (PCKA = 0x24)

Round up the first decimal place of 35.9 MHz to a whole number (= 36) and convert it into a binary number.

If the value set in these bits is smaller than the actual operating frequency of the flash sequencer, the flash memory programming/erasure characteristics cannot be guaranteed. If the value set in these bits is greater than the actual operating frequency of the flash sequencer, the flash memory programming/erasure characteristics can be guaranteed but the FACI command processing time such as the time programming/erasure takes will increase. The minimum FACI command processing time is obtained when the operating frequency of the flash sequencer is the same as the PCKA value.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits control writing permission to the PCKA bit.

52.4.29 FSUACR : Flash Startup Area Control Register

Base address: FACI = 0x4011_E000

Offset address: 0xE8

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	SAS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SAS[1:0]	Startup Area Select 0 0: Startup area is selected by BTFLG bit 0 1: Startup area is selected by BTFLG bit 1 0: Startup area is temporarily switched to the default area (block 0) 1 1: Startup area is temporarily switched to the alternate area (block 1).	R/W*1 *3
7:2	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	W*2

Note: S-TYPE-6, P-TYPE-2
Note 1. Following described the write condition of these bits (these conditions are required at the same time).
1. Access size to this register is 16 bits
2. The value of KEY[7:0] is 0x66
3. The FSPR bit is 1.

Note 2. Written values are not retained by these bits (always read 0x00).
Note 3. Only Secure access can write to this register. Both Secure access and Non-secure read access are allowed. Non-secure write access is denied, but TrustZone access error is not generated.

FSUACR sets the startup area for the boot swap function. Do not use this register in dual mode (the DUALSEL.BANKMD[2:0] bits are 000b). In dual mode, starting up proceeds from startup area 0.

SAS[1:0] bits (Startup Area Select)

The SAS[1:0] bits select the startup area. Three methods are available for changing the startup area.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits control writing permission to the SAS [1:0] bits.

52.4.30 FCKMHZ : Data Flash Access Frequency Register

Base address: FLAD = 0x4011_C000
FLAD_NS = 0x5011_C000

Offset address: 0x40

Bit position:	7	6	5	4	3	2	1	0
Bit field:	FCKMHZ[7:0]							
Value after reset:	0	0	1	1	1	1	0	0

Bit	Symbol	Function	R/W
7:0	FCKMHZ[7:0]	Data Flash Access Frequency Register These bits optimize the speed of reading the data flash memory.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: If the security attribution is configured as Secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

This register optimizes the speed of reading the data flash memory.

Set the frequency of the peripheral module clock (FCLK) of internal peripheral bus which is the clock for access to the data flash memory, in MHz units. For example, 35.9 MHz should be rounded up and set the frequency to 36. Number of cycles required for access to the data flash memory are inserted according to the frequency. When changing the frequency of the FCLK, follow the procedure below to modify the value of the data flash access frequency register (FCKMHZ) in either of the following ways according to whether operation is at a lower frequency before or after the change.

- When changing the speed from low to high: Modify FCKMHZ. After confirming the change by reading FCKMHZ, change the frequency.
- When changing the speed from high to low: Change the frequency. After the frequency is changed, modify FCKMHZ.

52.5 Flash Cache

52.5.1 Feature of flash cache

The FCACHE (Flash Cache) speeds up read access from bus master to the flash memory. The FCACHE includes:

- FCACHE1, for CPU instruction fetches
- FCACHE2, for CPU operand access and access from DMAC/DTC
- FLPF, for the prefetch access in CPU instruction fetches

Table 52.10 Flash Cache 1 (FCACHE1) overview

Item	Description
Cache Target Region	Secure alias : 0x0200_0000 - 0x021F_7FFF Non-secure alias : 0x1200_0000 - 0x121F_7FFF
Target Bus Master	CPU instruction Fetch
Capacity	256 bytes
Associativity	8WAY set associative
	128 bits/entry (128 bit aligned data), 2 entries/way
Access Cycle	Cache Hit : 0 wait Cache Miss : wait number of Flash Wait Cycle Register

Table 52.11 Flash Cache 2 (FCACHE2) overview

Item	Description
Cache Target Region	Secure alias : 0x0200_0000 - 0x021F_7FFF Non-secure alias : 0x1200_0000 - 0x121F_7FFF
Target Bus Master	CPU Operand Access and Access from DMAC/DTC
Capacity	16 bytes
Associativity	Full Associative
	128 bits/entry (128 bit aligned data), 1 entry
Access Cycle	Cache Hit : 0 wait Cache Miss : wait number of Flash Wait Cycle Register

Table 52.12 Prefetch Buffer (FLPF) overview

Item	Description
Cache Target Region	Secure alias : 0x0200_0000 - 0x021F_7FFF Non-secure alias : 0x1200_0000 - 0x121F_7FFF
Capacity	32 bytes
Associativity	Full Associative
	128 bits/entry (128 bit aligned data), 2 entries
Request Address	Next address of previous CPU Instruction
Access Cycle	Cache Hit : 0 wait Cache Miss : wait number of Flash Wait Cycle Register

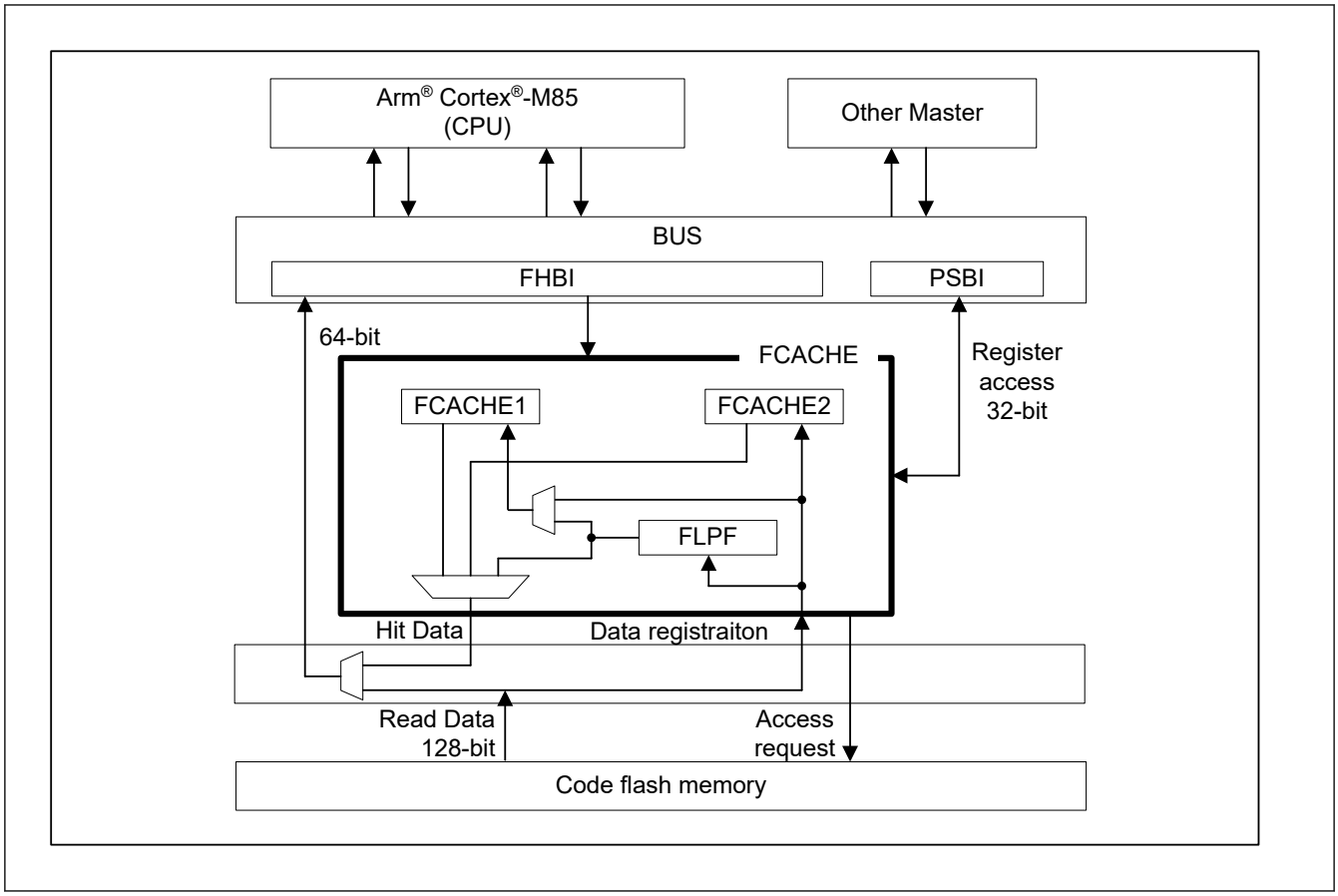


Figure 52.5 Block diagram of FCACHE

52.6 Operating Modes Associated with Flash Memory

Figure 52.6 is a diagram of the mode transitions associated with the flash memory. For the procedures for setting the modes, see section 6, Option-Setting Memory.

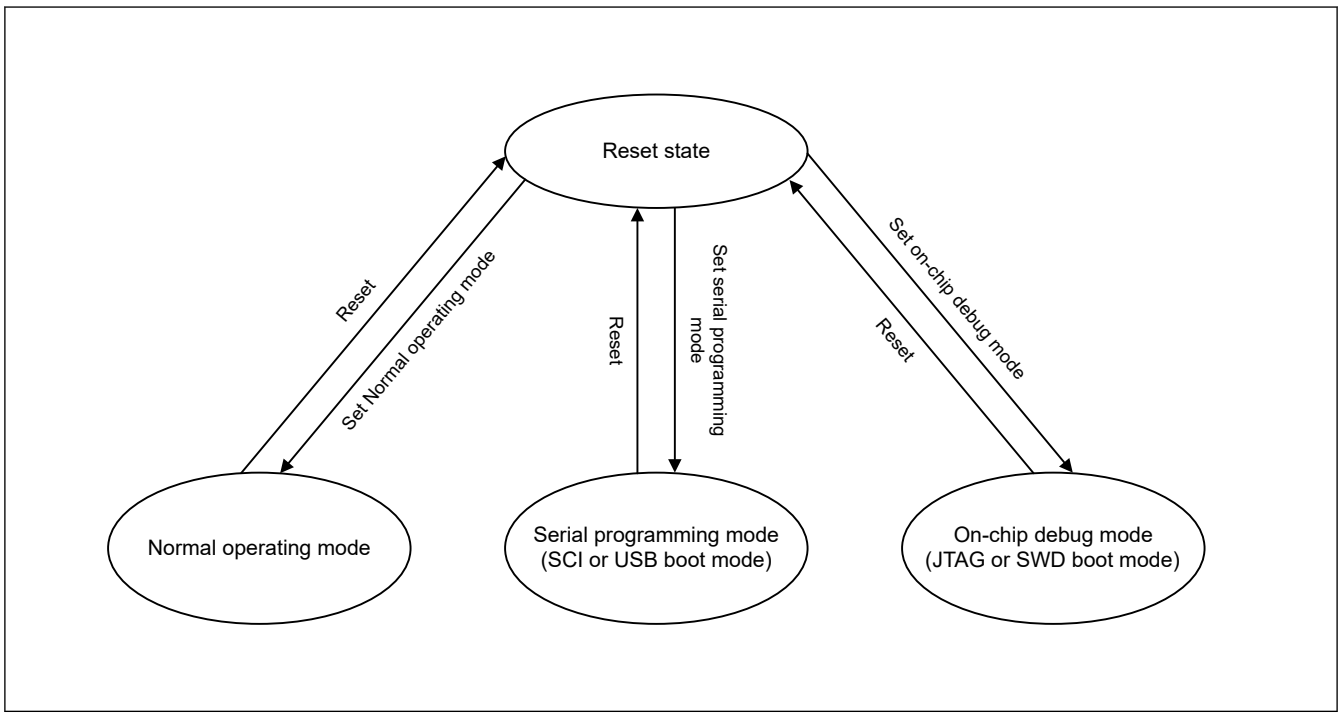


Figure 52.6 Mode Transitions Associated with Flash Memory

The flash memory area where programming and erasure are permitted and the boot program after a reset are different according to each mode. The differences between modes are listed in Table 52.13.

Table 52.13 Differences between Modes

Parameter	Normal operating mode	Serial programming mode (SCI or USB boot mode)	On-chip debug mode (JTAG or SWD boot mode)
Programmable and erasable areas	• Code flash memory • Data flash memory • Option setting memory (programming only)	• Code flash memory • Data flash memory • Option-setting memory (programming only)	• Code flash memory • Data flash memory • Option setting memory (programming only)
Erasure in block units	Possible	Possible	Possible
Boot program at a reset	User area program	Embedded program for serial programming	Depends on debug command

52.7 Overview of Functions

By using a dedicated flash-memory programmer to program the flash memory through a serial interface (serial programming) or JTAG/SWD interface (on-chip debug mode), the device can be rewritten regardless of whether this is before or after it is mounted on the target system.

Furthermore, security functions to prohibit rewriting or reading of the user program written to the flash memory are incorporated, and this can prevent falsification and illicit reading of the programs by third parties.

Programming by the user program (self-programming) is available to suit applications where the application on the target system may require updating after manufacturing or shipment. Protection features for the safe rewriting of the flash memory are also incorporated. Furthermore, interrupt processing during self-programming is supported, so programming can proceed at the same time as processing for external communications, etc., and this is the case in various situations. Table 52.14 lists the overview of the methods of programming and the corresponding operating modes.

Table 52.14 Programming methods

Programming method	Functional overview	Operating mode
Serial programming	A dedicated flash-memory programmer through the SCI or USBFS interface enables on-board programming of the flash memory after the device is mounted on the target system.	Serial programming mode
	A dedicated flash-memory programmer through the SCI or USBFS interface and a dedicated programming adapter board allow off-board programming of the flash memory, for example, programming of the device before it is mounted on the target system.	
Self-programming	A user program written to memory in advance of serial programming execution can also program the flash memory. The background operation capability makes it possible to fetch instructions or otherwise read data from the code flash memory while the data flash memory is programmed. As a result, a program resident in the code flash memory is able to program the data flash memory. For background operations that are not possible, instructions in the code flash memory cannot be fetched and data cannot be accessed while the code flash memory is being programmed by self-programming. In such cases, a program for programming from the internal SRAM or external memory must be transferred in advance and executed.	Normal operating mode
JTAG or SWD programming	A dedicated flash-memory programmer or an on-chip debugger through JTAG or SWD enables on-board programming of the flash memory after the device is mounted on the target system. A dedicated flash-memory programmer or an on-chip debugger through JTAG or SWD and a dedicated programming adapter board allow off-board programming of the flash memory, for example, programming of the device before it is mounted on the target system.	On-chip debug mode

Table 52.15 lists the functions of the flash memory. Serial programmer commands realize each function of serial programming, while reading of the flash memory by an FACI command or the user program realizes each function of self-programming.

Table 52.15 Basic Functions

Function	Functional overview	Availability	
		Serial programming	Self-programming
Blank check	Checks a specified block to ensure that writing to it has not already proceeded. Results of reading from data flash memory to which nothing is written after erasure are not guaranteed, so use blank checking to confirm that writing to memory has not proceeded after erasure.	Not supported	Supported (data flash programming only)
Block erasure	Erases the memory contents in the specified block	Supported	Supported
Programming	Writes to the specified address	Supported	Supported
CRC	Calculates the CRC in the specified range of the flash memory and transfers it to the flash programmer	Supported	Not supported
Read	Reads data programmed in the flash memory	Supported	Not supported (read by user program is possible)
Start-up program protection functions	Configures the start-up program protection functions	Supported	Supported
Option function selection	Selects the option function, and modifies the initial setting of this MCU	Supported	Supported
Dual bank function	Switches different modes (linear or dual)	Supported	Supported
Block swap function	Setting block swap functions	Supported	Supported
Block protection	Setting block protection	Supported	Supported
Device lifecycle transition	Transitions the device lifecycle	Supported	Not supported
Memory security attribution	Setting the memory security attribution	Supported	Not supported
Key	Injects key	Supported	Supported (except the key related to device lifecycle transition)
All erasure	Erase the flash memory to the state after shipment	Supported	Not supported

The flash memory supports various security functions.

Table 52.16 lists the security functions supported by the flash memory.

Table 52.16 Lists of Security Functions

Function	Description
Security flag for Start-up area select	Start-up area selection can be protected by setting of security flag (FSPR).
Permanently block protection	Programming or erasure of each block of code flash memory can be protected permanently.
Protection for TrustZone	Programming or erasure area, readable area, register access, and FACI command operation are protected by ARM TrustZone security.
Data flash configuration area protection	When target area of configuration set command includes protection area by lock bit, flash sequencer detects an error and enters command lock state.
Anti-rollback Counter	Anti-rollback counter used only by Secure applications for firmware updates.
Programming or erasure mode protection	Only Secure developer can enter the programming or erasure mode for code flash.

52.8 Operating Modes of the Flash Sequencer

The flash sequencer has three operating modes as shown in Figure 52.7. Transitions between modes are initiated by changing the value of the FENTRYR register.

When the value of the FENTRYR register is 0x0000, the flash sequencer is in read mode. In this mode, it does not receive FACI commands. The code flash memory and data flash memory are both readable.

When the value of the FENTRYR register is 0x0001, the flash sequencer is in code flash P/E mode where the code flash memory can be programmed or erased by FACI commands. In this mode, the data flash memory is readable. In addition, the code flash memory is not readable if background operation (BGO) is disabled. If BGO is enabled, the code flash memory

that is not selected by FSADDR register is readable when the FRDY bit in the FSTATR register is 0. As for the condition for enabling BGO, see section 52.15.2. Background Operation.

When the value of the FENTRYR register is 0x0080, the flash sequencer is in data flash P/E mode where the data flash memory can be programmed or erased by FACI commands. In this mode, the data flash memory is not readable. However, the code flash memory is readable.

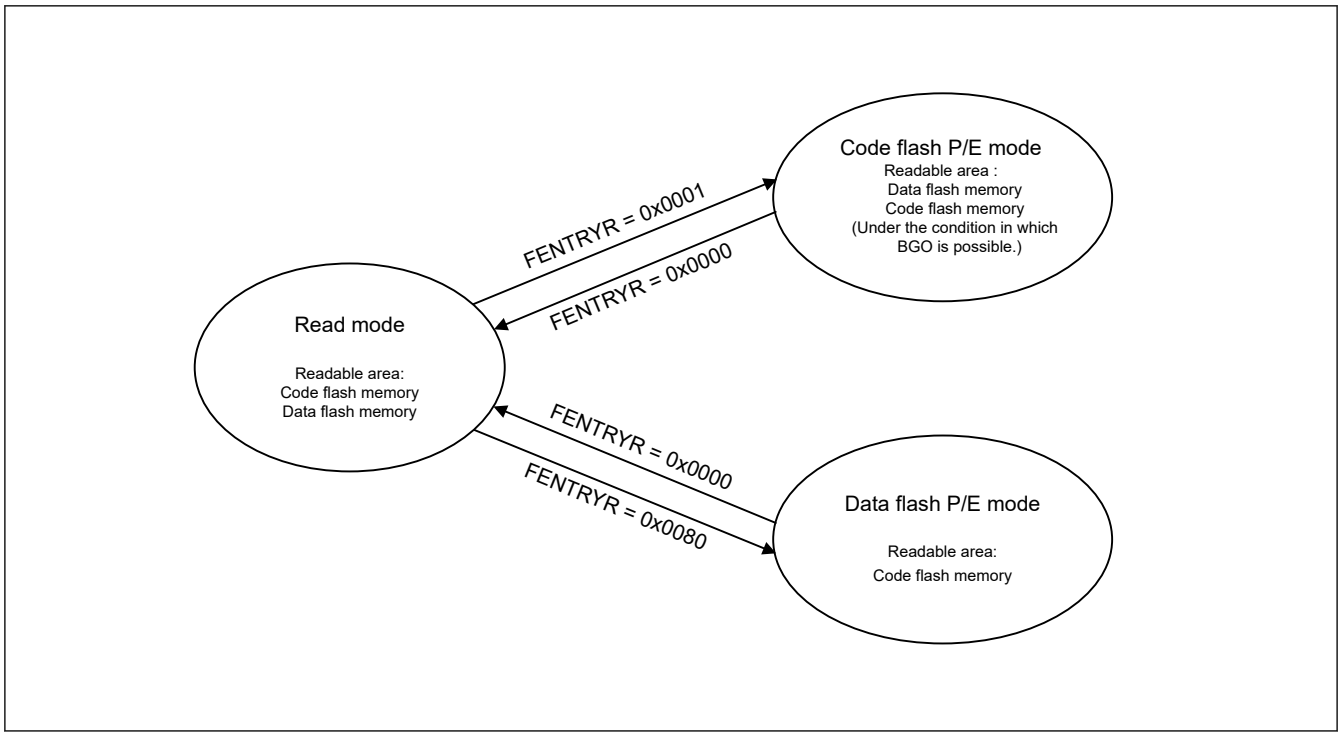


Figure 52.7 Modes of the flash sequencer

52.9 FACI Commands

52.9.1 List of FACI Commands

The FACI controls the FCU according to the specified FACI commands.

This section describes information about the FACI commands and Table 52.17 lists the FACI commands.

Table 52.17 FACI commands (1 of 2)

FACI command	Function
Program	Programs the user area and data area. Units of programming are 128 bytes for the user area and 4, 8, or 16 bytes for the data area.
Block erase	Erases user area and data area. The erase unit is 8 KB or 32 KB for user area, and 64 bytes for data flash.
Multi block erase	Erases data area. The erase unit is 64, 128, 256 bytes for data flash.
P/E suspend	Suspends programming or erasure processing.
P/E resume	Resumes suspended programming or erasure processing.
Status clear	Initializes the ILGLERR, ERSERR, PRGERR, ILGCOMERR, FESETERR, SECERR, and OTERR bits in the FSTATR register and the CMDLK, CFAE, OTERR and TZFERR bits in the FASTAT register, and the flash sequencer released from command-locked state.

Table 52.17 FACI commands (2 of 2)

FACI command	Function
Forced stop	Forcibly stops processing of FACI commands and initializes the FSTATR and FASTAT registers.
Blank check	Checks if data areas are blank. Units of Blank Check: 4 bytes to data flash memory capacity (specified in 4-byte units).
Configuration set	Sets the option-setting memory. Units of setting: • 16 bytes (Code flash memory) • 4/16 bytes (Data flash memory)
Increment Counter (Secure access only)	Increments Anti-rollback counter value . The target of counter is selected by FCNTSELR register. Update data unit is 1bit.
Refresh Counter recovery flow (Secure access only)	This is used to anti-rollback counter recovery flow. The target of counter is selected by FCNTSELR register. This command can refresh anti-rollback counter area without increasing counter value.
Read Counter (Secure access only)	Read Anti-rollback counter value. The target of counter is selected by FCNTSELR register. Reading data is output to FCNTDATAR0 and FCNTDATAR1 registers. Read unit is 64bit.

The FACI commands are issued by writing to the FACI command-issuing area (see Table 52.4). When write access as shown in Table 52.18 proceeds in the specified state, the flash sequencer executes the processing associated with the given command (see section 52.9.2. Relationship between the Flash Sequencer State and FACI Commands).

Table 52.18 FACI command formats (1 of 2)

FACI commands	Number of write access	Write data to the FACI command-issuing area			
		1st access	2nd access	3rd to (N+2)th access	(N+3)th access
Program (user area) N = 64	67	0xE8	0x40 (=N)	WD1 to WD64	0xD0
Program (data area) 4-byte programming: N = 2 8-byte programming: N = 4 16-byte programming: N = 8	N+3	0xE8	0x02 (=N) 0x04 (=N) 0x08 (=N)	WD1 to WDN	0xD0
Block Erase (user area 8K/32K Bytes)	2	0x20	0xD0	—	—
Block Erase (data area 64 bytes)	2	0x20	0xD0	—	—
Multi block erase (data area 64/128/256 bytes)	2	0x21	0xD0	—	—
P/E suspend	1	0xB0	—	—	—
P/E resume	1	0xD0	—	—	—
Status Clear	1	0x50	—	—	—
Forced Stop	1	0xB3	—	—	—
Blank Check	2	0x71	0xD0	—	—
Configuration set (Code flash memory) N = 8	11	0x40	0x08 (=N)	WD1 to WD8	0xD0
Configuration set (Data flash memory) N = 2 (4 Bytes) N = 8 (16 Bytes)	N+3	0x40	0x02 (=N) 0x08 (=N)	WD1 to WDn	0xD0
Increment Counter	2	0x35	0xD0	—	—

Table 52.18 FACI command formats (2 of 2)

FACI commands	Number of write access	Write data to the FACI command-issuing area			
		1st access	2nd access	3rd to (N+2)th access	(N+3)th access
Refresh Counter	2	0x37	0xD0	—	—
Read Counter	2	0x39	0xD0	—	—

Note: WDN (N = 1, 2, …): Nth 16-bit data to be programmed.

The flash sequencer clears the FSTATR.FRDY bit to 0 at the start of a command processing other than the Status Clear command, and sets this bit to 1 on completion.

If the FRDYIE.FRDYIE bit setting is 1, a flash ready (FRDY) interrupt is generated when the FSTATR.FRDY bit is set to 1.

52.9.2 Relationship between the Flash Sequencer State and FACI Commands

The FACI commands are accepted according to the mode/state of the flash sequencer. FACI commands should be issued after transitioning of the flash sequencer to the code flash P/E mode or data flash P/E mode and after checking the state of the flash sequencer.

Use the FSTATR and FASTAT registers to check the state of the flash sequencer. In addition, the occurrence of errors in general can be checked by reading the CMDLK bit in the FASTAT register. The value of the CMDLK bit is the logical OR of the following bits in the FSTATR register:

- ILGLERR
- ILGCOMERR
- FESETERR
- SECERR
- OTERR
- TZFERR
- ERSERR
- PRGERR
- FLWEERR.

Table 52.19 lists the available FACI commands in each operating mode.

Table 52.19 Operating mode and available FACI commands

Operating mode	FENTRYR	Available FACI commands
Read mode	0x0000	None
Code flash P/E mode	0x0001	Program Block erase P/E suspend P/E resume Status Clear Forced Stop Configuration set
Data flash P/E mode	0x0080	Program Block erase Multi block erase P/E suspend P/E resume Status Clear Forced Stop Blank Check Configuration set Increment Counter Refresh Counter Read Counter

Table 52.20 shows the state of the flash sequencer and acceptable FACI commands. An appropriate mode is assumed to have been set before the commands are executed.

Table 52.20 Acceptable FACI commands and state of the flash sequencer

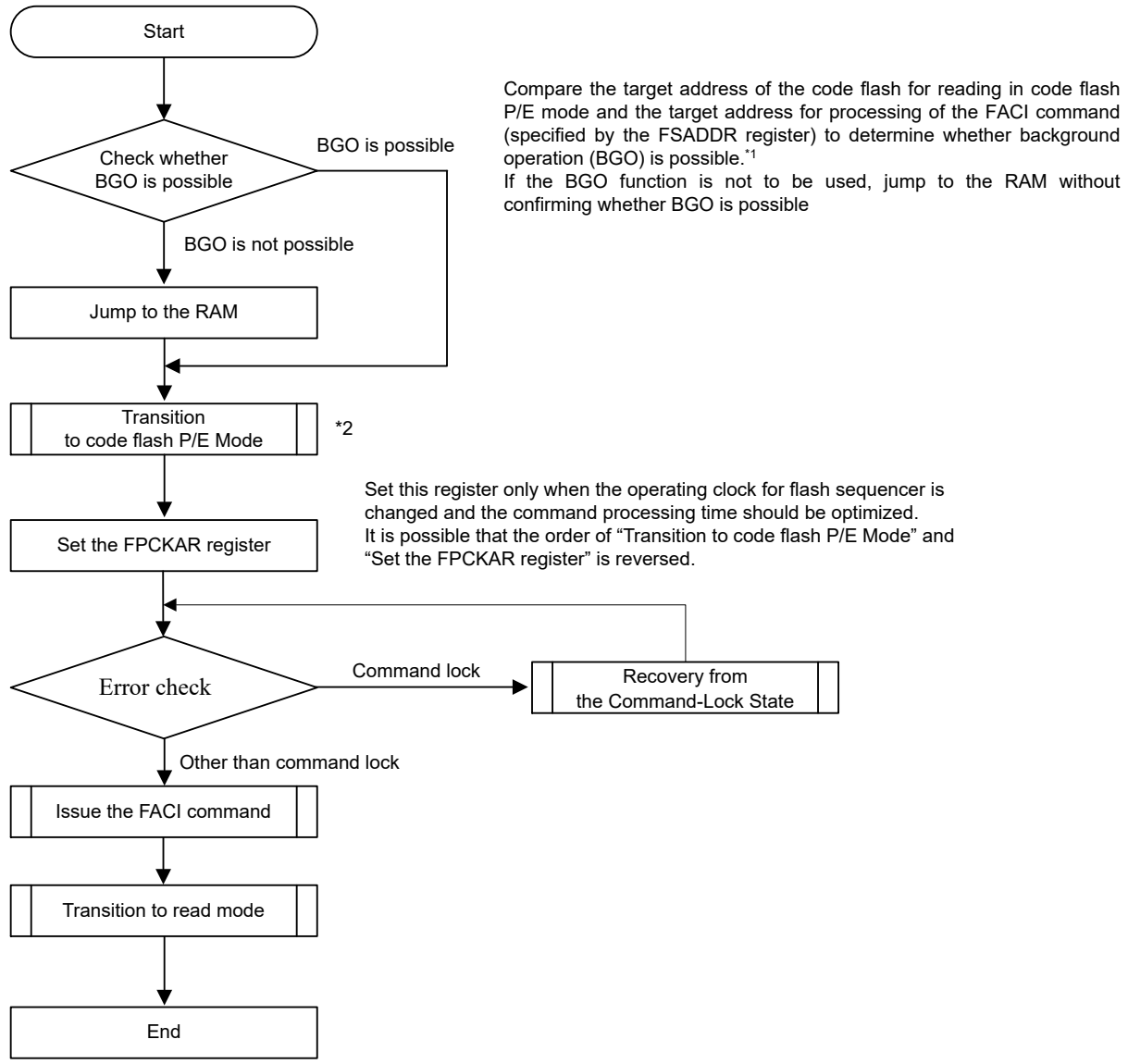
	Program, block erase or multi block erase command processing	Configuration set, configuration clear, increment counter, refresh counter or read counter command processing	Program, block erase or multi block erase command suspension processing	Blank check command processing	Programming suspended	Erasure suspended	Programming while erasure is suspended	Command-locked state (FRDY = 1)	Command-locked state (FRDY = 0)	Processing of forced stop command	Other state
FRDY bit	0	0	0	0	1	1	0	1	0	0	1
SUSRDY bit	1	0	0	0	0	0	0	0	0	0	0
ERSSPD bit	0	0	0/1	0/1	0	1	1	0/1	0/1	0	0
PRGSPD bit	0	0	0/1	0/1	1	0	0	0/1	0/1	0	0
CMDLK bit	0	0	0	0	0	0	0	1	1	0	0
Program	X	X*3	X	X	X	O*2	X	X	X	X	O
Block erase or multi block erase	X	X*3	X	X	X	X	X	X	X	X	O
P/E suspend	O	X*3	X	X	X	X	X	—	X	X	—
P/E resume	X	X*3	X	X	O	O	X	X	X	X	X
Status clear	X	X*3	X	X	O	O	X	O	X	X	O
Forced stop	O	O*3	O	O	O	O	O	O	O	O	O
Blank check	X	X*3	X	X	O*1	O*1	X	X	X	X	O*1
Configuration set	X	X*3	X	X	X	X	X	X	X	X	O
Increment Counter	X	X*3	X	X	X	X	X	X	X	X	O*1
Refresh Counter	X	X*3	X	X	X	X	X	X	X	X	O*1
Read Counter	X	X*3	X	X	X	X	X	X	X	X	O*1

Note: O: Acceptable
X: Not acceptable (places the sequencer in the command-locked state)
—: Ignored
Note 1. Only acceptable in data flash P/E mode.
Note 2. Acceptable when programming area is other than erase suspending block.
Note 3. When configuration set is processing and when FSTATR.DBFULL bit is 1, do not issue this command.

52.9.3 Usage of FACI Commands

52.9.3.1 Overview of Command Usage in Code Flash P/E Mode

Figure 52.8 show an overview of FACI command usage in code flash P/E mode. For the available commands in code flash P/E mode, see Table 52.19.

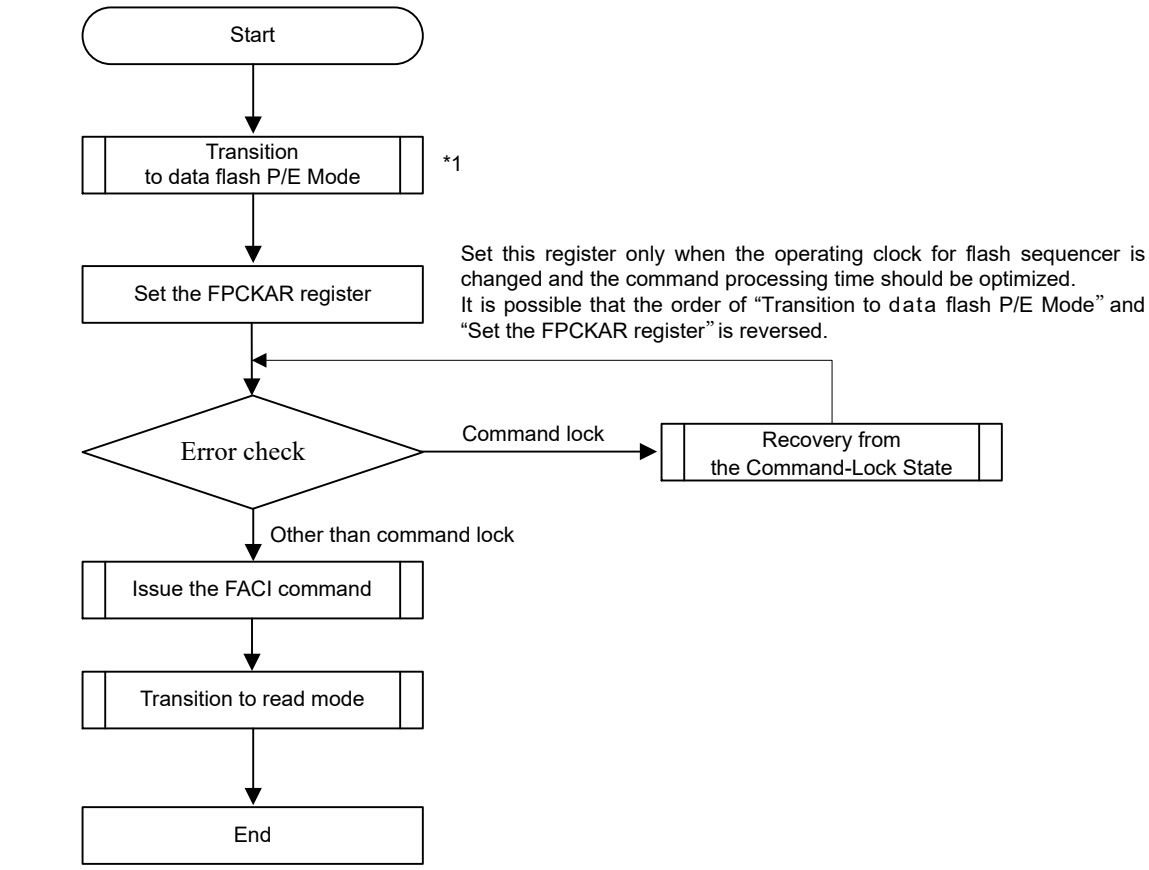


Note 1. See section 52.15.2. Background Operation
Note 2. For the condition in which BGO is possible, see section 52.9.3.3. Transition to Code Flash P/E Mode

Figure 52.8 Overview of command usage in code flash P/E mode

52.9.3.2 Overview of Command Usage in Data Flash P/E Mode

Figure 52.9 shows an overview of FACI command usage in data flash P/E and Table 52.19 lists the available commands in data flash P/E mode.



Note 1. See section 52.9.3.4. Transition to Data Flash P/E Mode

Figure 52.9 Overview of command usage in data flash P/E mode

52.9.3.3 Transition to Code Flash P/E Mode

To issue FACI commands for the code flash memory, a transition to code flash P/E mode is required by setting the FENTRYC bit in the FENTRYR register to 1.

Figure 52.10 shows the procedure to transition to code flash P/E mode.

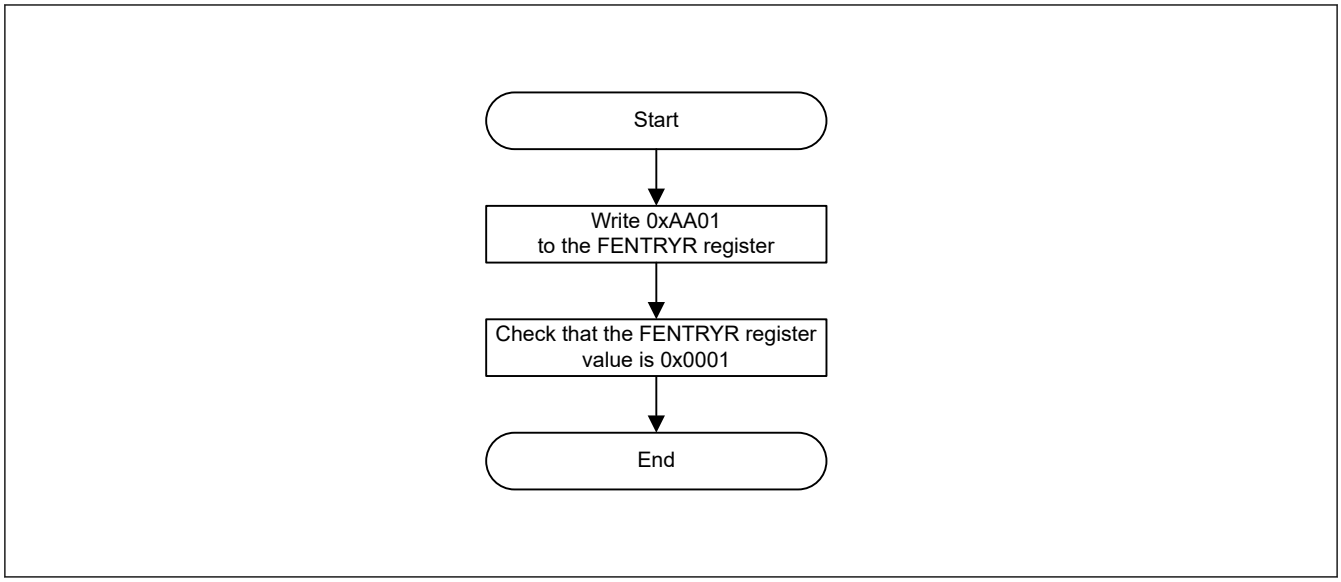


Figure 52.10 Procedure to transition to code flash P/E mode

52.9.3.4 Transition to Data Flash P/E Mode

To issue FACI commands for the data flash memory, a transition to data flash P/E mode is required by setting the FENTRYD bit in the FENTRYR register to 1.

Figure 52.11 shows the procedure to transition to data flash P/E mode.

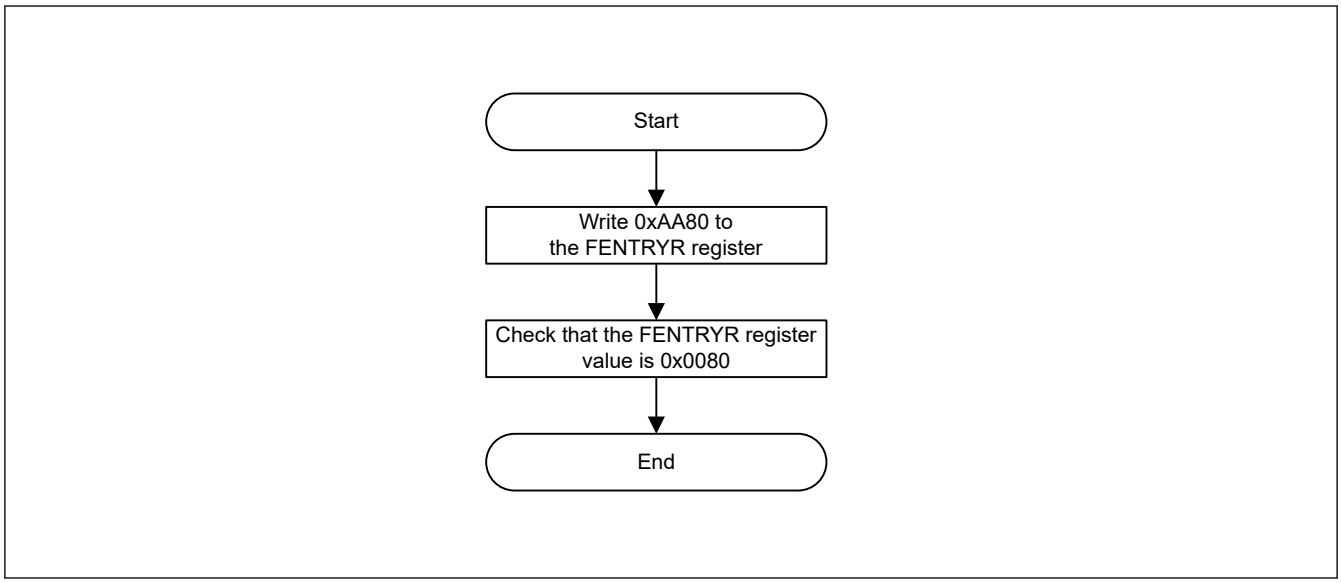
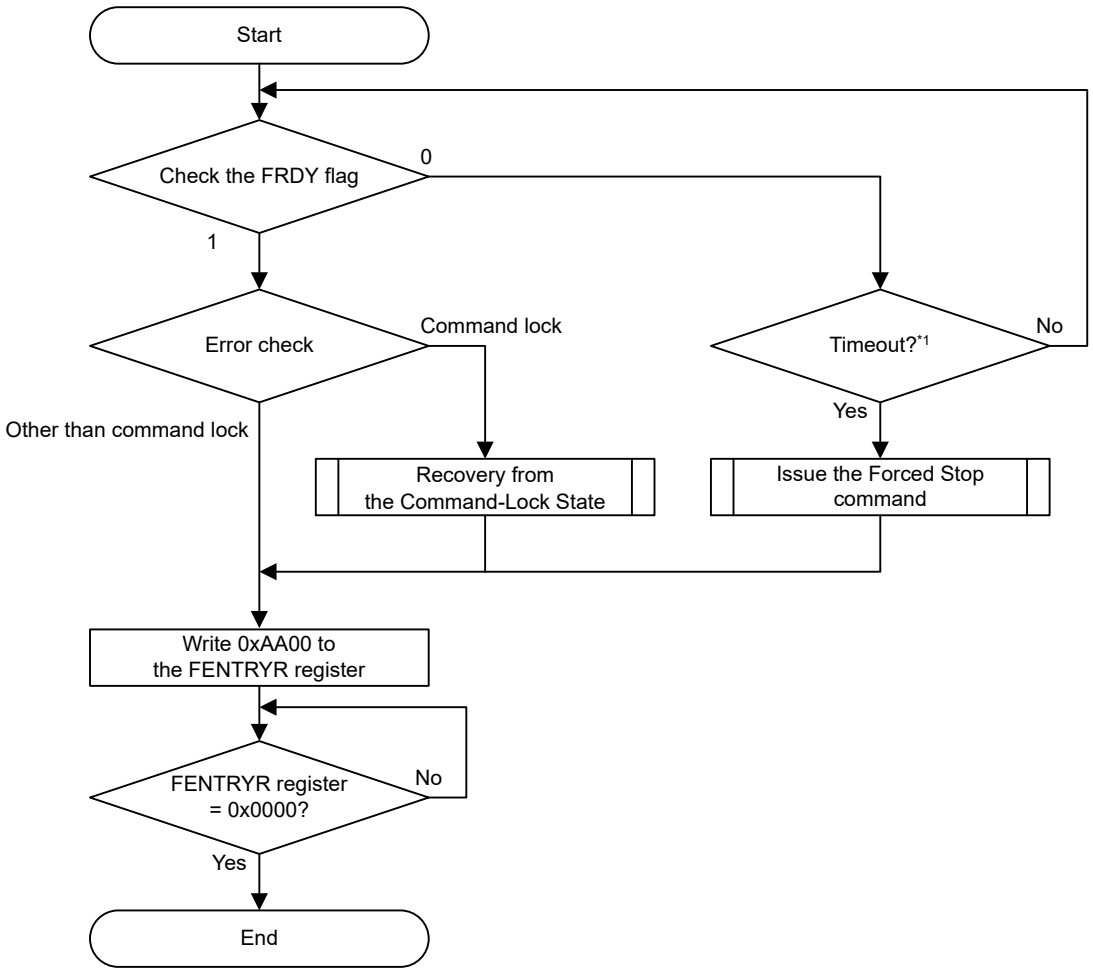


Figure 52.11 Procedure to transition to data flash P/E mode

52.9.3.5 Transition to Read Mode

To read the flash memory, a transition to read mode is required by setting the FENTRYR register to 0x0000. The transition to read mode should be made after the flash sequencer completes the processing and while operation is not in the command-locked state.

Figure 52.12 shows the procedure to transition to read mode.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time for processing of the FACI command that is in progress (see section 60, Electrical Characteristics)

Figure 52.12 Procedure to transition to read mode

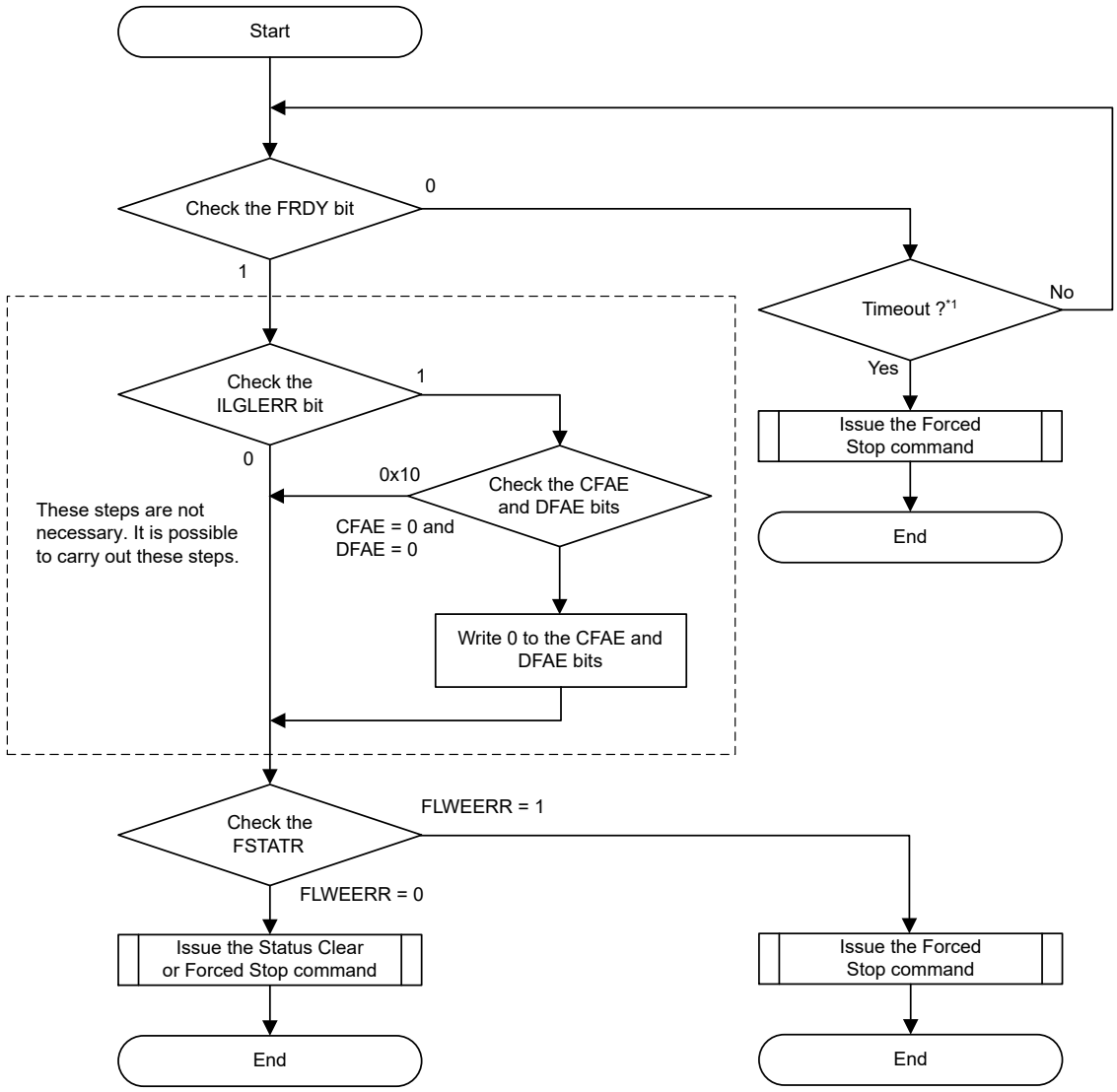
52.9.3.6 Recovery from the Command-Locked State

When the flash sequencer enters the command-locked state, FACI commands cannot be accepted. To release the sequencer from the command-locked state, use the status clear command, forced stop command, or FASTAT register.

When the command-locked state is detected by checking for an error before issuing the P/E suspend command, the FRDY bit in the FSTATR register might be 0 even though command processing has not completed. If processing is not complete by the maximum programming/erasure time specified in the electrical characteristics, this is a timeout and the flash sequencer must be stopped with the forced stop command.

The FLWEERR bit in the FSTATR register does not change from 1 to 0 with the status clear command. When these bits are set to 1, use the forced stop command to release from the command-locked state. Bits other than FRDY and FLWEERR in FSTATR register that indicate the command-locked state can be changed from 1 to 0 with the status clear or forced stop command.

Figure 52.13 shows the recovery flow from the command-locked state.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time for processing of the FACI command that is in progress (see section 60, Electrical Characteristics)

Figure 52.13 Recovery flow from the command-locked state

52.9.3.7 Program Command

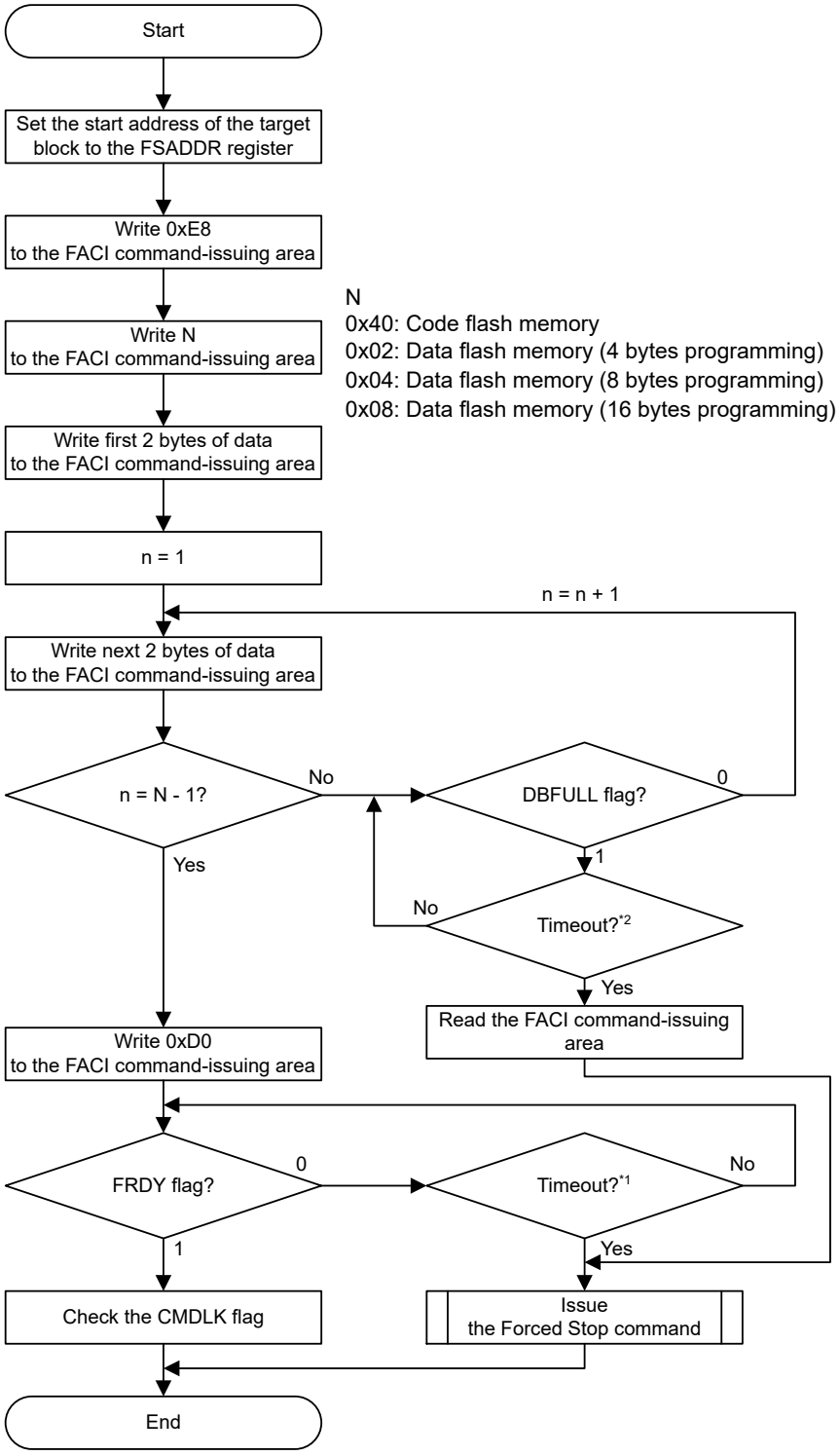
Program commands are used to write to the user and data areas. Before issuing the FACI program command, set the first address of the target block in the FSADDR register.

Program processing is started by writing the last 16 bits of program data (WD64 for code flash and WD2, WD4, and WD8 for data flash in Table 52.18) to the FACI command-issuing area before 0xD0, the final value of the FACI command, is written. Therefore, if it takes time from writing the last 16 bits of program data to writing the final FACI command value 0xD0 (for example, due to interrupt processing), the FSTATR.FRDY bit is set to 1 when program processing is completed even if 0xD0 has not been written. In this case, the program processing has been completed, but the FACI command reception has not been completed. If a subsequent FACI command is issued in this state, an illegal command error will occur. To avoid illegal command errors, disable interrupts with processing that issues FACI commands while program commands are issued (from writing 0xE8 to writing 0xD0).

If the target area for program command processing includes an area that is not for writing, write 0xFFFF to the corresponding area. If a program command is issued while the FACI internal data buffer is full, a wait period may occur on

the peripheral bus, which may affect the communication performance of other peripheral modules. To avoid the occurrence of a waiting period, set the DBFULL bit in the FSTATR register to 0 when issuing a FACI command. Writing to the data area will not cause the data buffer to become full.

Figure 52.14 shows the usage of the program command.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time of programming. See section 60, Electrical Characteristics about those time

Note 2. 1.4 µs (f_{FCLK} ≥ 20 MHz)

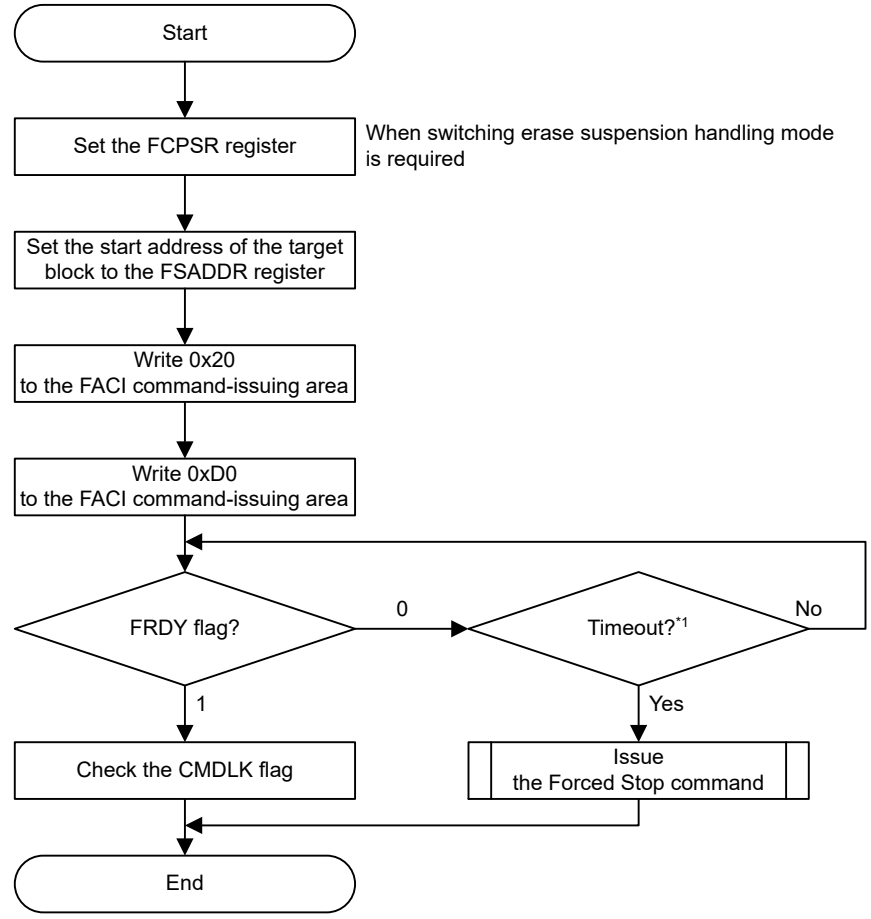
Figure 52.14 Usage flow of the program command

52.9.3.8 Block Erase Command

The block erase command is used for erasing user area or data area. The erase unit is one block. Before issuing a block erase command, set the first address of the target block to FSADDR register. Writing 0xD0 at the second write access of the FACI command triggers the FACI to start the block erase command processing. Completion of command processing can be confirmed with the FRDY bit of FSTATR register.

Set the FCPSR registers before issuing the block erase command. Additionally, FCPSR must be set when the erasure-suspended mode is to be switched.

Figure 52.15 shows the usage of the block erase command.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time of erasure. See section 60, Electrical Characteristics.

Figure 52.15 Usage flow of the block erase command

52.9.3.9 Multi Block Erase Command

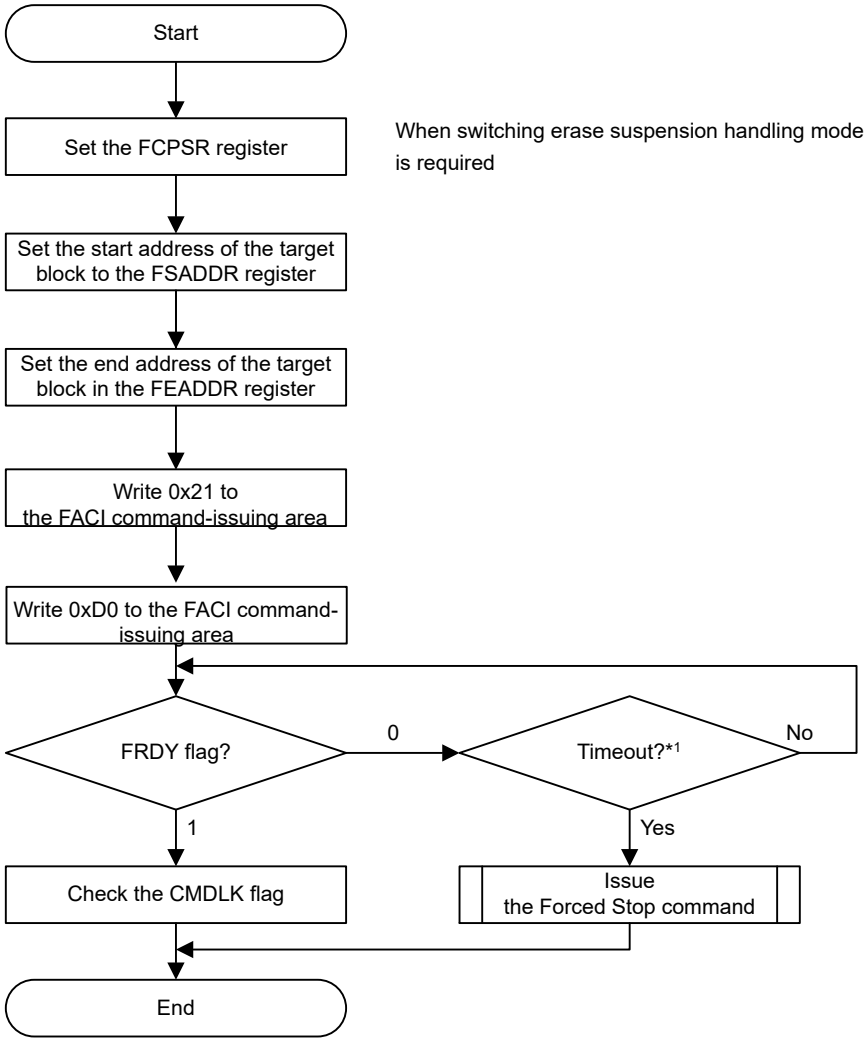
The multi block erase command is used for erasing data area. The erase unit is 64, 128, or 256 bytes. Before issuing the multi block erase command, set the start address to FSADDR register and the end address to FEADDR register. Writing 0xD0 at the second write access of the FACI command triggers FACI to start the multi block erase command processing. Completion of command processing can be confirmed with the FRDY bit of FSTATR register.

Set the FCPSR registers before issuing the multi block erase command. Additionally, FCPSR must be set when the erasure-suspended mode is to be switched.

The erase size is specified by both the FSADDR and FEADDR settings. Table 52.21 describes how to set the FSADDR and FEADDR.

Table 52.21 Settings for the erase size

Erase size	FSADDR	FEADDR
64 bytes	FSA0 to FSA5 = 0 (64 byte-boundary)	FSADDR + 0x3C
128 bytes	FSA0 to FSA6 = 0 (128 byte-boundary)	FSADDR + 0x7C
256 bytes	FSA0 to FSA7 = 0 (256 byte-boundary)	FSADDR + 0xFC



Note 1. Judgment of the timeout is based on 1.1 times the maximum time of erasure. See section 60, Electrical Characteristics.

Figure 52.16 Usage flow of the multi block command

52.9.3.10 P/E Suspend Command

The P/E suspend command is used to suspend programming/erasure. Before issuing a P/E suspend command, check that the CMDLK bit in the FASTAT register is 0, and that the execution of programming/erasure is performed normally. To confirm that the P/E suspend command can be received, check that the SUSRDY bit in the FSTATR register is 1. After issuing a P/E suspend command, read the CMDLK bit to confirm that no error occurs.

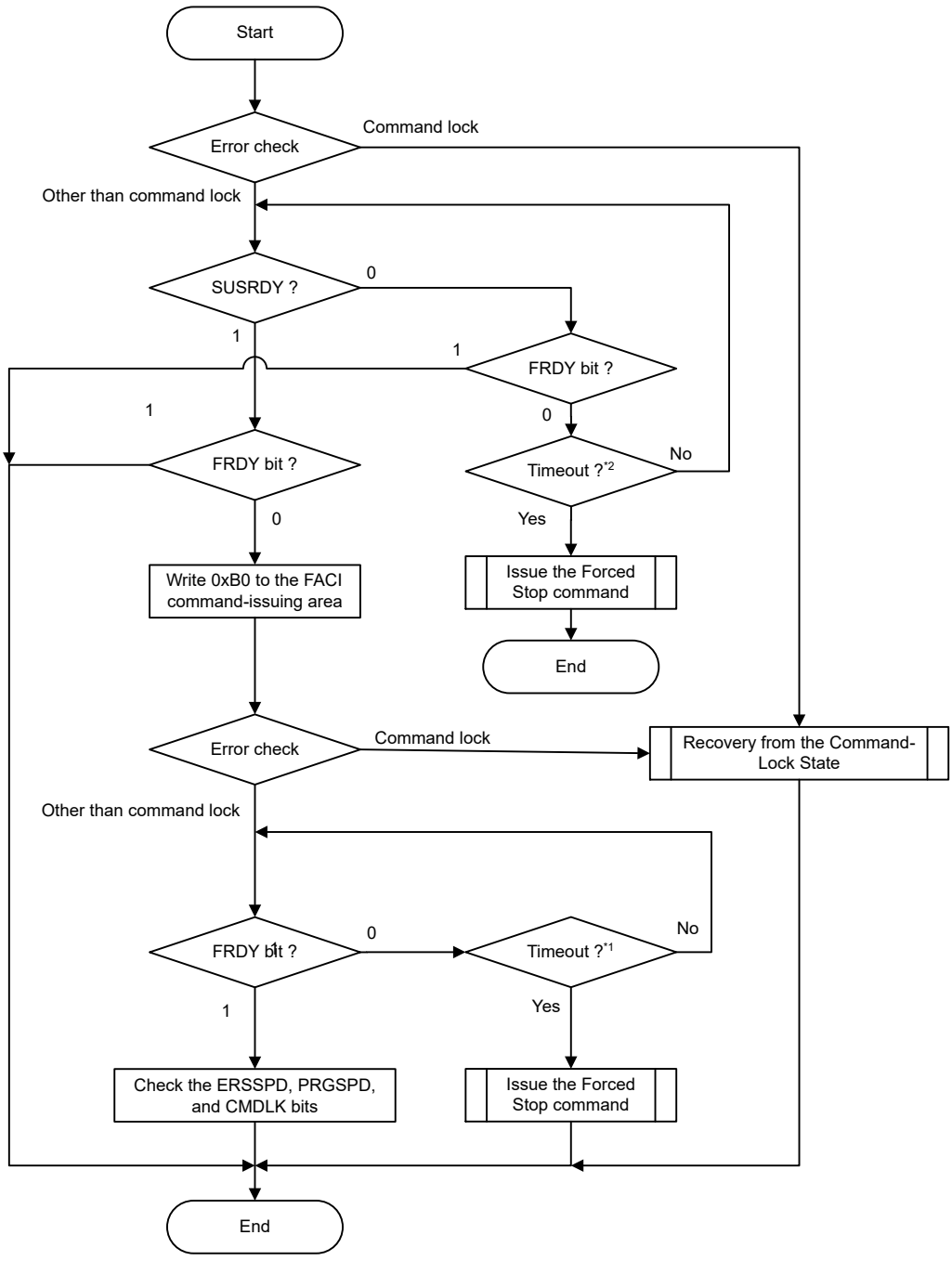
If an error occurs during programming/erasure, the CMDLK bit is set to 1. When programming/erasure processing has finished from the time when the SUSRDY bit is 1 to when the P/E suspend command is received, no error occurs and the

suspended state is not entered (the FRDY bit in the FSTATR register is 1 and the ERSSPD and PRGSPD bits in FSTATR are 0).

When a P/E suspend command is received and the programming/erasure suspend processing finishes normally, the flash sequencer enters the suspended state, the FRDY bit is set to 1, and the ERSSPD or PRGSPD bit is 1. After issuing a P/E suspend command, check that the ERSSPD or PRGSPD bit is 1 and the suspended state is entered, then proceed with the subsequent flow. If a P/E resume command is issued in the subsequent flow even when the suspended state is not entered, an illegal command error occurs and the flash sequencer shifts to the command-locked state (see section 52.11.2. Error Protection).

If the erasure suspended state is entered, programming to blocks other than an erasure target block can be performed. Additionally, the programming and erasure suspended states can shift to read mode by clearing the FENTRYR register.

Figure 52.17 shows the usage of the P/E suspend command.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time of suspend delay during erasure for both code flash and data flash.
See section 60, Electrical Characteristics.
Note 2. 4.4µs (f_{FCLK} >= 20MHz)

Figure 52.17 Usage flow of the P/E suspend command

(1) Suspension during Programming

When issuing a P/E suspend command during flash memory programming, the flash sequencer suspends programming processing. Figure 52.18 shows the suspend programming operation. When receiving programming-related command, the flash sequencer clears the FRDY bit in the FSTATR register to 0 to start programming. If the flash sequencer enters the

state in which the P/E suspend command can be received after programming starts, it sets the SUSRDY bit in the FSTATR register to 1.

When a P/E suspend command is issued, the flash sequencer receives the command and clears the SUSRDY bit to 0. If the flash sequencer receives a P/E suspend command while a programming pulse is applied, the flash sequencer continues with the pulse. After the specified pulse application time, the flash sequencer finishes pulse application, starts the programming suspend processing, and sets the PRGSPD bit in the FSTATR register to 1.

When a suspended processing finishes, the flash sequencer sets the FRDY bit to 1 to enter the programming suspended state. When receiving a P/E resume command in the programming suspended state, the flash sequencer clears the FRDY and PRGSPD bits to 0 and resumes programming.

Figure 52.18 shows the timing for suspension during programming.

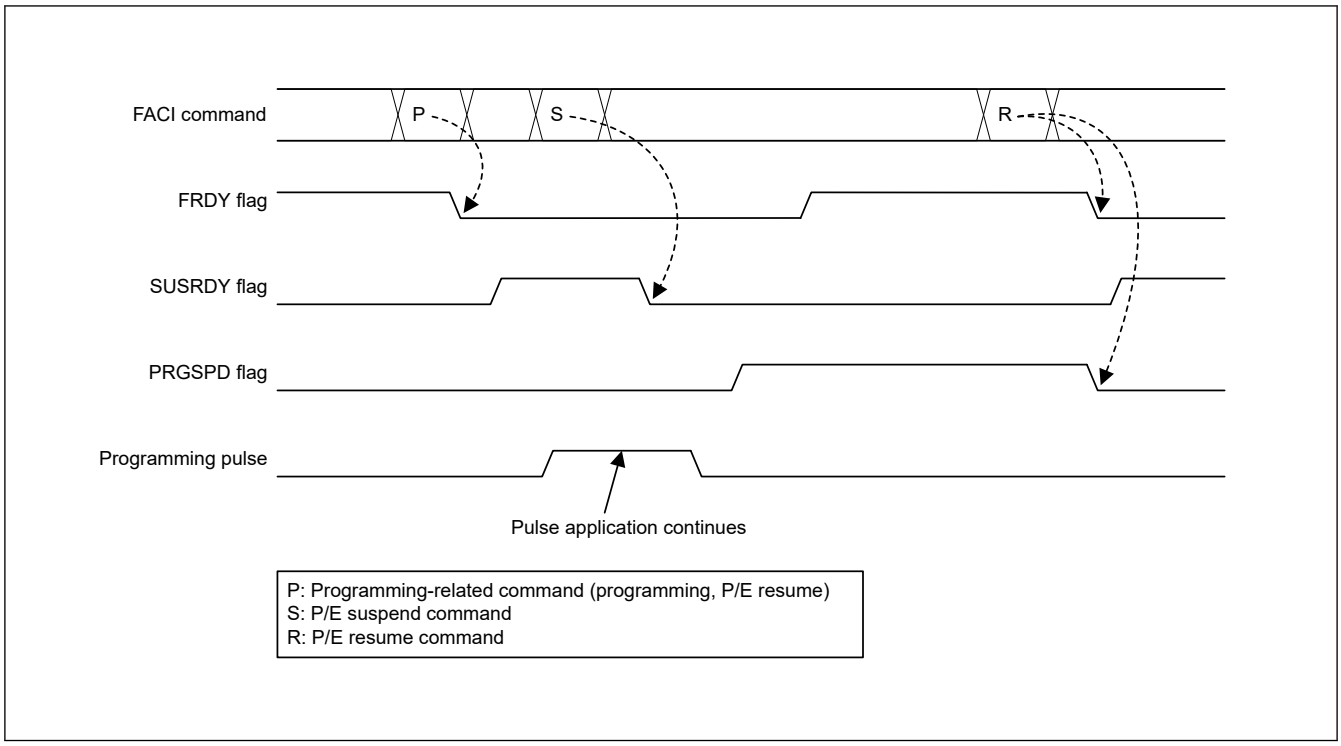


Figure 52.18 Suspension during programming

(2) Suspension during Erasure (Suspension Priority Mode)

The flash sequencer has a suspension priority mode for the suspension of erasure. Figure 52.19 shows the suspend operation of erasure when the erasure suspend mode is set to the suspension priority mode (the ESUSPMD bit in the FCPSR register is 0).

When receiving an erasure-related command, the flash sequencer clears the FRDY bit in the FSTATR register to 0 to start erasure. If the flash sequencer enters the state in which the P/E suspend command can be received after erasure starts, it sets the SUSRDY bit in the FSTATR register to 1.

When a P/E suspend command is issued, the flash sequencer receives the command and clears the SUSRDY bit to 0.

When receiving a suspend command during erasure, the flash sequencer starts the suspend processing and sets the ERSSPD bit in the FSTATR register to 1 even when it is applying an erasure pulse. When the suspended processing finishes, the flash sequencer sets the FRDY bit to 1 to enter the erasure suspended state. When receiving a P/E resume command in the erasure suspended state, the flash sequencer clears the FRDY and ERSSPD bits to 0 and resumes erasure. Operations of the FRDY, SUSRDY, and ERSSPD bits at the suspension and resumption of erasure are the same, regardless of the erasure suspend mode.

The setting of the erasure suspend mode affects the control method of erasure pulses. In suspension priority mode, when receiving a P/E suspend command while erasure pulse A that has not been previously suspended is being applied, the flash sequencer suspends the application of erasure pulse A and enters the erasure suspended state. When receiving a P/E suspend command while reapplying erasure pulse A after erasure is resumed with a P/E resume command, the flash sequencer

continues to apply erasure pulse A. After the specified pulse application time, the flash sequencer finishes erasure pulse application and enters the erasure suspended state.

When the flash sequencer receives a P/E resume command next and erasure pulse B is being applied, the flash sequencer receives a P/E suspend command again, and the application of erasure pulse B is then suspended. In suspension priority mode, delays due to suspension can be minimized because the application of an erasure pulse is suspended once per pulse, and priority is given to the suspend processing.

If the interval of suspension after resume is longer than t_{REST1} (Resume time: priority on suspension, resume after the 1st suspend for the same pulse), suspend delay will be always t_{SESD1} (Suspend delay: priority on suspension, the 1st suspend for the same pulse).

If the interval of suspension after resume is shorter than t_{REST1}, suspend delay becomes either t_{SESD1} or t_{SESD2} (Suspend delay: priority on suspension, the 2nd suspend for the same pulse).

(The value of t_{REST1} / t_{SESD1} / t_{SESD2}, see section 60, Electrical Characteristics.)

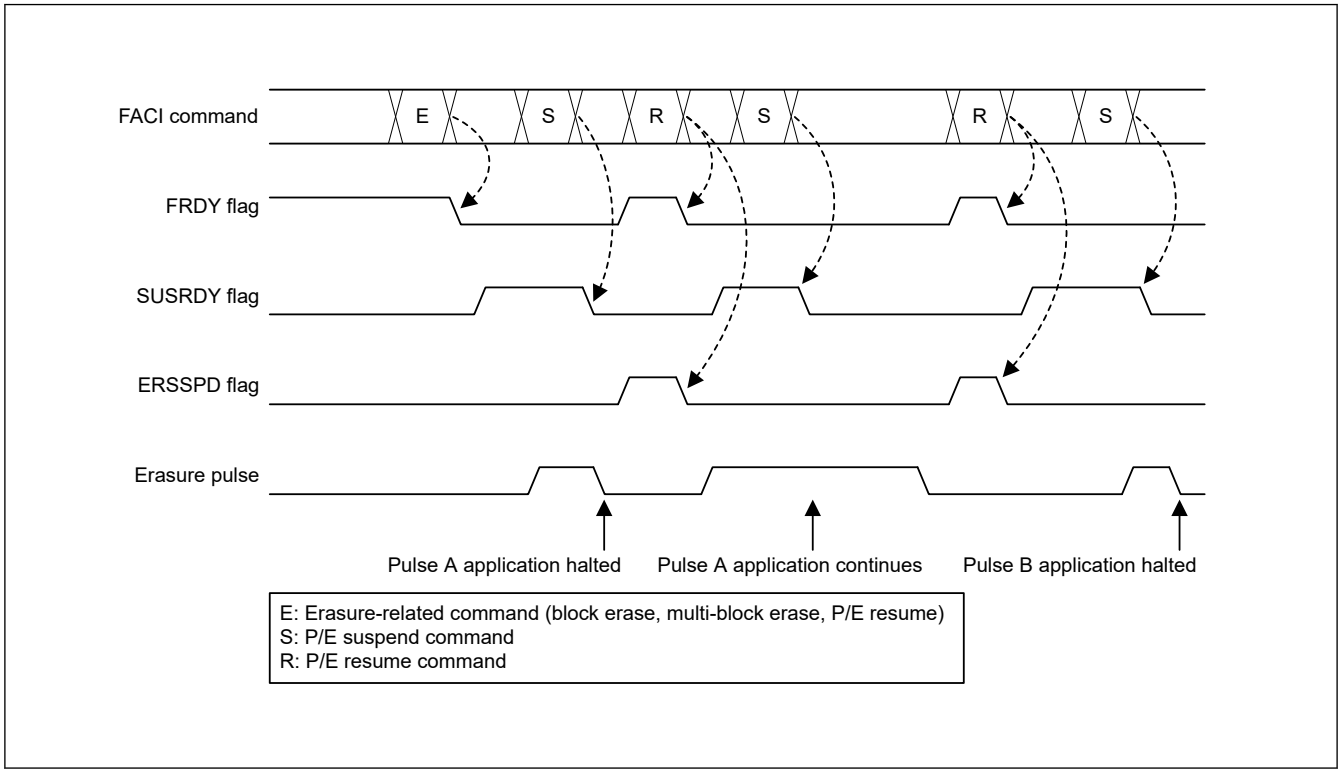


Figure 52.19 Suspension during erasure (suspension priority mode)

(3) Suspension during Erasure (Erasure Priority Mode)

The flash sequencer has an erasure priority mode for the suspension of erasure. Figure 52.20 shows the suspend operation of erasure when the erasure suspend mode is set to the erasure priority mode (the ESUSPMD bit in the FCPSR register is 1). The control method of erasure pulses in erasure priority mode is the same as that of programming pulses for the programming suspend processing.

If the flash sequencer receives a P/E suspend command while an erasure pulse is applied, the flash sequencer continues to apply the pulse. In this mode, the required time for the erasure processing can be reduced compared to the suspension priority mode because the re-application of erasure pulses does not occur when a P/E resume command is issued.

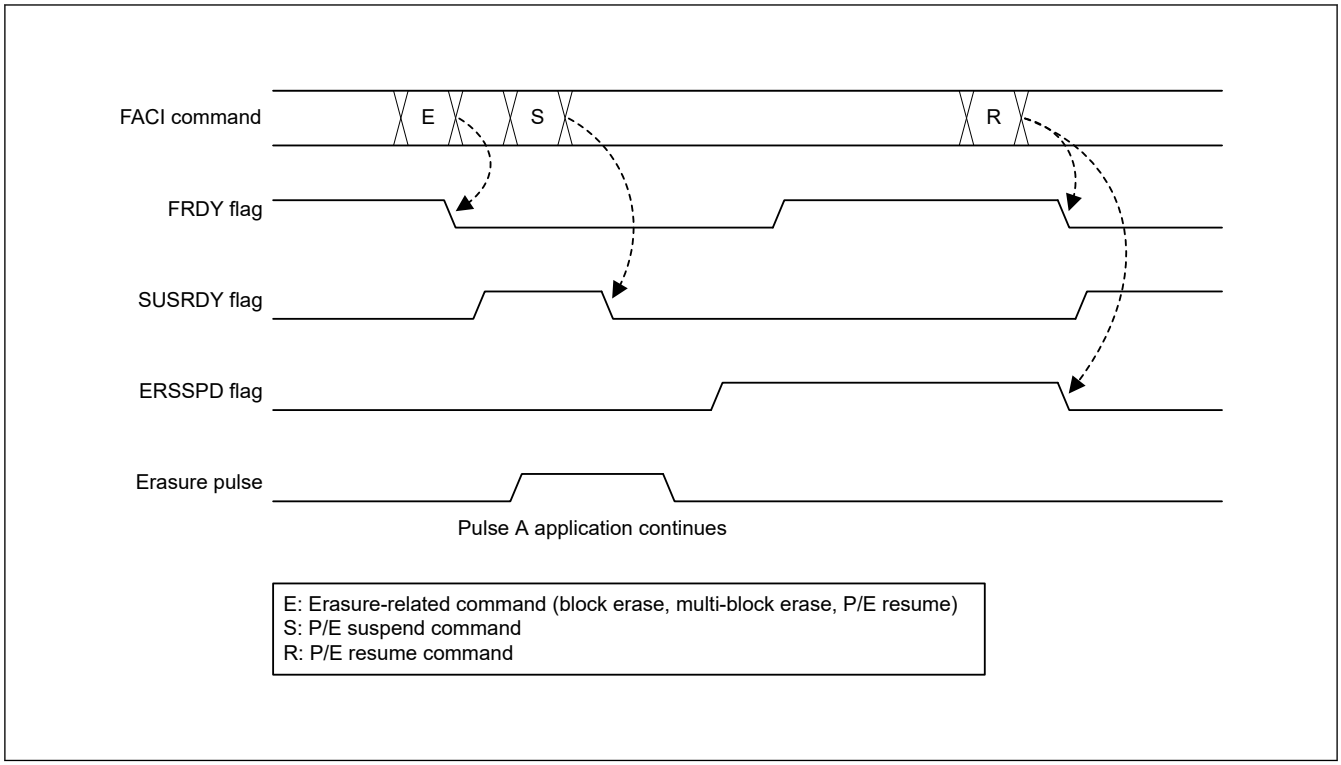
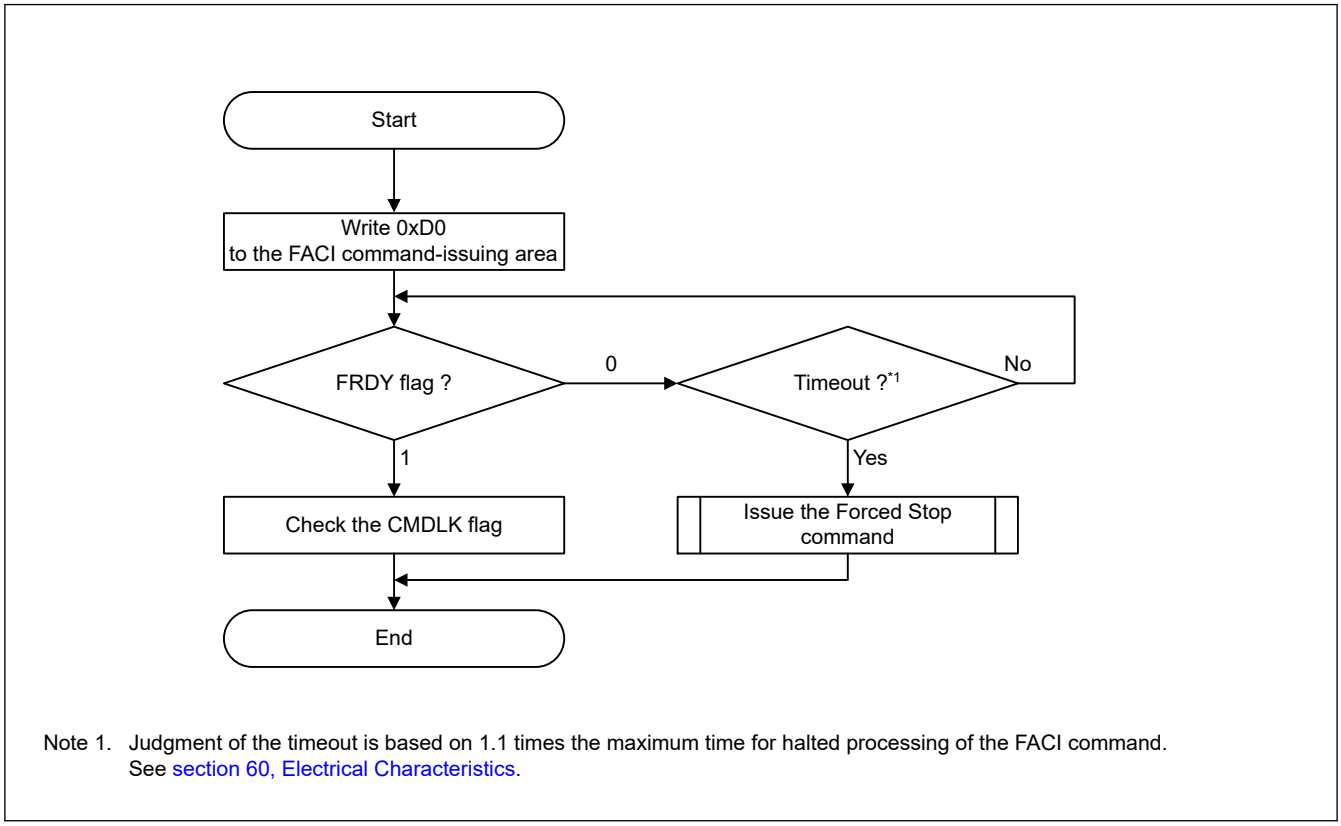


Figure 52.20 Suspension during erasure (erasure priority mode)

52.9.3.11 P/E Resume Command

The P/E resume command is used to resume suspended programming or erasure. If the FENTRYR setting has been modified during suspension, issue a P/E resume command only after resetting FENTRYR to the previous value that was held before the P/E suspend command was issued. Figure 52.21 shows usage of the P/E resume command.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time for halted processing of the FACI command. See section 60, Electrical Characteristics.

Figure 52.21 Usage flow of the P/E resume command

52.9.3.12 Status Clear Command

The status clear command is used to clear the command-locked state (see section 52.9.3.6. Recovery from the Command-Locked State).

You can use the status clear command to clear the following bits in the FSTATR register in the command-locked state:

- ILGLERR
- ILGCOMERR
- FESETERR
- SECERR
- OTERR
- ERSERR
- PRGERR
- TZFERR

Figure 52.22 shows usage of the status clear command.

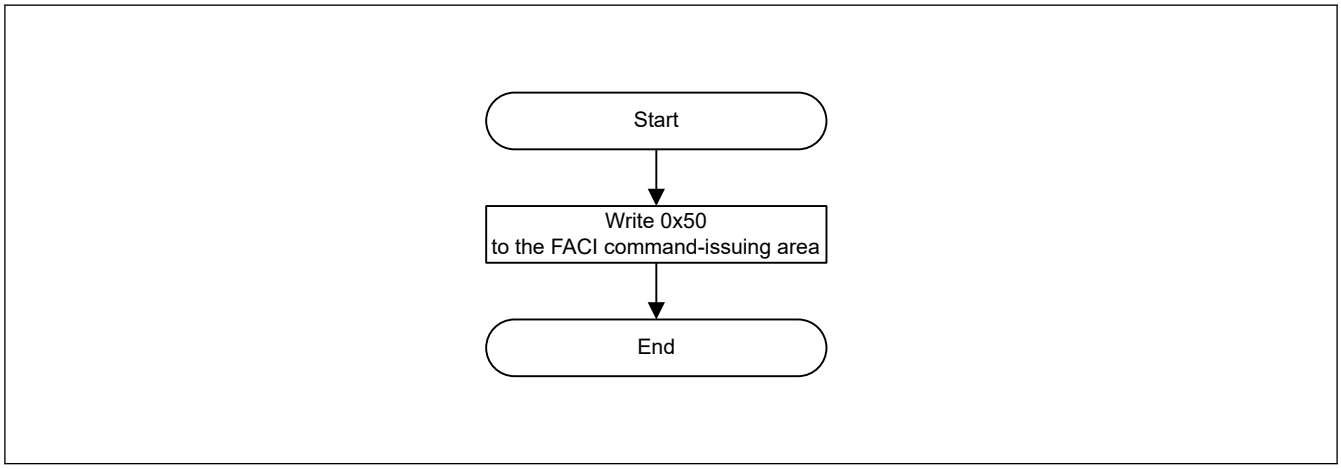


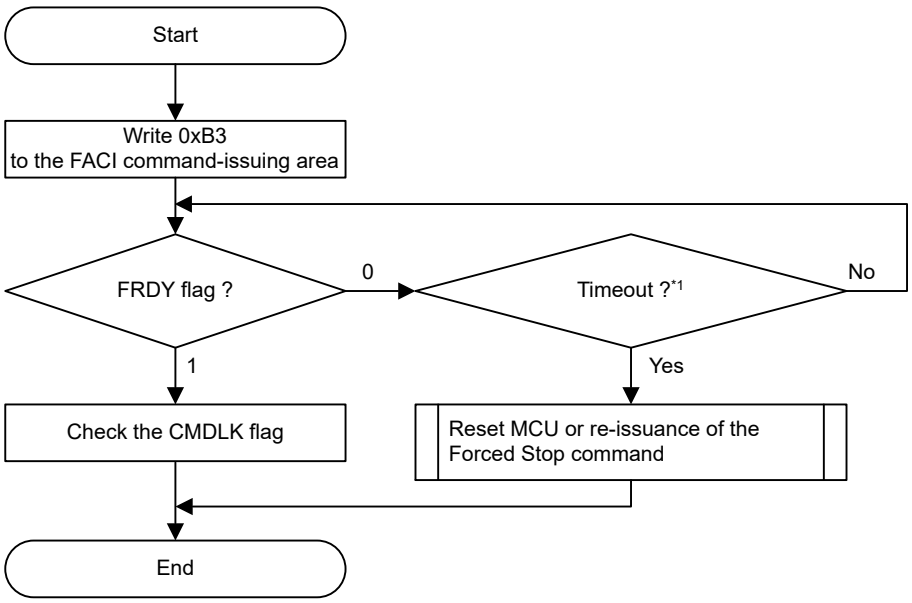
Figure 52.22 Usage flow of the status clear command

52.9.3.13 Forced Stop Command

The forced stop command is used to forcibly end command processing by the flash sequencer. Although this command halts command processing more quickly than the P/E suspension command, values from the programming or erasure that are in progress are not guaranteed. Additionally, resumption of processing is not possible. Processing of programming or erasure that is halted by the forced stop command is also defined as one programming round.

Executing the forced stop command also initializes part of the FACI, the whole FCU, the FSTATR and FASTAT registers. This command can be used in the procedure for recovery from the command-locked state and for processing in response to a timeout of the flash sequencer (see section 52.9.3.6. Recovery from the Command-Locked State).

Figure 52.23 shows usage of the forced stop command.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time of the Forced Stop command latency. See section 60, Electrical Characteristics.

Figure 52.23 Usage flow of the forced stop command

(1) Notes on Using the Forced Stop Command during Command Issue

When using the forced stop command at the timeout occurrence by DBFULL bit of the program command, writing in the FACI command-issuing area is sometimes processed as writing in data of the program command. See Table 52.4 in section 52.3. Address Space for information on the FACI command-issuing area to force a command lock. Then issue a forced stop command with return method from the command lock status (see Figure 52.14). Locking commands is possible in any case where the unit for reading the FACI command issuing area is 8, 16, or 32 bits.

52.9.3.14 Blank Check Command

The blank check command is used to confirm that an area is in the non-programmed state. Values read from the data flash memory that have been erased but not yet programmed again that is in the non-programmed state, are undefined.

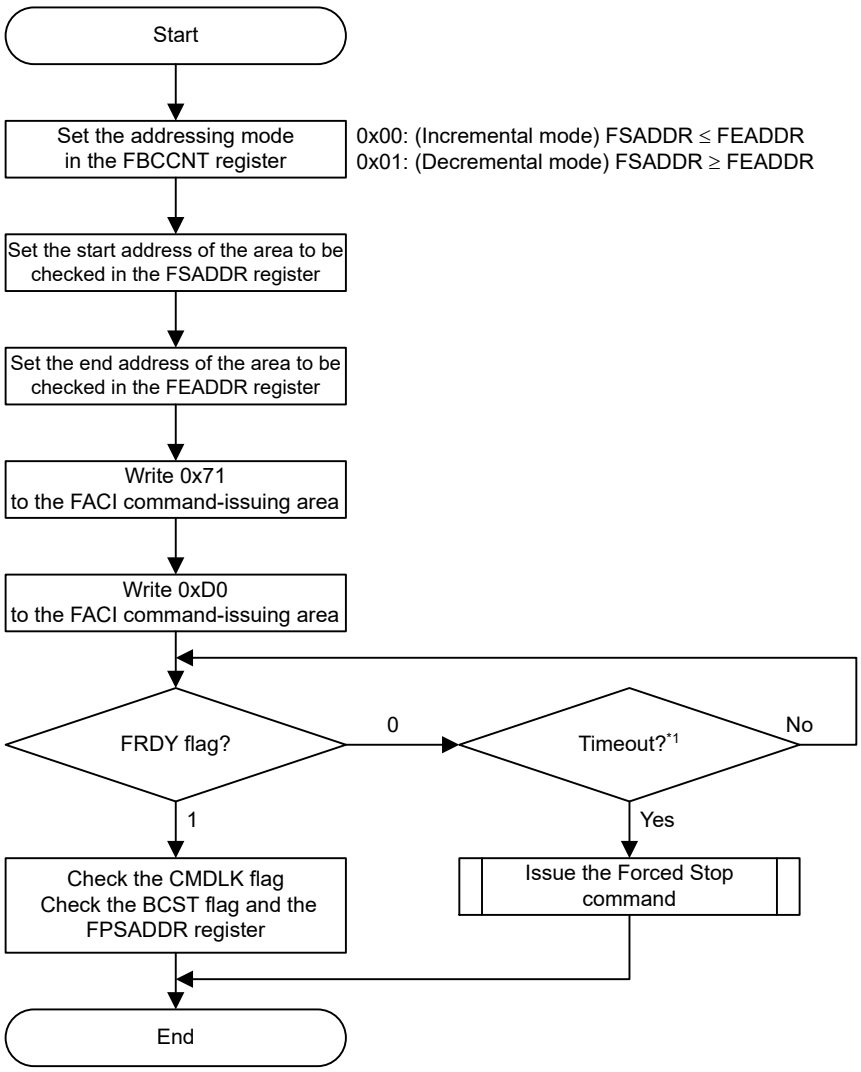
Before issuing the Blank Check command, set addressing mode, start address, and end address of the target area for Blank Check to the FBCCNT, FSADDR, and FEADDR registers. When Blank Check addressing mode is set to decremental mode (i.e. FBCCNT.BCDIR = 1), address specified in FSADDR should be equal to or larger than address in FEADDR.

On the other hand, the address in FSADDR should be equal to or smaller than address in FEADDR when Blank Check addressing mode is set to incremental mode (i.e. FBCCNT.BCDIR = 0).

If the settings of the BCDIR bit, FSADDR, and FEADDR are inconsistent, the flash sequencer enters the command-locked state. The size of the target area for Blank Check is in the range from 4 bytes to the data flash memory capacity and is set in units of 4 bytes.

Write 0x71 and 0xD0 to the FACI command-issuing area to start Blank Check. Completion of processing can be confirmed by the FRDY bit of the FSTATR register. At the end of processing, the result of Blank Check is stored in the BCST bit in the FBCSTAT register. If non-programmed data exists within the target area for Blank Check, flash sequencer stops Blank Check command operation. In this case, address of non-programmed data is indicated to FPSADDR register.

Figure 52.24 shows usage of the blank check command.



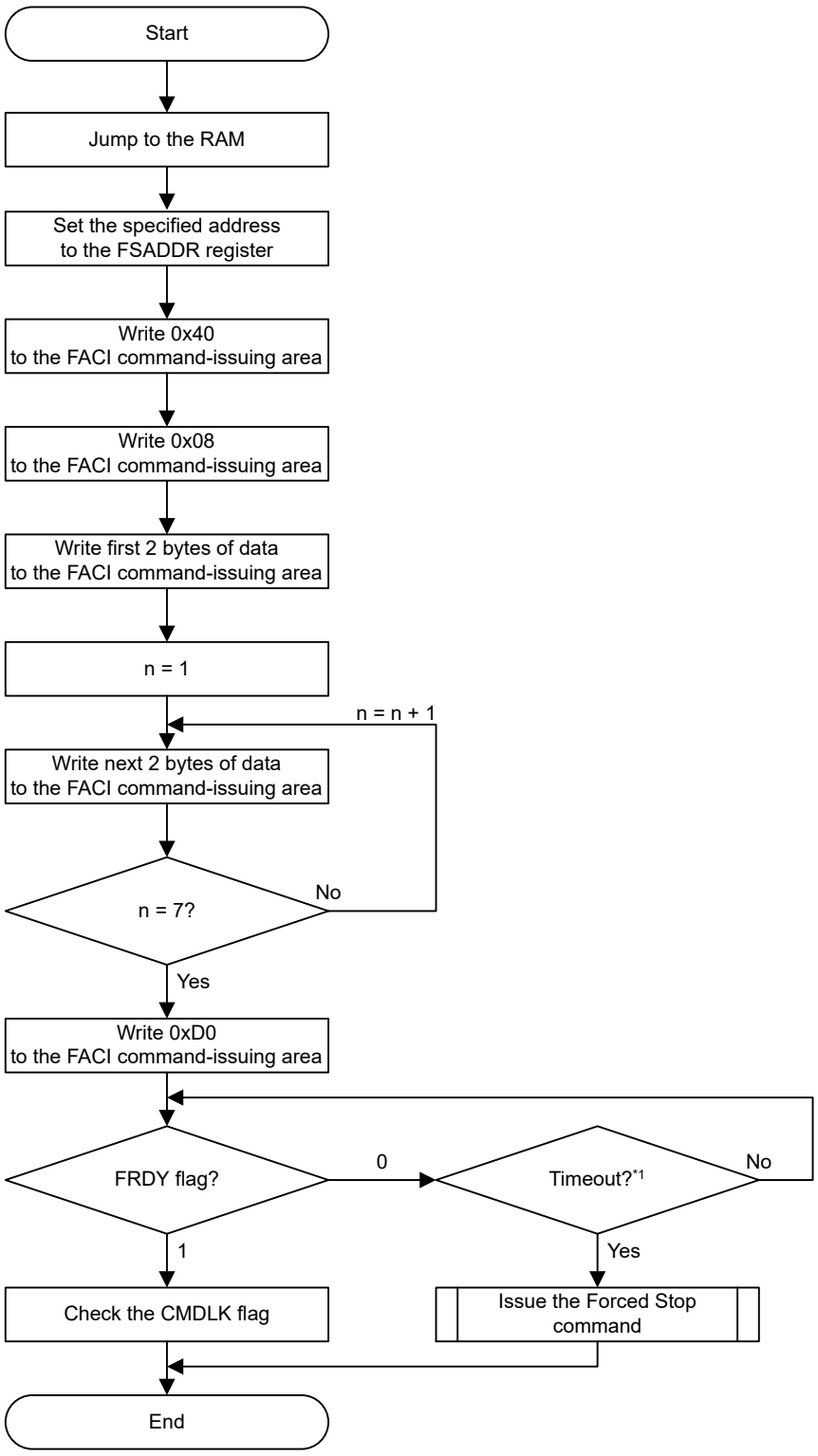
Note 1. Judgment of the timeout is based on 1.1 times the maximum time of Blank Checking for the whole target range. See section 60, Electrical Characteristics.

Figure 52.24 Usage flow of the blank check command

52.9.3.15 Configuration Set Command

The Configuration set command is used to set option-setting memory. Before issuing the Configuration set command, set the specified address (shown in Table 52.22) in the FSADDR register. Writing 0xD0 to the FACI command-issuing area in the final access for issuing the FACI command starts FACI processing of the Configuration set command.

Figure 52.25 shows usage of the configuration set command.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time of programing in option-setting memory. See section 60, Electrical Characteristics.

Figure 52.25 Usage flow of the configuration set command

The correspondence between the possible target data for configuration setting and the address value set in the FSADDR register is shown in Table 52.22. For details, see section 52.4.12. FSADDR : FACI Command Start Address Register.

Table 52.22 Address Used by Configuration Set Command for Secure alias (Code flash memory)

Address	FSADDR Register Value	Setting Data	Operation of additional writing		Timing when the Setting is Enabled
			SAS.FSPR bit is 1	SAS.FSPR bit is 0	
0x0300_A100	0x0300_A100	Option Function Select Register 0 (OFS0)	Writable	Writable	At a reset
0x0300_A104		Option Function Select Register 2 (OFS2)			
0x0300_A110	0x0300_A110	Dual Mode Select Register (DUALSEL)	Writable	Writable	At a reset
0x0300_A134	0x0300_A130	Start-up Area Setting Register (SAS)	Writable	Not writable*1	When a reset or command is executed
0x1300_A180	0x1300_A180	Option Function Select Register 1 (OFS1)	Writable	Writable	At a reset
0x1300_A190	0x1300_A190	Bank Select Register (BANKSEL)	Writable	Writable	At a reset
0x1300_A1C0	0x1300_A1C0	Block Protect Setting Register (BPS)	Writable*2	Writable*2	When a reset or command is executed
0x1300_A1E0	0x1300_A1E0	Permanent Block Protect Setting Register (PBPS)	Writable*3 (from 1 to 0 only)	Writable*3 (from 1 to 0 only)	When a reset or command is executed
0x0300_A200	0x0300_A200	Option Function Select Register 1 Secure (OFS1_SEC)	Writable	Writable	At a reset
0x0300_A210	0x0300_A210	Bank Select Register Secure (BANKSEL_SEC)	Writable	Writable	At a reset
0x0300_A240	0x0300_A240	Block Protect Setting Register Secure (BPS_SEC)	Writable*4	Writable*4	When a reset or command is executed
0x0300_A260	0x0300_A260	Permanent Block Protect Setting Register Secure (PBPS_SEC)	Writable*5 (from 1 to 0 only)	Writable*5 (from 1 to 0 only)	When a reset or command is executed
0x0300_A280	0x0300_A280	Option Function Select Register 1 Select (OFS1_SEL)*6	Writable	Writable	At a reset
0x0300_A290	0x0300_A290	Bank Select Register Select (BANKSEL_SEL)	Writable	Writable	At a reset
0x0300_A2C0	0x0300_A2C0	Block Protect Setting Register Select (BPS_SEL)	Writable*6	Writable*6	At a reset

Note 1. The SAS.FSPR bit cannot be restored to 1 by using the Configuration set command once it is set to 0. Therefore, setting the start-up area select flags again becomes impossible. (when the Configuration set command is issued to the address of 0x0100A134, the command is locked.) Exercise extra caution when handling the SAS.FSPRbit.
Note 2. Once PBPS[n] bit is set to 0, the BPS[n] bit cannot be restored to 1 by using the Configuration set command.
Note 3. Once these bits are set to 0, the bits cannot be restored to 1 by using the Configuration set command. The PBPS[n] bit cannot be set to 0 by using the Configuration set command when the BPS[n] bit is 1.
Note 4. Once PBPS_SEC[n] bit is set to 0, the BPS_SEC[n] bit cannot be restored to 1 by using the Configuration set command.
Note 5. Once these bits are set to 0, the bits cannot be restored to 1 by using the Configuration set command. The PBPS_SEC[n] bit cannot be set to 0 by using the Configuration set command when the BPS_SEC[n] bit is 1.
Note 6. When the Permanent Block Protect bit is enabled, Block protect setting register cannot be written.

Table 52.23 Address Used by Configuration Set Command for Secure alias (Data flash memory) (1 of 2)

Address	FSADDR	Setting Data	Operation of additional writing		Timing when the Setting is Enabled
	Register Value		Lock bit is 1	Lock bit is 0	
0x2703_0080~ 0x2703_017C	0x2703_0080~ 0x2703_017C	Lockable Configuration Data Area 0 (LK_CD_A0)	Writable*1	Not Writable*1	At a reset
0x2703_0180~ 0x2703_027C	0x2703_0180~ 0x2703_027C	Lockable Configuration Data Area 1 (LK_CD_A1)	Writable*1	Not Writable*1	At a reset
0x2703_0280~ 0x2703_037C	0x2703_0280~ 0x2703_037C	Lockable Configuration Data Area 2 (LK_CD_A2)	Writable*1	Not Writable*1	At a reset

Table 52.23 Address Used by Configuration Set Command for Secure alias (Data flash memory) (2 of 2)

Address	FSADDR	Setting Data	Operation of additional writing		Timing when the Setting is Enabled
	Register Value		Lock bit is 1	Lock bit is 0	
0x2703_0380~ 0x2703_0384	0x2703_0380~ 0x2703_0384	Configuration Data Lock Bit Area 0 (CD0_LK)	Writable	Writable	When a reset or command is executed
0x2703_0388~ 0x2703_038C	0x2703_0388~ 0x2703_038C	Configuration Data Lock Bit Area 1 (CD1_LK)	Writable (from 1 to 0 only)	Writable (from 1 to 0 only)	When a reset or command is executed
0x2703_0390	0x2703_0390	Configuration Data Lock Bit Area 2 (CD2_LK)	Writable (from 1 to 0 only)	Writable (from 1 to 0 only)	When a reset or command is executed
0x2703_03C0	0x2703_03C0	Anti-Rollback Counter Lock Setting (ARCLS)	Writable (from 1 to 0 only)	Writable (from 1 to 0 only)	When a reset or command is executed
0x2703_03C2		Anti-Rollback Counter Configuration Setting for ARC_NSEC (ARCCS)	Writable*2	Writable*2	When a reset or command is executed

Note 1. For details of relation of write-protection area and lock bit, see section 52.12.4. Data Flash Configuration Area Protection.
Note 2. The ARC_NSEC counter selection bit cannot be changed after set to 00b or 01b. When using the ARC_NSEC counter, this counter can be selected either of 4 lines of 64 bits (00b setting) or 1 line of 256 bits (01b setting).

52.9.3.16 Increment Counter Command

The Increment Counter command is used to security function, safety function. "Increment Counter" command is used for updating 3 types of anti-rollback counter (ARC_SEC, ARC_NSEC, and ARC_OEMBL). The counter update is executed increment processing and cannot be decrement processing.

Writing "0xD0" at 2nd write access of flash sequencer command triggers FACI to start “Increment Counter” command processing. Completion of command processing can be confirmed by FRDY bit of FSTATR register.

If "Command Lock" status occurs during command execution, refresh the counter with "Refresh Counter" command after "Command Lock" state is released.

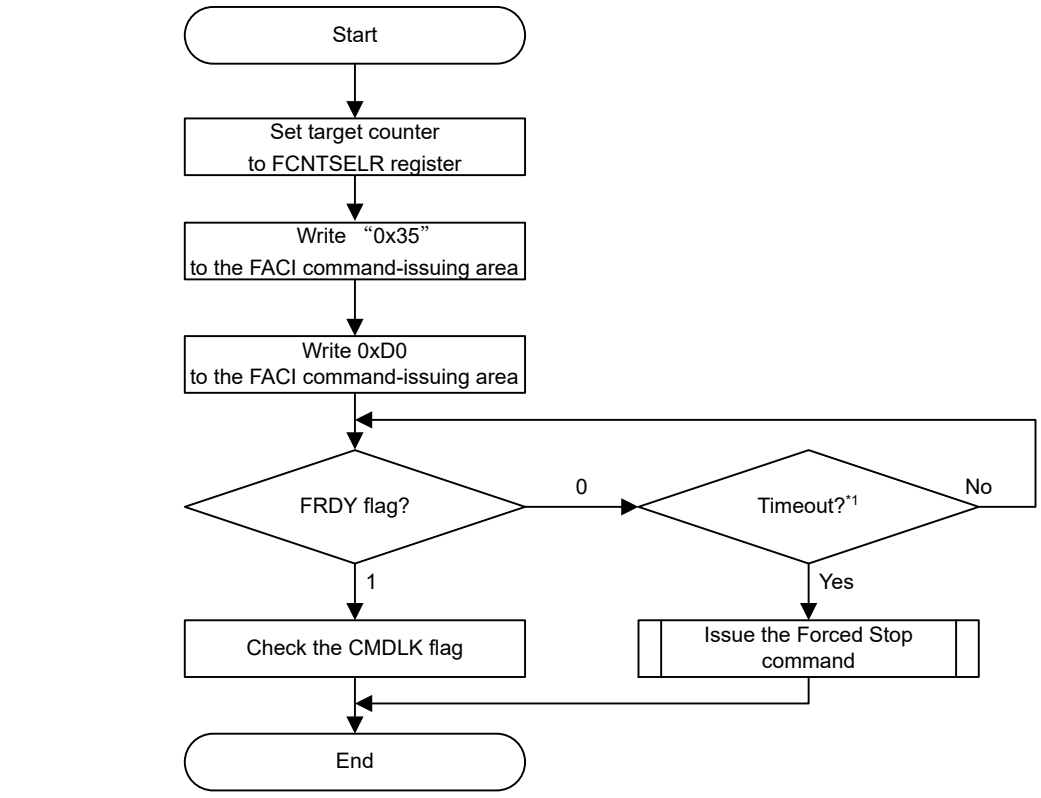


Figure 52.26 Usage of the Increment Counter Command

Anti-rollback counter value is not increased the following conditions by “Increment Counter” command.

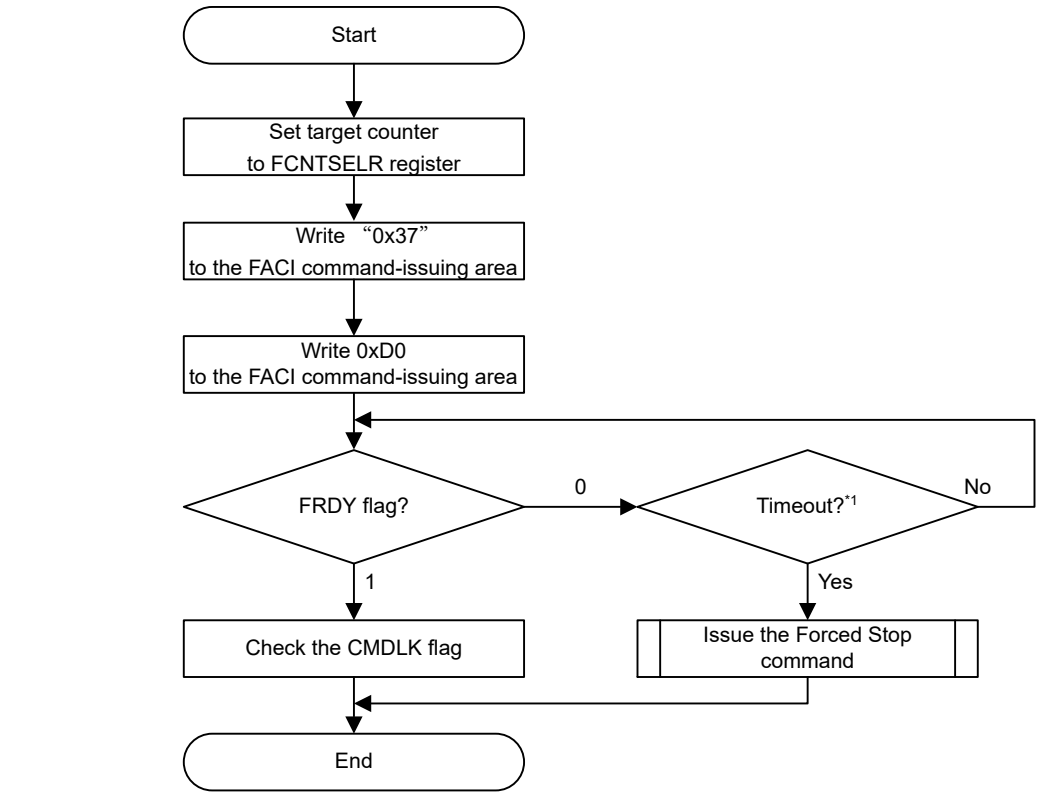
- Target counter is protected by ARCSEC_LK, ARCNSEC_LK0~3, or ARCOEMBL_LK.
- The previous command is not "Read Counter" command against ARC_OEMBL when target counter is ARC_OEMBL.
- The anti-rollback counter for OEM_BL cannot be incremented more than OEM_BL version number specified in the image version of the header of the code certificate. After updated the OEM_BL, the anti-rollback counter for OEM_BL cannot be incremented. For details, see section 43, Security Features .
- Target counter value is maximum. (Error does not occur)
- Target counter is ARC_NSEC when CNF_ARCNSEC0~1 is invalid setting.

52.9.3.17 Refresh Counter Command

The Refresh Counter command is used to security function, safety function. "Refresh Counter" command is used for update of anti-rollback counter when power failure is occurred while “Increment Counter” command processing. Counter value is not changed by "Refresh Counter" command.

Writing "0xD0" at 2nd write access of flash sequencer command triggers FACI to start “Increment Counter” command processing. Completion of command processing can be confirmed by FRDY bit of FSTATR register.

If "Command Lock" status occurs during command execution, refresh the counter with "Refresh Counter" command after "Command Lock" state is released.



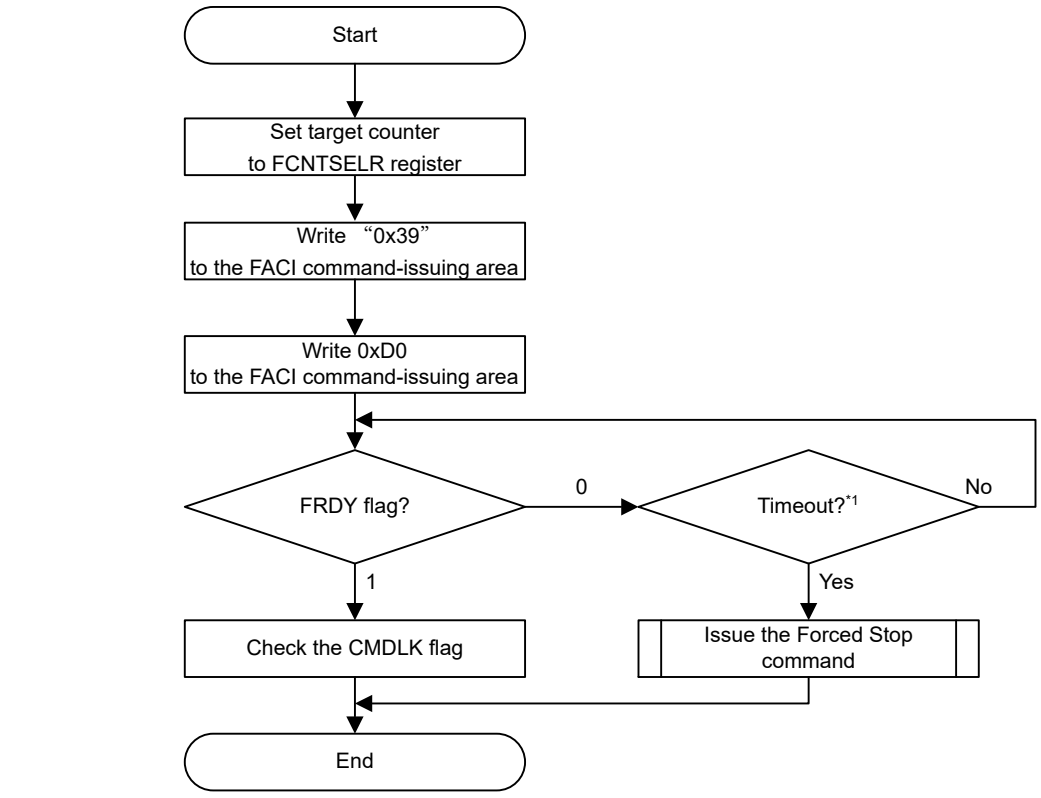
Note 1. 50 msec (FCLK ≥ 20 MHz)

Figure 52.27 Usage of the Refresh Counter Command

52.9.3.18 Read Counter Command

The Read Counter command is used to security function, safety function. "Read Counter" command is used for reading 3 types of anti-rollback counter (ARC_SEC, ARC_NSEC, and ARC_OEMBL). Flash sequencer automatically judges valid area of anti-rollback counter, reads 64 bits data, and stores to FCNTDATAR0 and FCNTDATAR1 registers.

Writing "0xD0" at 2nd write access of flash sequencer command triggers FACI to start “Read Counter” command processing. Completion of command processing can be confirmed by FRDY bit of FSTATR register.



Note 1. 50 msec (FCLK ≥ 20 MHz)

Figure 52.28 Usage of the Read Counter Command

52.10 Suspend Operation

Reading from the flash memory is not possible during programming or erasure if the conditions for background operation given in Table 52.39 are not satisfied. When a P/E suspend command is issued to suspend the programming or erasure of the flash memory, reading from the flash memory is enabled. Regarding P/E suspend commands, there are one suspend command mode for programming and two suspend command modes for erasure (suspension priority mode and erasure priority mode). To resume suspended programming or erasure, the P/E resume command is available. For details on the suspend operation, see Figure 52.17.

52.11 Protection Function

52.11.1 Software Protection

Software protection disables programming and erasure of the code flash memory through the settings of control registers and block protect setting in the user area. If an attempt is made to issue an FACI command against software protection, the flash sequencer enters the command-locked state.

52.11.1.1 Protection through FWEPROR

Unless the FWEPROR.FLWE[1:0] bits are set to 01b, programming cannot proceed in any mode.

52.11.1.2 Protection by FENTRYR

When the FENTRYR register is set to 0x0000, the flash sequencer enters read mode. In read mode, FACI commands cannot be accepted. If an attempt is made to issue an FACI command in read mode, the flash sequencer enters the command-locked state.

52.11.1.3 Protection by Block Protect Setting

Each block in user area has the block protect setting (BPS or BPS_SEC). When the FBPROT0 or FBPROT1 register is 0x0000 and the block protect bit is 0, issuing the Program or Block Erase command to user area of the code flash causes the command-locked state. To program or erase the block whose block protect bit is 0, set the FBPROT0 or FBPROT1 register to 0x0001.

The block protect setting can be locked by the permanent block protect setting (PBPS or PBPS_SEC). When the permanent block protect setting and the block protect setting are 0, issuing a Program or Block erase command to user area of the code flash causes the flash sequencer to enter the command-locked state regardless of the FBPROT0 and FBPROT1 register settings.

Valid block protect setting (BPS or BPS_SEC) depends on the Block Protect Select bit (BPS_SEL).

See section 52.12.2. Permanent Block Protect Setting for details of the block protect setting and permanent block protect setting. See section 52.4.17. FBPROT0 : Flash Block Protection Register and section 52.4.18. FBPROT1 : Flash Block Protection for Secure Register for more information.

For details of block protect setting (BPS or BPS_SEC) and block protect select (BPS_SEL), see section 6, Option-Setting Memory.

The protected area by the block protect setting is always determined by the address of the FSADDR register setting regardless of the address swapping function setting (the startup area select, the block swap in linear mode, and the startup bank select in dual mode). Table 52.24 to Table 52.29 show the relation of user area and the block protect setting in each function setting.

(1) In case of the linear mode

- BPS[0] to BPS[n] or BPS_SEC[0] to BPS_SEC[n] are assigned to the block of user area (for example, address is 0x00_0000 to the last block address).
- BPS[0]/BPS_SEC[0] and BPS[1]/BPS_SEC[1] are assigned to the block of user area depending on the startup area select setting (SAS.BTFLG bit). (See section 52.11.3. Start-Up Program Protection).
- The second half of FLI user area is assigned to block protect setting depending on the block swap select setting (BANKSEL.BLCKSWP[6:0] bits). See section 52.11.5. Block Swap Function.

Table 52.24 shows the block protect setting when the startup area select is disabled (not swapping).

Table 52.25 and later show example of the block protect setting when the address conversion function is used.

Table 52.24 Example of Block Protect setting when SAS.BTFLG is 1

FSADDR[23:0]	Block size	Block protect setting	User area block number	Notes
The last block address	32 KB	BPS[n] or BPS_SEC[n]	Block n	—
⋮	⋮	⋮	⋮	—
0x01_8000 to 0x01_FFFF	32 KB	BPS[9] or BPS_SEC[9]	Block 9	—
0x01_0000 to 0x01_7FFF	32 KB	BPS[8] or BPS_SEC[8]	Block 8	—
0x00_E000 to 0x00_FFFF	8 KB	BPS[7] or BPS_SEC[7]	Block 7	—
0x00_C000 to 0x00_DFFF	8 KB	BPS[6] or BPS_SEC[6]	Block 6	—
⋮	⋮	⋮	⋮	—
0x00_2000 to 0x00_3FFF	8 KB	BPS[1] or BPS_SEC[1]	Block 1	Not swap block 0 and block 1 in this startup area select setting
0x00_0000 to 0x00_1FFF	8 KB	BPS[0] or BPS_SEC[0]	Block 0	Not swap block 0 and block 1 in this startup area select setting

Table 52.25 Example of Block Protect setting when SAS.BTFLG is 0

FSADDR[23:0]	Block size	Block protect setting	User area block number	Notes
The last block address	32 KB	BPS[n] or BPS_SEC[n]	Block n	—
⋮	⋮	⋮	⋮	—
0x01_8000 to 0x01_FFFF	32 KB	BPS[9] or BPS_SEC[9]	Block 9	—
0x01_0000 to 0x01_7FFF	32 KB	BPS[8] or BPS_SEC[8]	Block 8	—
0x00_E000 to 0x00_FFFF	8 KB	BPS[7] or BPS_SEC[7]	Block 7	—
0x00_C000 to 0x00_DFFF	8 KB	BPS[6] or BPS_SEC[6]	Block 6	—
⋮	⋮	⋮	⋮	—
0x00_2000 to 0x00_3FFF	8 KB	BPS[1] or BPS_SEC[1]	Block 0	Swap block 0 and block 1 in this startup area select setting
0x00_0000 to 0x00_1FFF	8 KB	BPS[0] or BPS_SEC[0]	Block 1	Swap block 0 and block 1 in this startup area select setting

Table 52.26 Example of Block Protect setting when BANKSEL.BLCKSWP is 1

FSADDR[23:0]	Block size	Block protect setting	User area block number	Notes
The last block address	32 KB	BPS[n] or BPS_SEC[n]	Block n	—
⋮	⋮	⋮	⋮	—
The upper side address of the block swap target	32 KB	BPS[b] or BPS_SEC[b]	Block b	Not swap block a and block b in this block swap select setting
⋮	⋮	⋮	⋮	—
The lower side address of the block swap target	32 KB	BPS[a] or BPS_SEC[a]	Block a	Not swap block a and block b in this block swap select setting
⋮	⋮	⋮	⋮	—
0x01_8000 to 0x01_FFFF	32 KB	BPS[9] or BPS_SEC[9]	Block 9	—
0x01_0000 to 0x01_7FFF	32 KB	BPS[8] or BPS_SEC[8]	Block 8	—
0x00_E000 to 0x00_FFFF	8 KB	BPS[7] or BPS_SEC[7]	Block 7	—
0x00_C000 to 0x00_DFFF	8 KB	BPS[6] or BPS_SEC[6]	Block 6	—
⋮	⋮	⋮	⋮	—
0x002000 to 0x00_3FFF	8 KB	BPS[1] or BPS_SEC[1]	Block 1	—
0x00_0000 to 0x00_1FFF	8 KB	BPS[0] or BPS_SEC[0]	Block 0	—

Table 52.27 Example of Block Protect setting when BANKSEL.BLCKSWP is 0 (1 of 2)

FSADDR[23:0]	Block size	Block protect setting	User area block number	Notes
The last block address	32 KB	BPS[n] or BPS_SEC[n]	Block n	—
⋮	⋮	⋮	⋮	—
The updating side address of the block swap target	32 KB	BPS[b] or BPS_SEC[b]	Block a	Swap block a and block b in this block swap select setting
⋮	⋮	⋮	⋮	—
The operating side address of the block swap target	32 KB	BPS[a] or BPS_SEC[a]	Block b	Swap block a and block b in this block swap select setting

Table 52.27 Example of Block Protect setting when BANKSEL.BLCKSWP is 0 (2 of 2)

FSADDR[23:0]	Block size	Block protect setting	User area block number	Notes
⋮	⋮	⋮	⋮	—
0x01_8000 to 0x01_FFFF	32 KB	BPS[9] or BPS_SEC[9]	Block 9	—
0x01_0000 to 0x01_7FFF	32 KB	BPS[8] or BPS_SEC[8]	Block 8	—
0x00_E000 to 0x00_FFFF	8 KB	BPS[7] or BPS_SEC[7]	Block 7	—
0x00_C000 to 0x00_DFFF	8 KB	BPS[6] or BPS_SEC[6]	Block 6	—
⋮	⋮	⋮	⋮	—
0x00_2000 to 0x00_3FFF	8 KB	BPS[1] or BPS_SEC[1]	Block 1	—
0x00_0000 to 0x00_1FFF	8 KB	BPS[0] or BPS_SEC[0]	Block 0	—

(2) In case of the dual mode (DUALSEL.BANKMD[2:0] = 000b)

- BPS[0] to BPS[n] or BPS_SEC[0] to BPS_SEC[n] are assigned the block of bank 0 user area (for example, address is 0x00000000 to the last block address in the lower side bank).
- BPS[70] to BPS[70 + n] or BPS_SEC[70] to BPS_SEC[70 + n] are assigned to the block of bank 1 user area (for example, address is 0x00200000 to the last block address in the upper side bank).

Bank0 is upper side bank (BANKSEL.BANKSWP[2:0] = 000b)

- BPS[0] to BPS[n] or BPS_SEC[0] to BPS_SEC[n] are assigned to the block of bank1 user area (e.g. Address is 0x00_0000 to The last block address in lower side bank).
- BPS[70] to BPS[70+n] or BPS_SEC[70] to BPS_SEC[70+n] are assigned to the block of bank0 user area (e.g. Address is 0x20_0000 to The last block address in upper side bank).

Table 52.28 and Table 52.29 show example of the block protect setting in the dual mode. See section 52.11.4. Dual Bank Function for details of dual bank function (DUALSEL.BANKMD[2:0] and BANKSEL.BANKSWP[2:0] bits).

Table 52.28 Example of Block Protect setting when BANKSEL.BANKSWP[2:0] is 111b (1 of 2)

FSADDR[23:0]	Block size	Block protect setting	User area block number	Notes
The last block address in upper side bank	32 KB	BPS[70 + n] or BPS_SEC[70 + n]	Block 70 + n	Not swap bank 0 and bank 1 in this startup bank switch setting
⋮	⋮	⋮	⋮	
0x21_8000 to 0x21_FFFF	32 KB	BPS[79] or BPS_SEC[79]	Block 79	
0x21_0000 to 0x21_7FFF	32 KB	BPS[78] or BPS_SEC[78]	Block 78	
0x20_E000 to 0x20_FFFF	8 KB	BPS[77] or BPS_SEC[77]	Block 77	
0x20_C000 to 0x20_DFFF	8 KB	BPS[76] or BPS_SEC[76]	Block 76	
⋮	⋮	⋮	⋮	
0x20_2000 to 0x20_3FFF	8 KB	BPS[71] or BPS_SEC[71]	Block 71	
0x20_0000 to 0x20_1FFF	8 KB	BPS[70] or BPS_SEC[70]	Block 70	

Table 52.28 Example of Block Protect setting when BANKSEL.BANKSWP[2:0] is 111b (2 of 2)

FSADDR[23:0]	Block size	Block protect setting	User area block number	Notes
The last block address in lower side bank	32 KB	BPS[n] or BPS_SEC[n]	Block n	Not swap bank 0 and bank 1 in this startup bank switch setting
⋮	⋮	⋮	⋮	
0x01_8000 to 0x01_FFFF	32 KB	BPS[9] or BPS_SEC[9]	Block 9	
0x01_0000 to 0x01_7FFF	32 KB	BPS[8] or BPS_SEC[8]	Block 8	
0x00_E000 to 0x00_FFFF	8 KB	BPS[7] or BPS_SEC[7]	Block 7	
0x00_C000 to 0x00_DFFF	8 KB	BPS[6] or BPS_SEC[6]	Block 6	
⋮	⋮	⋮	⋮	
0x00_2000 to 0x00_3FFF	8 KB	BPS[1] or BPS_SEC[1]	Block 1	
0x00_0000 to 0x00_1FFF	8 KB	BPS[0] or BPS_SEC[0]	Block 0	

Table 52.29 Relation of User area and Block Protect setting when BANKSEL.BANKSWP[2:0] is 000b

FSADDR[23:0]	Block size	Block protect setting	User area block number	Notes
The last block address in the upper side bank	32 KB	BPS[70 + n] or BPS_SEC[70 + n]	Block n	Swap bank 0 and bank 1 in this startup bank switch setting
⋮	⋮	⋮	⋮	
0x21_8000 to 0x21_FFFF	32 KB	BPS[79] or BPS_SEC[79]	Block 9	
0x21_0000 to 0x21_7FFF	32 KB	BPS[78] or BPS_SEC[78]	Block 8	
0x20_E000 to 0x20_FFFF	8 KB	BPS[77] or BPS_SEC[77]	Block 7	
0x20_C000 to 0x20_DFFF	8 KB	BPS[76] or BPS_SEC[76]	Block 6	
⋮	⋮	⋮	⋮	
0x20_2000 to 0x20_3FFF	8 KB	BPS[71] or BPS_SEC[71]	Block 1	
0x20_0000 to 0x20_1FFF	8 KB	BPS[70] or BPS_SEC[70]	Block 0	
The last block address in the lower side bank	32 KB	BPS[n] or BPS_SEC[n]	Block 70 + n	Swap bank 0 and bank 1 in this startup bank switch setting
⋮	⋮	⋮	⋮	
0x01_8000 to 0x01_FFFF	32 KB	BPS[9] or BPS_SEC[9]	Block 79	
0x01_0000 to 0x01_7FFF	32 KB	BPS[8] or BPS_SEC[8]	Block 78	
0x00_E000 to 0x00_FFFF	8 KB	BPS[7] or BPS_SEC[7]	Block 77	
0x00_C000 to 0x00_DFFF	8 KB	BPS[6] or BPS_SEC[6]	Block 76	
⋮	⋮	⋮	⋮	
0x00_2000 to 0x00_3FFF	8 KB	BPS[1] or BPS_SEC[1]	Block 71	
0x00_0000 to 0x00_1FFF	8 KB	BPS[0] or BPS_SEC[0]	Block 70	

52.11.2 Error Protection

Error protection detects the issuing of illegal FACI commands, illegal access, and flash sequencer malfunction. FACI command acceptance is disabled (command-locked state) in response to the detection of these errors. The flash memory cannot be programmed or erased while the flash sequencer is in the command-locked state. For release from the command-locked state, issue the Status Clear or Forced Stop command. The Status Clear command can only be used while the FRDY bit in the FSTATR register is 1. The Forced Stop command can be used regardless of the value of the FRDY bit. While the CMDLKIE bit in the FAEINT register is 1, a flash access error (FIFERR) interrupt is generated if the flash sequencer enters the command-locked state (the CMDLK bit in the FASTAT register is set to 1).

If the flash sequencer enters the command-locked state in response to a command other than the P/E suspend command during programming or erasure processing, the flash sequencer continues the processing for programming or erasure. In this

state, the P/E suspend command cannot be used to suspend the processing for programming or erasure. If a command is issued in the command-locked state, the ILGLERR bit becomes 1 and the other bits retain the values set from previous error detection.

Table 52.30 shows the error protection types and status bit values after error detections.

Table 52.30 Error protection type (1 of 3)

Error type	Description	ILGCOMERR	FESETERR	SECERR	OTERR	TZFERR	ILGLERR	ERSERR	PRGERR	FLWEERR	CFAE	DFAE
FENTRYR setting error	The value set in FENTRYR is not 0x0000, 0x0001, or 0x0080	0	1	0	0	0	1	0	0	0	0	0
	The FENTRYR setting at suspension is different from that at resumption	0	1	0	0	0	1	0	0	0	0	0
Illegal command error	An undefined size is specified in the first cycle of the command. (not byte-write)	1	0	0	0	0	1	0	0	0	0	0
	An undefined code is written in the first access of the FACI command	1	0	0	0	0	1	0	0	0	0	0
	The value specified in the last access of the multiple-access FACI command is not 0xD0	1	0	0	0	0	1	0	0	0	0	0
	The value (N) specified in the second write access of the FACI command in the program or configuration set command is wrong	1	0	0	0	0	1	0	0	0	0	0
	Blank Check command is issued with inconsistent BCDIR, FSADDR, and FEADDR settings (see section 52.4.13. FEADDR : FACI Command End Address Register)	1	0	0	0	0	1	0	0	0	0	0/1 *1
	A multi block erase command is issued with inconsistent FSADDR and FEADDR settings. ● FSADDR > FEADDR ● FEADDR is set to reserved area.	1	0	0	0	0	1	0	0	0	0	0/1 *1
	Increment Counter, Refresh Counter, or Read Counter command is issued with inconsistent FCNTSELR[2:0] setting. (Include CNF_ARCNS0,1 setting is the mismatch.)	1	0	0	0	0	1	0	0	0	0	0
	An FACI command not acceptable in each mode is issued (see Table 52.19)	1	0	0	0	0	1	0	0	0	0	0
	An FACI command is issued when command acceptance conditions are not satisfied (see Table 52.20)	0/1	0/1	0/1	0/1	0	1	0/1	0/1	0/1	0/1	0/1
	A program, block erase or multi block erase command is issued against the area protected by the block protect setting (see section 52.11.1.3. Protection by Block Protect Setting)	1	0	0	0	0	1	0	0	0	0	0

Table 52.30 Error protection type (2 of 3)

Error type	Description	ILGCOMERR	FESETERR	SECERR	OTERR	TZFERR	ILGLERR	ERSERR	PRGERR	FLWEERR	CFAE	DFAE
Illegal command error	Configuration set command is issued against Data flash configuration area protected by lock bit setting	1	0	0	0	0	1	0	0	0	0	0
	Increment Counter or Refresh Counter command is issued against anti-rollback area protected by lock bit setting	1	0	0	0	0	1	0	0	0	0	0
	A program command is issued against the erase area in erase suspend	1	0	0	0	0	1	0	0	0	0	0
	Increment counter command against ARC_OEMBL is issued when the RSIP-E51A register setting is same or smaller than ARC_OEMBL value. (see section 52.12.5. Anti-Rollback Counter)	1	0	0	0	0	1	0	0	0	0	0
	Increment counter command against ARC_OEMBL is issued when previous command is not read counter command against ARC_OEMBL.	1	0	0	0	0	1	0	0	0	0	0
	A command is issued against flash sequencer command illegal status (see Table 52.17).	0/1	0/1	0/1	0/1	0	1	0/1	0/1	0/1	0/1	0/1
Erasure error	An error occurs during erasure	0	0	0	0	0	0	1	0	0	0	0
Programming error	An error occurs during programming	0	0	0	0	0	0	0	1	0	0	0
Code flash memory access violation	An FACI command is issued to the reserved portion of the user area in code flash P/E mode	0	0	0	0	0	1	0	0	0	1	0
	Configuration set command is issued to the reserved configuration area in code flash P/E mode.	0/1	0	0	0	0	1	0	0	0	1	0
	Program, block erase, or configuration set command is issued to TrustZone protection area in code flash P/E mode.	0/1	0	0	0	1	1	0	0	0	1	0
Data flash memory access violation	A program or block erase command is issued to the reserved data area in data flash P/E mode	0	0	0	0	0	1	0	0	0	0	1
	A multi block erase command is issued to the reserved data area in data flash P/E mode. (FSADDR is set to reserved data area).	1	0	0	0	0	1	0	0	0	0	1
	Blank Check command is issued to reserved data area in data flash P/E mode. (FSADDR is set to reserved data area).	1	0	0	0	0	1	0	0	0	0	1
	Configuration set command has been issued to reserved configuration area in data flash P/E mode.	0/1	0	0	0	0	1	0	0	0	0	1
	A program, block erase, multi block erase, blank check, configuration set, increment counter, refresh counter, or read counter command is issued to TrustZone protection area in data flash P/E mode.	0	0	0	0	0	1	0	0	0	0	1

Table 52.30 Error protection type (3 of 3)

Error type	Description	ILGCOMERR	FESETERR	SECERR	OTERR	TZFERR	ILGLERR	ERSERR	PRGERR	FLWEERR	CFAE	DFAE
Security error	Configuration set command for the SAS.BTFLG bit setting is issued when the SAS.FSPR bit is 0 (see section 52.9.3.15. Configuration Set Command)	0	0	1	0	0	1	0	0	0	0	0
Others	An FACI command-issuing area is accessed in read mode	0	0	0	1	0	1	0	0	0	0	0
	An FACI command-issuing area is read in code flash P/E mode or data flash P/E mode	0	0	0	1	0	1	0	0	0	0	0
Flash write erase protection error	A flash memory write protection error is detected by the FWEPROR register setting*2 during command processing by the flash sequencer	0	0	0	0	0	0	0/1	0/1	1	0	0

Note 1. DFAE value depends on the FSADDR setting.
Note 2. For details on the FWEPROR register, see section 52.4.8. FWEPROR : Flash P/E Protect Register.

52.11.3 Start-Up Program Protection

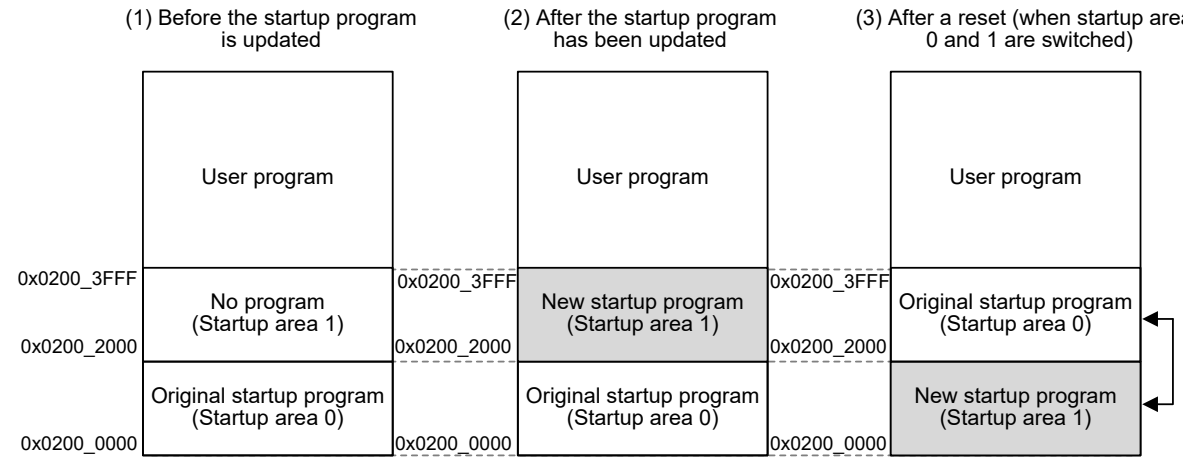
Protection of the startup program is for protection of the program to be started after a reset (the startup program). This function provides a way to safely update the startup program when rewriting is suspended during a reset.

The startup area is 8 Kbytes in size and is assigned to the user area in the code flash memory. This function uses the values of the SAS.BTFLG bit and the FSUACR.SAS[1:0] bits to change the area where the startup program is stored in block units (see Figure 52.29 to Figure 52.32).

In protection of the startup program, the state of the selection of the startup area can be fixed by the FSPR bit. However, the SAS.FSPR bit never be restored to 1 once the flag is set to 0. Exercise extra caution when handling the SAS.FSPR bit.

In addition, this protection cannot be used when dual mode is selected by the bank mode switching function (when the BANKMD[2:0] bits are 000b).

For details of SAS.FSPR bit and DUALSEL.BANKMD[2:0] bits, see section 6, Option-Setting Memory.



Note: The addresses are used for reading, programming, and erasure. Startup area 0 is erased when Block Erase command is issued for the address 0x0200_0000 with allocation shown under 3.

1. Before the startup program is updated
   When the SAS.BTFLG bit is 1 and the FSUACR.SAS[1:0] bits are 0xb, the address range from 0x0100_2000 to 0x0200_3FFF is selected as startup area 1 and the address range from 0x0200_0000 to 0x0200_1FFF is selected as startup area 0.
2. After the startup program has been updated
   A new startup program can be written to startup area 1. At this time, startup area 0 will still contain the previous startup program, so rewriting by starting up from startup area 0 remains possible if rewriting to the startup area 1 fails.
3. After a reset (when startup areas 0 and 1 are switched)
   When rewriting startup area 1 is successfully completed, switch the assignment of startup areas 0 and 1 by rewriting the startup area selection bit (the SAS.BTFLG bit) with Configuration set command. Afterwards, the program is started up from startup area 1 after a reset.

Figure 52.29 Concept of Protection of the Startup Program for Secure alias

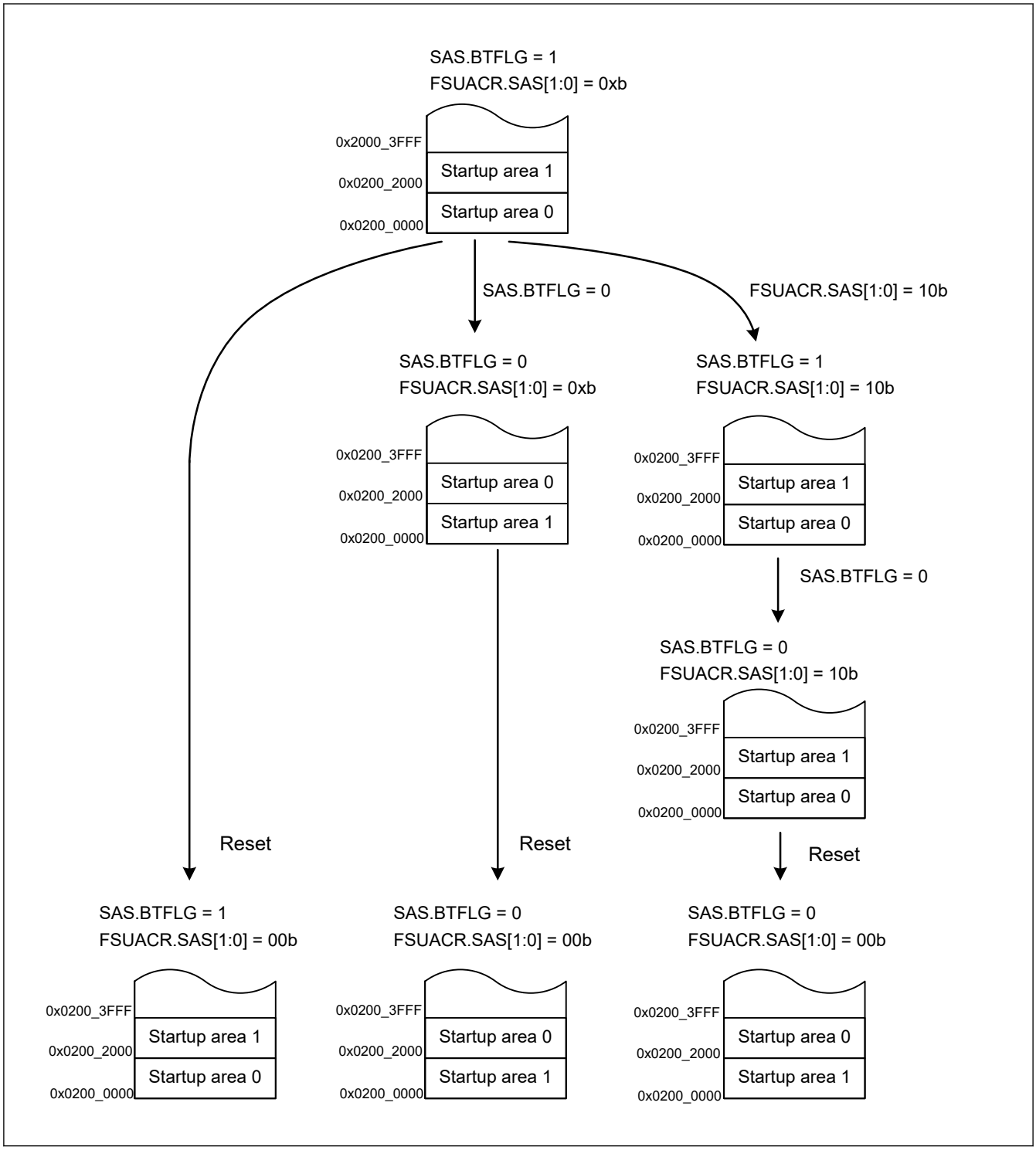


Figure 52.30 Example 1 of Transitions for Startup Program Protection Settings for Secure alias

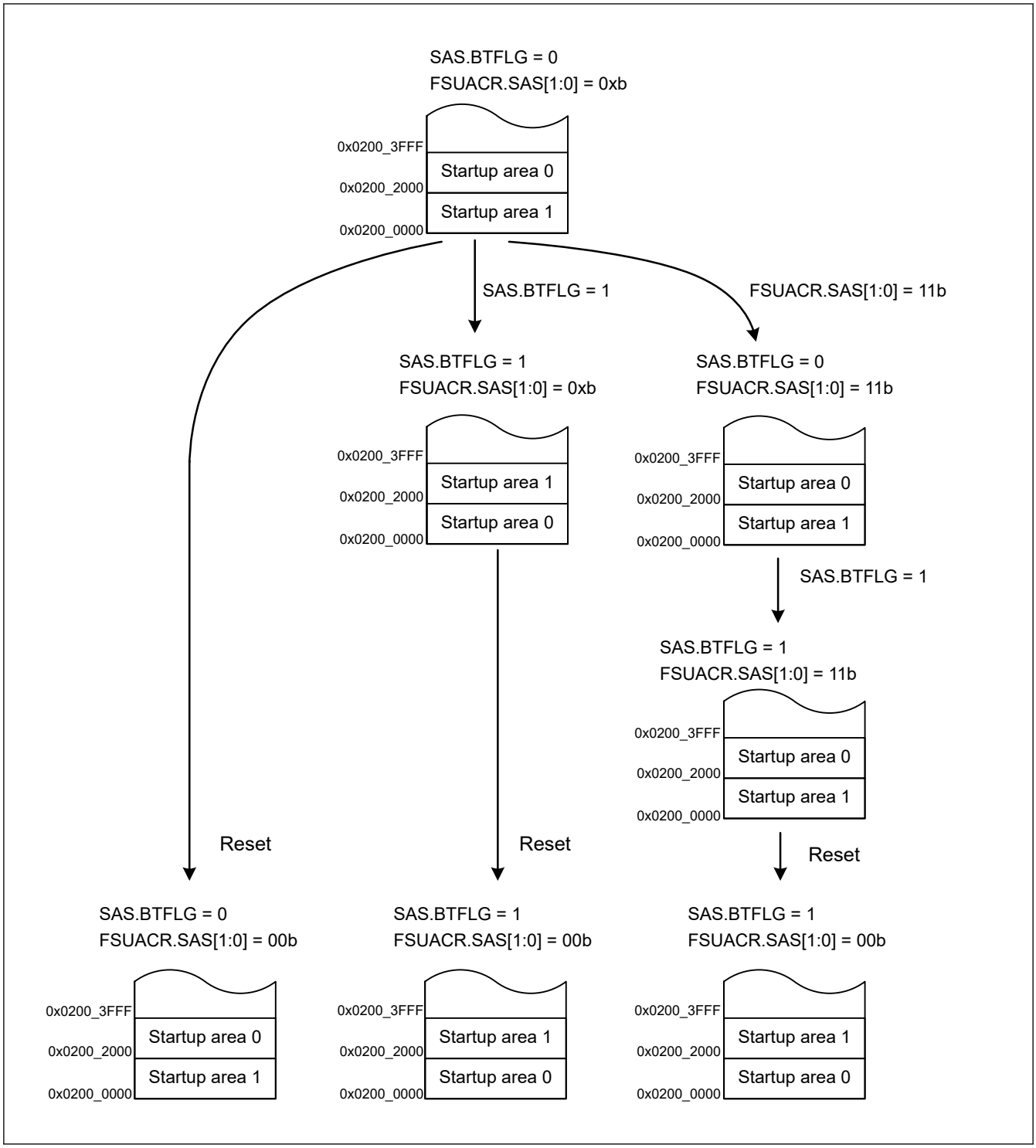
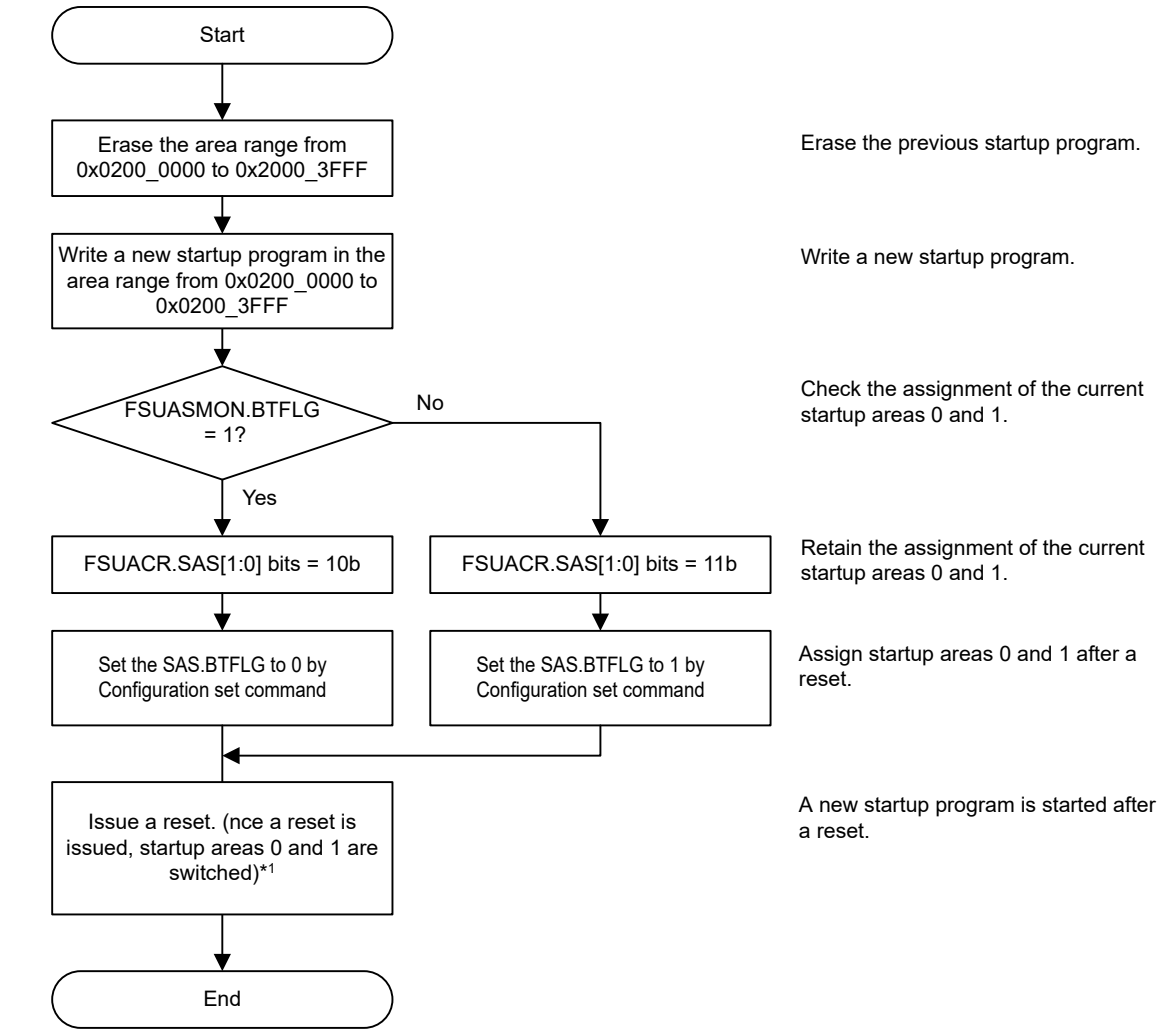


Figure 52.31 Example 2 of Transitions for Startup Program Protection Settings for Secure alias



Note 1. For details, see Figure 52.30 and Figure 52.31.

Figure 52.32 Concept of Protection of the Startup Program for Secure alias

52.11.4 Dual Bank Function

This protection uses the functions of bank mode switching and startup bank selection to update a program while a user program is running and to provide a safe method of updating in cases where programming is suspended during a reset.

52.11.4.1 Switching Bank Modes

The bank mode switching function selects either linear mode in which the user area in the code flash memory is used as one area, or dual mode in which the user area is divided into two bank areas. Figure 52.33 shows an example of flow of switching bank modes. A reset after setting the DUALSEL.BANKMD[2:0] bits in the option setting memory determines the mode of the bank mode switching function. Selecting dual mode enables the startup bank selection function.

When dual mode is selected by bank mode switching function (the DUALSEL.BANKMD[2:0] bits are 000b), start-up program protection function cannot be used.

For details of DUALSEL.BANKMD[2:0] bits, see section 6, Option-Setting Memory.

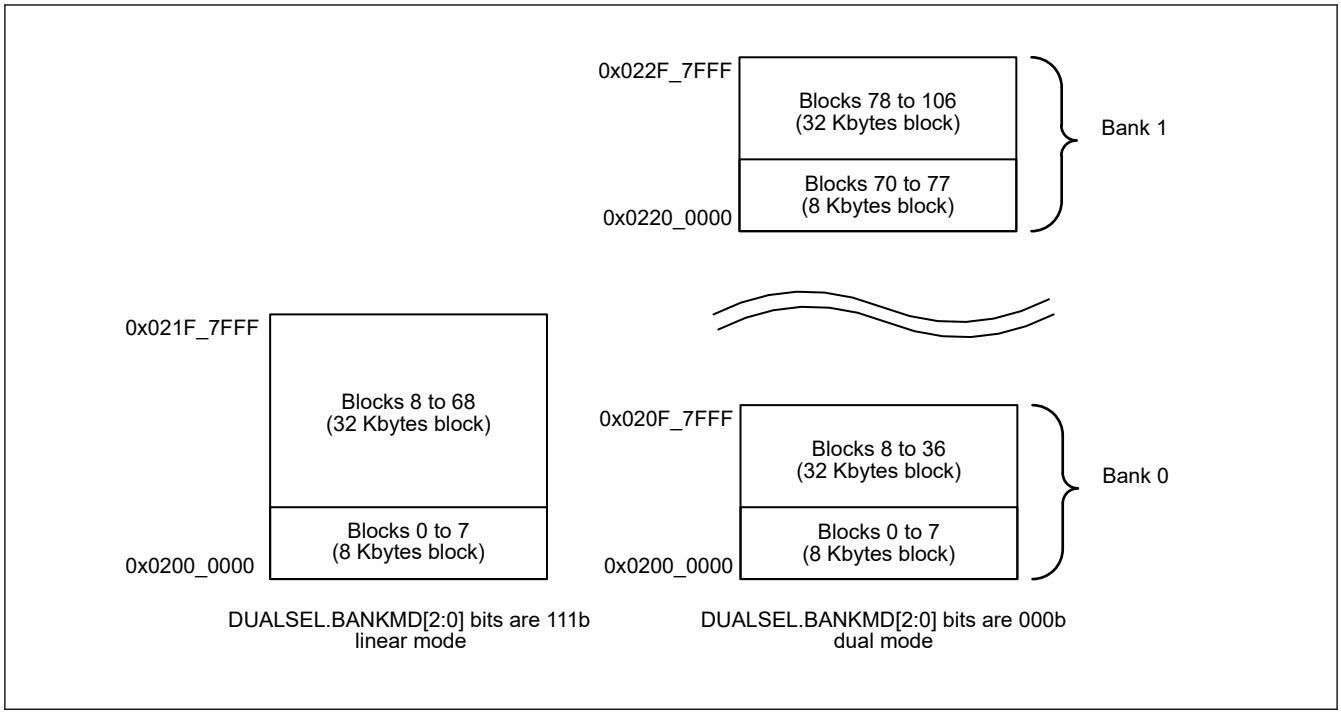


Figure 52.33 Example of Flow of Switching Bank Modes for Secure alias (For Products with 2 Mbytes of Code Flash Memory)

52.11.4.2 Selecting the Startup Bank

Startup bank selection provides a way to safely update the program by selecting a bank area to be started in dual mode (when the DUALSEL.BANKMD[2:0] bits are 000b) when programming is suspended during a reset. Figure 52.34 is a schematic view of startup bank selection and Table 52.31 shows an example of the flow of startup bank selection. A reset after setting the value of the BANKSEL.BANKSWP[2:0] bits in the option-setting memory changes the addresses of banks 0 and 1 and booting up a program proceeds from the updated area. When the address is switched by using startup bank selection, the programming/erasure target for the FACI commands is also switched. This function is invalid in linear mode.

For details of DUALSEL.BANKMD[2:0] bits and BANKSEL.BANKSWP[2:0] bits, see section 6, Option-Setting Memory.

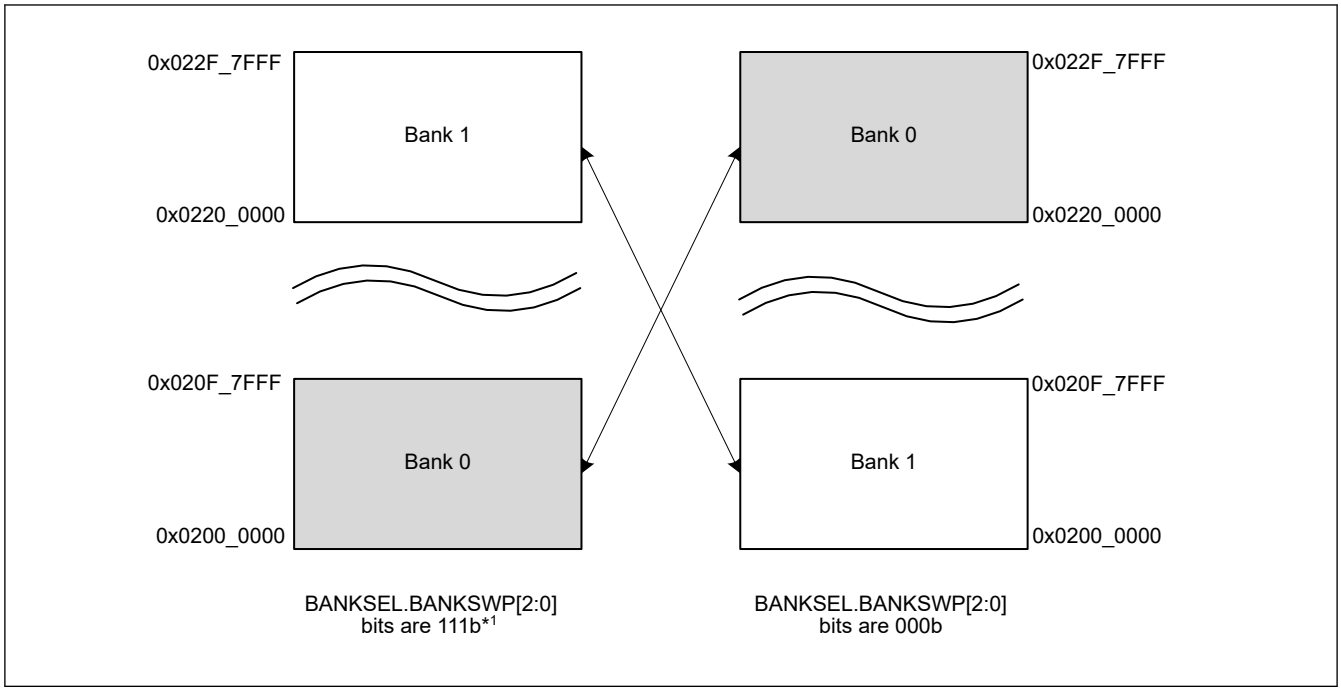


Figure 52.34 Example of Startup Bank Selection for Secure alias (For Products with 2 MB of Code Flash Memory)

Table 52.31 Example of Startup Bank Selection Flow for Secure alias (For Products with 2 MB of Code Flash Memory)

No.	Step Name	Description
1	Erase block	Erase blocks to be programed in the address range from 0x0220_0000 to 0x022F_7FFF
2	Program a new software	Program a new software in the address range from 0x0220_0000 to 0x022F_7FFF
3	Read the value	Read the value of the BANKSEL.BANKSWP [2:0] bits.
4	Write an inverted value	Write an inverted value in the BANKSEL.BANKSWP[2:0] bits*1.
5	Issue a reset	Issue a reset (A reset switches the banks.)

Note 1. Set the inverse of the read value of the BANKSEL.BANKSWP[2:0] bits (000b or 111b).

52.11.5 Block Swap Function

This protection uses the functions of block swap selection to update a program while a user program is running and to provide a safe method of updating in cases where programming is suspended during a reset.

52.11.5.1 Selecting the Block Swap

Block swap selection provides a way to safely update the program when programming is suspended during a reset. Figure 52.35 is a schematic view of block swap selection and Figure 52.36 shows an example of the flow of block swap selection. A reset after setting the value of the BANKSEL.BLCKSWP bit in the option-setting memory changes the addresses of block A and B. When the address is switched, the Programming/erasure target for the FACI commands is also switched.

In addition, this protection cannot be used when dual mode is selected by the bank mode switching function (when the DUALSEL.BANKMD[2:0] bits are 000b).

For details of DUALSEL.BANKMD[2:0] bits and BANKSEL.BLCKSWP bit, see section 6, Option-Setting Memory.

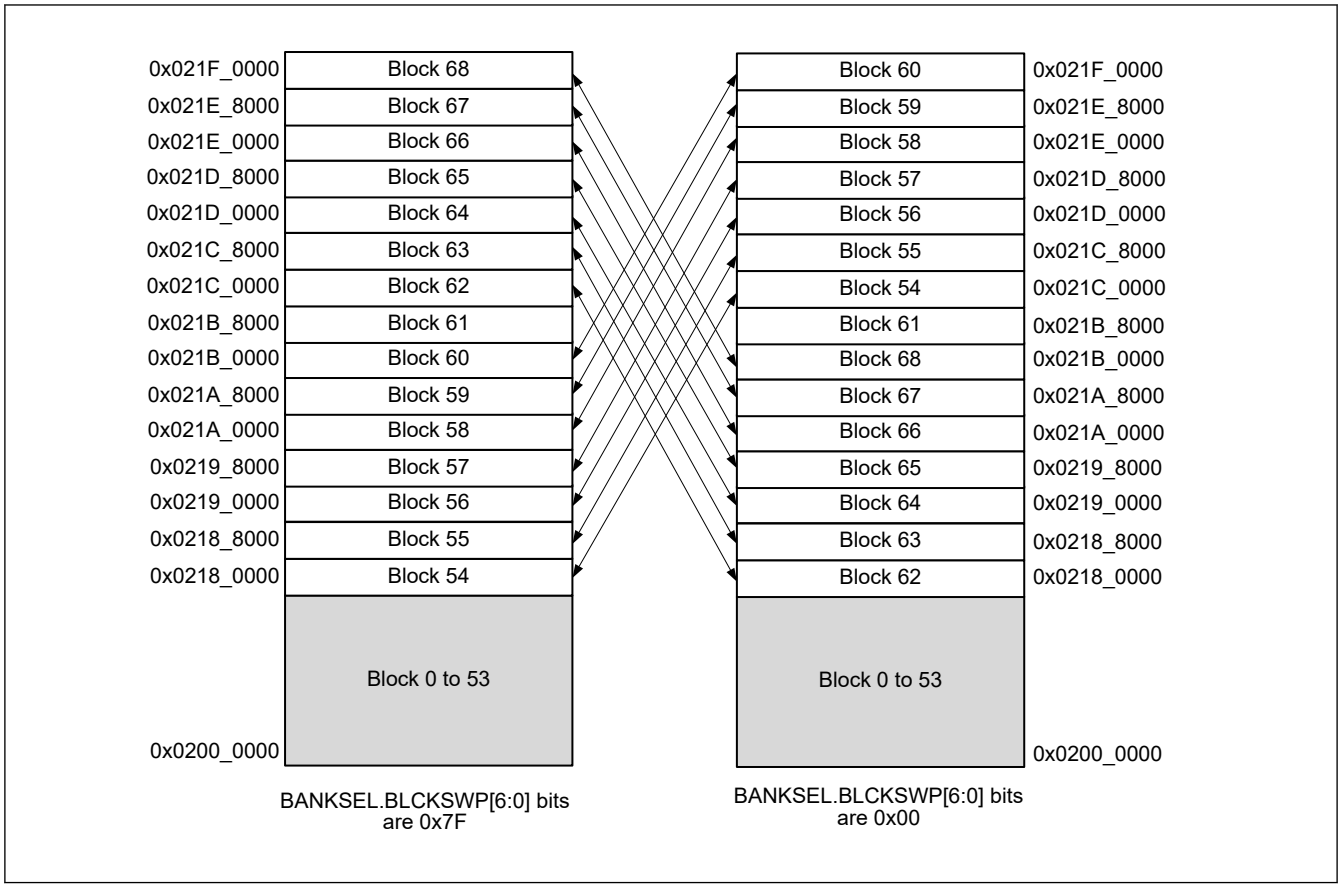
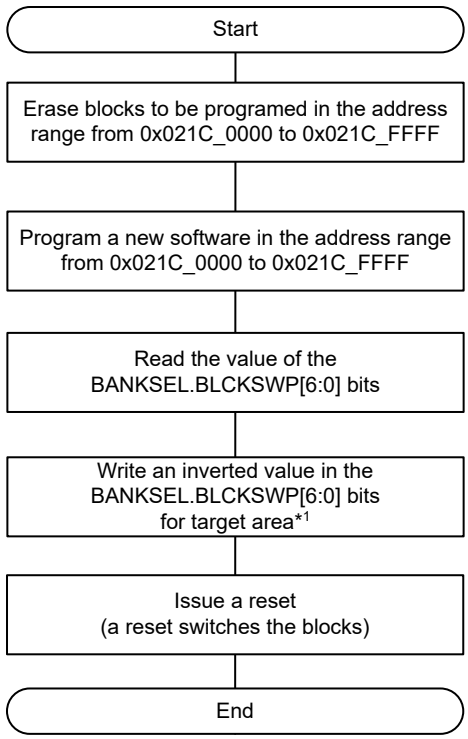


Figure 52.35 Example of Block Swap Selection for Secure alias (For Products with 2 Mbytes of Code Flash Memory)



Note 1. See Table 6.2 for the correspondence between the target region and the BANKSEL.BLCKSWP[6:0] bits.

Figure 52.36 Example of Block Swap Selection Flow for Secure alias (For Products with 2 Mbytes of Code Flash Memory)

52.12 Security Function

The flash sequencer supports the following security functions:

- Security flag for startup area
- Permanent block protect setting
- Flash memory protection for TrustZone
- Code Flash P/E mode Entry Protection
- Data flash configuration area protection
- Anti-rollback Counter

52.12.1 Security Flag for Startup Area Select

The security flag (SAS.FSPR) for the startup area is located in the option-setting memory.

When the SAS.FSPR bit is 0, issuing the configuration set command to change the SAS.BTFLG bit causes the flash sequencer to be in the command-locked state. Also, when the SAS.FSPR bit is 0, it is invalid to write to the Startup Area Select bits SAS[1:0] in the FSUACR register. The SAS.FSPR bit enables protection.

52.12.2 Permanent Block Protect Setting

The permanent block protect setting is the clear protection for the block protection setting. User area cannot be permanently updated by the FACI command when the permanent block protect setting is enabled. See section 52.11.1.3. Protection by Block Protect Setting for more details.

The block protect setting and the permanent block protect setting have the write/clear protection against the configuration set command. The flash sequencer does not detect an error when the configuration set command is issued to the write/clear protected settings.

Figure 52.37 and Table 52.32 show the write/clear protection against the block protect setting (BPS[n]) and the permanent protect setting (PBPS[n]). Figure 52.38 and Table 52.33 show the write/clear protection against the block protect setting for Secure (BPS_SEC[n]) and permanent protect setting for Secure (PBPS_SEC[n]).

Effective permanent block protect setting (PBPS or PBPS_SEC) depends on block protect select (BPS_SEL). For details of permanent block protect setting (PBPS or PBPS_SEC) and block protect select (BPS_SEL), see section 6, Option-Setting Memory.

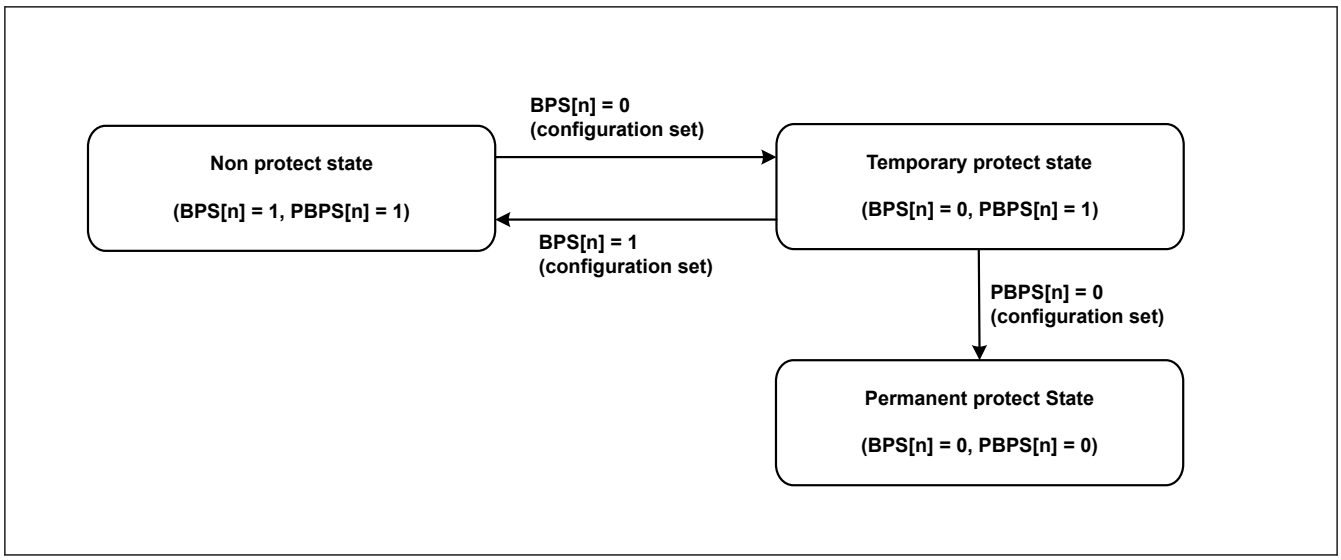


Figure 52.37 Status transition of flash sequencer by BPS[n] and PBPS[n]

Table 52.32 Write/clear protection of BPS[n] , PBPS[n] and BPS_SEL[n]

Current state		Updatable state by configuration set command				
BPS[n]	PBPS[n]	BPS[n] = 1	BPS[n] = 0	PBPS[n] = 1	PBPS[n] = 0	BPS_SEL[n] = 1
1	1	✓	✓	✓	X	✓
1	0	—	—	—	—	—
0	1	✓	✓	✓	✓	✓
0	0	X	✓	X	✓	X

Note:
- ✓ indicates updatable by configuration set command.
- X indicates not updatable by configuration set command (error does not occur).
- — indicates not reaching to this state.

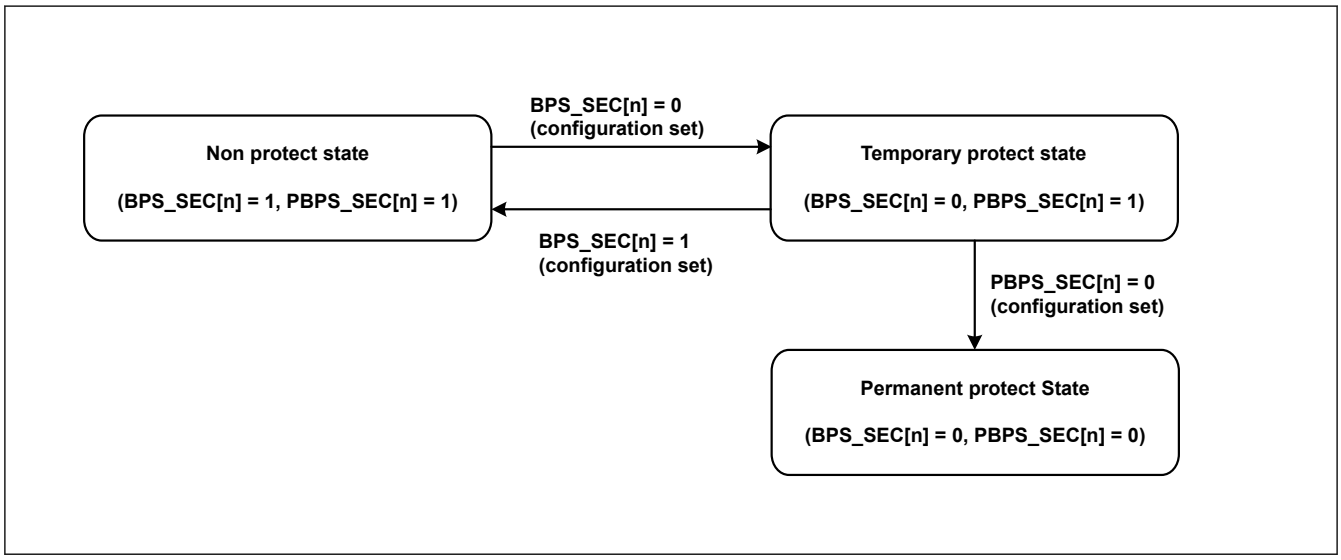


Figure 52.38 Status transition of flash sequencer by BPS_SEC[n] and PBPS_SEC[n]

Table 52.33 Write/clear protection of BPS_SEC[n], PBPS_SEC[n] and BPS_SEL[n]

Current state		Updatable state by configuration set command				
BPS_SEC[n]	PBPS_SEC[n]	BPS_SEC[n] = 1	BPS_SEC[n] = 0	PBPS_SEC[n] = 1	PBPS_SEC[n] = 0	BPS_SEL[n] = 0
1	1	✓	✓	✓	X	✓
1	0	—	—	—	—	—
0	1	✓	✓	✓	✓	✓
0	0	X	✓	X	✓	X

Note:
- ✓ indicates updatable by configuration set command.
- X indicates not updatable by configuration set command (error does not occur).
- — indicates not reaching to this state.

52.12.3 Flash Memory Protection for TrustZone

Information in this section focuses on the flash sequencer operation.

The flash memory provides the following types of protect function against Non-secure access:

- Protection for flash memory area (P/E)
- Protection for flash memory area (read)
- Protection for registers
- Code flash P/E mode entry protection

52.12.3.1 Protection for Flash Memory Area (P/E)

This function protects the Secure region of the code flash and data flash from FACI commands of Non-secure access. The condition of protection depends on the FACI command, the access attribution, and the memory boundary setting.

For details of Secure region, see section 43, Security Features .

See Table 52.34 for information on protection of the flash memory area (P/E).

Table 52.34 Protection for the flash memory area (P/E) (1 of 2)

FACI command	Target area			Issuing FACI command by Non-secure access	Issuing FACI command by Secure access
Program Block Erase	Code flash memory	User area (Non-secure area)	Secure alias*1	X	X
			Non-secure alias*2	✓	✓
		User area (Secure area)	Secure alias*1	X	✓
			Non-secure alias*2	X	X
	Data flash memory	Data area (Non-secure area)	Secure*1 alias	X	X
			Non-secure alias*2	✓	✓
		Data area (Secure area)	Secure alias*1	X	✓
			Non-secure alias*2	X	X
Multi Block Erase Blank Check	Data flash memory	Data area (Non-secure area)	Secure alias*3	X	X
			Non-secure alias*4	✓	✓
		Data area (Secure area)	Secure alias*3	X	✓
			Non-secure alias*4	X	X

Table 52.34 Protection for the flash memory area (P/E) (2 of 2)

FACI command	Target area			Issuing FACI command by Non-secure access	Issuing FACI command by Secure access
Configuration set	Code flash memory	Configuration area (Non-secure area)	Secure alias*1	X	X
			Non-secure alias*2	✓	✓
		Configuration area (Secure area)	Secure alias*1	X	✓
			Non-secure alias*2	X	X
	Data flash memory	Configuration area (Secure area only)	Secure alias*1	X	✓
			Non-secure alias*2	X	X
Increment Counter Refresh Counter Read Counter	Data flash memory	Anti-rollback counter area (Secure area only)	-	X	✓

Note:
- ✓ indicates FACI command operation is not prohibited.
- X indicates FACI command operation is prohibited. Error occurs when the area is selected, and the FACI command is executed.

Note 1. The Secure alias is judged when the address of FSADDR[28] is 0.
Note 2. The Non-secure alias is judged when the address of FSADDR[28] is 1.
Note 3. The Secure alias is judged when the address of FSADDR[28] and FEADDR[28] are 0.
Note 4. The Non-secure alias is judged when the address of FSADDR[28] and FEADDR[28] are 1.

When the target area of FACI command is the user area of code flash, the flash sequencer compares the FSADDR register setting with the memory boundary setting of the code flash and determines whether the target area is in the Secure region.

In linear mode, the memory boundary can be set to 0x0200_0000 to 0x021F_0000 in 32 KB unit.

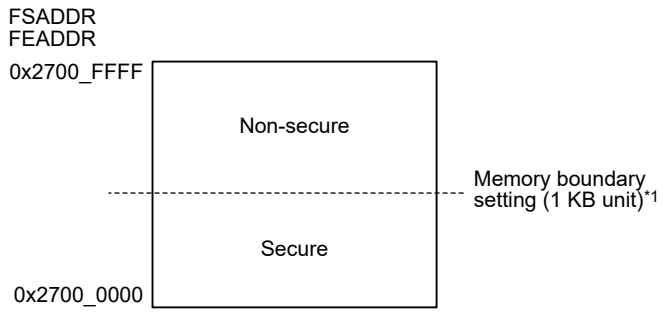
In dual mode, the memory boundary can be set to 0x0200_0000 to 0x0220_0000 in 32 KB unit. When the memory boundary is set to 0x0220_0000 or greater in the dual mode, the entire user area is defined as the Secure region.

Figure 52.39 shows details of the Secure and Non-secure attribute of user area in the code flash.

Note 1. Comparing FSADDR with memory boundary setting of user area and judges whether the target area is in Secure area.

Figure 52.39 Secure and Non-secure region in user area for Secure alias

When the target area of the issuing FACI command is the data area of data flash, the flash sequencer compares the FSADDR/FEADDR register setting with the memory boundary setting of the data flash and determines whether the target area is in Secure region. The memory boundary can be set to 0x0270_0000 to 0x0270_FC00 in 1 KB unit. Figure 52.40 shows details of the Secure and Non-secure attribute of data area in the data flash.



Note 1. Comparing FSADDR/FEADDR with memory boundary setting of data area and judges whether the target area is in Secure area.

Figure 52.40 Secure and Non-secure region in data area for Secure alias

See Figure 52.41 in the details of Secure and Non-secure region of option-setting memory. The flash sequencer judges that target area is Secure region from the FSADDR register setting.

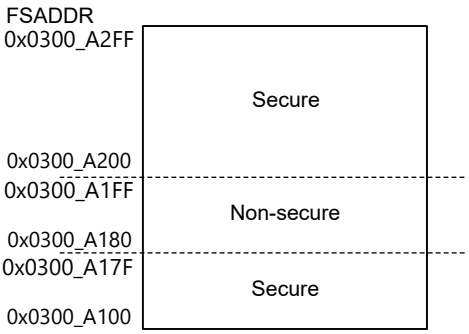


Figure 52.41 Secure and Non-secure region in option-setting memory for Secure alias

52.12.3.2 Protection for Flash Memory Area (Read)

This function protects the Secure region of code flash and data flash from Non-secure bus access.

For details of Secure region, see section 43, Security Features .

52.12.3.3 Protection for Register

The flash sequencer registers have write-access protection against Non-secure access. Table 52.35 shows details of the protected registers of the flash sequencer.

Table 52.35 Protected registers of the flash sequencer for TrustZone

Protection target register	Security attribute setting	Notes
FCKMHZ register	SA register setting (FSAR.FCKMHZSA)	For details of section 52.4.4. FSAR : Flash Security Attribution Register
Flash Access area	SA register setting (FSAR.FACICMISA)	For details of section 52.4.4. FSAR : Flash Security Attribution Register
FACI registers*1 and FWEPROR registers	SA register setting (FSAR.FACICMRSA)	For details of section 52.4.4. FSAR : Flash Security Attribution Register
FCTRCNTR, FCTRLSR FCTRADDR, and FCTRSTATR registers	SA register setting (FSAR.FACITRSA)	For details of section 52.4.4. FSAR : Flash Security Attribution Register
FMEPROT, FCNTSELR, FCNTDATAR0 to 1, FBPROT1 and FSUACR registers	Always Secure	—

Note 1. The target FACI registers are FASTAT, FAEINT, FRDYIE, FSADDR, FEADDR, FBPROT0, FSTATR, FENTRYR, FSUINITR, FCMDR, FBCCNT, FBCSTAT, FPSADDR, FSUASMON, FCPSR, and FPCKAR.

52.12.3.4 Code Flash P/E Mode Entry Protection

The flash sequencer has protection function of code flash P/E by the FMEPROT register for the Secure developer. Secure function can prevent disturbance of reading code flash memory by this protection function. See section 52.4.14. FMEPROT : Flash P/E Mode Entry Protection Register.

For applications that do not require Non-secure region programming/erasure other than from Secure function, it is recommended to always disable Non-secure function of code flash programming/erasure by enabling the protection function of FMEPROT register.

For details, see Figure 52.42 of the code flash P/E sequence example by Non-secure function.

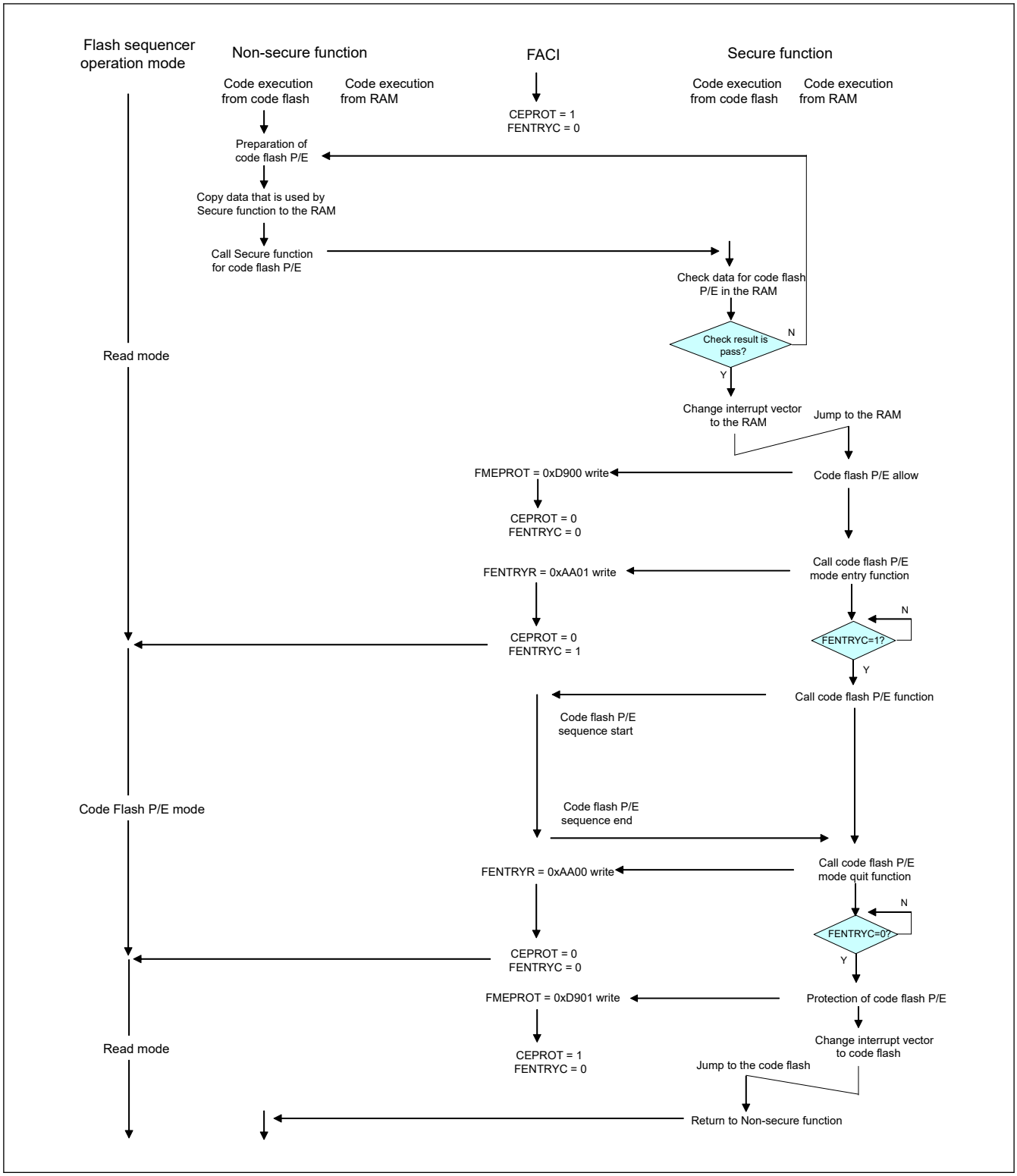


Figure 52.42 Code Flash P/E Sequence Example by Non-secure function (Using Secure function for code flash P/E)

52.12.4 Data Flash Configuration Area Protection

Data flash configuration area has 3 types of lockable area (LK_CD_A0, LK_CD_A1, and LK_CD_A2) by lock bit. When target area of “Configuration set” command includes protection area by lock bit, flash sequencer detects an error and enters "Command Lock" state.

LK_CD_A0 and LK_CD_A1 are protected 4 bytes data by 1 lock bit.

LK_CD_A2 are protected 16 bytes data by 1 lock bit, and 32 bytes (0x2703_0360 to 0x2703_037F) in LK_CD_A2 have writing protection that cannot update except for serial programing mode. For details, see section 52.12.3.1. Protection for Flash Memory Area (P/E). See Table 52.36 in detail of protection condition of lockable area in data flash configuration area.

Table 52.36 Protection Condition of Lockable area in Data Flash Configuration Area for Secure alias

Lockable Area		Protection Size	Lock Bit	Remarks
Area Name	address			
LK_CD_A0	0x2703_0080 to 0x2703_0083	4 bytes	CD0_LK0	In the case of the 16-byte Configuration set command, data cannot be updated if any of the lock bits corresponding to the update address is 0.
	0x2703_0084 to 0x2703_0087	4 bytes	CD0_LK1	
	:	:	:	
	0x2703_0170 to 0x2703_0173	4 bytes	CD0_LK60	
	0x2703_0174 to 0x2703_0177	4 bytes	CD0_LK61	
	0x2703_0178 to 0x2703_017B	4 bytes	CD0_LK62	
	0x2703_017C to 0x2703_017F	4 bytes	CD0_LK63	
LK_CD_A1	0x2703_0180 to 0x2703_0183	4 bytes	CD1_LK0	
	0x2703_0184 to 0x2703_0187	4 bytes	CD1_LK1	
	:	:	:	
	0x2703_0270 to 0x2703_0273	4 bytes	CD1_LK60	
	0x2703_0274 to 0x2703_0277	4 bytes	CD1_LK61	
	0x2703_0278 to 0x2703_027B	4 bytes	CD1_LK62	
	0x2703_027C to 0x2703_027F	4 bytes	CD1_LK63	
LK_CD_A2	0x2703_0280 to 0x2703_028F	16 bytes	CD2_LK0	
	0x2703_0290 to 0x2703_029F	16 bytes	CD2_LK1	
	:	:	:	
	0x2703_0340 to 0x2703_034F	16 bytes	CD2_LK12	
	0x2703_0350 to 0x2703_035F	16 bytes	CD2_LK13	
	0x2703_0360 to 0x2703_036F	16 bytes	CD2_LK14	This lockable area and lock bit cannot be updated except for serial programing mode.
	0x2703_0370 to 0x2703_037F	16 bytes	CD2_LK15	

52.12.5 Anti-Rollback Counter

See section 6.2.21. ARCLS : Anti-Rollback Counter Lock Setting to section 6.2.25. ARC_OEMBLn : Anti-Rollback Counter for OEMBL (n = 0, 1) for Anti-rollback counter related Option flash memory.

Anti-rollback counter cannot be decremented and is used in the firmware update. Only Secure applications are available. Anti-rollback counter shifts one bit to the left with the increment counter command from the initial value of 0. The least

significant bit (bit 0) is set to 1. When anti-rollback counter is maximum value, flash sequencer does not change counter by Increment Counter command (does not occur error).

There are three commands that control the counters. For details, see section 52.9.3.16. Increment Counter Command, section 52.9.3.17. Refresh Counter Command, and section 52.9.3.18. Read Counter Command.

Figure 52.43 shows anti-rollback counter increment image when increment counter command is issued.

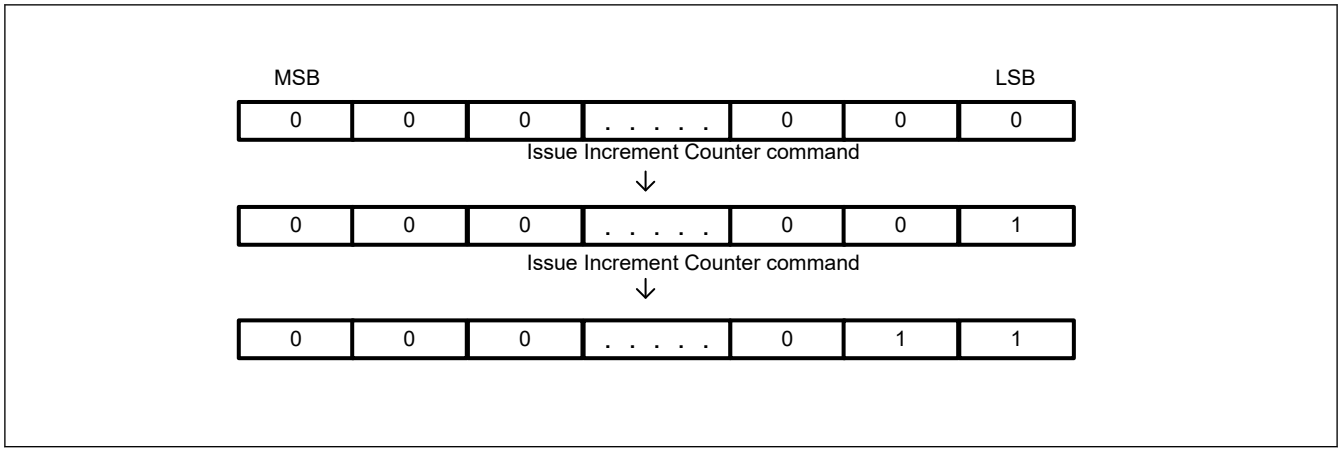


Figure 52.43 Increment image of anti-rollback counter

Anti-rollback counter has 3 types of counters (ARC_SEC, ARC_NSEC, and ARC_OEMBL) and lock bit (ARCSEC_LK, ARCNSEC_LK0 to 3, ARCOEMBL_LK) for protection each counter.

ARC_NSEC can be configured as 1 × 256 or 4 × 64 bits by configuration area setting (CNF_ARCNS0 and CNF_ARCNS0) and cannot be updated in invalid setting.

ARC_OEMBL is used in conjunction with the Secure Boot security feature when updating the OEM boot loader. For details, see section 43.7. Secure Boot. ARC_OEMBL is protected by RSIP-E51A. It can only be incremented when version of the OEMBL update (validated using the RSIP OEMBL integrity check procedure) is higher than the current value stored in FCNTDATAR0 and FCNTDATAR1 registers by the read counter command. For details on RSIP-E51A see section 44, Renesas Secure IP (RSIP-E51A).

52.12.5.1 Anti-Rollback Counter Recovery Flow

Figure 52.44, Figure 52.45 and Figure 52.46-4 shows the recovery flow in the case of OEM_BL anti-rollback counter. The recovery flow for rest anti-rollback counter should be considered necessary processing to depend on customer's application by Secure software developer. As an example, the flow in dual mode is shown. In linear mode, the same operation can be performed using start-up area select, but application size that be managed by anti-rollback counter should be limited to 8 KB.

Example Operation: Update from Bank0 selected in Dual mode

Notation

UCF: Update Complete Flag set/clear by Secure software.
ICF: Increment Complete Flag set/clear by Secure software.
Set status: Indicate that update processing is completed.
Clear status: Indicate that the update processing is not completed.

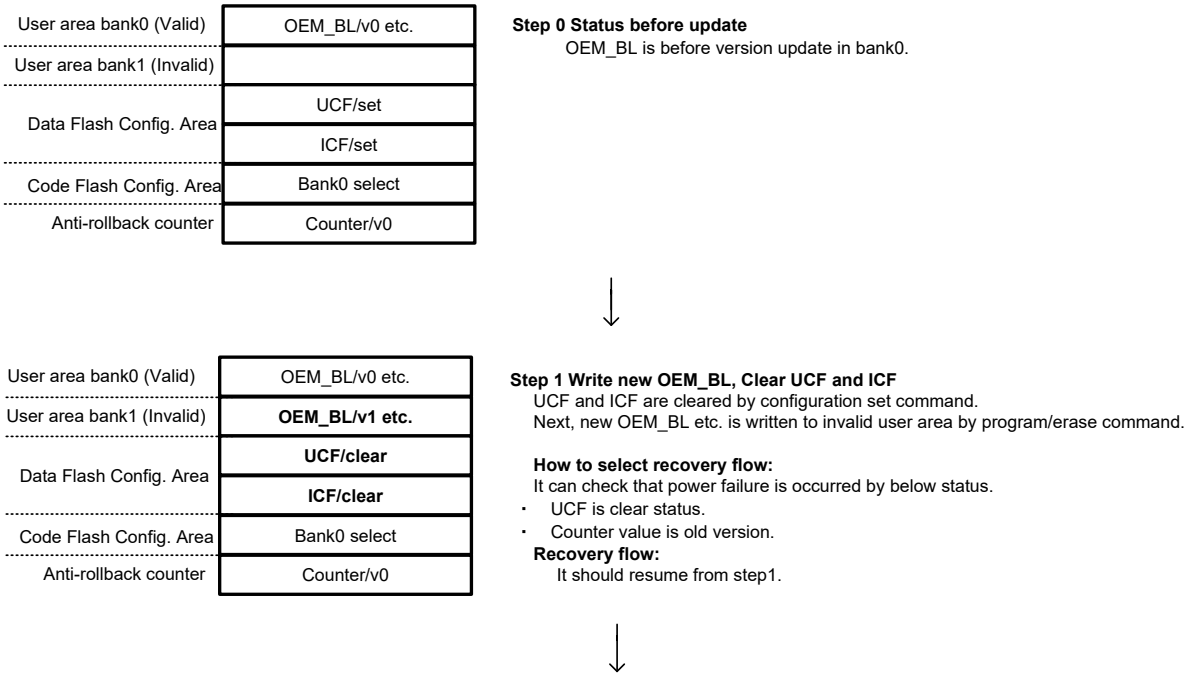
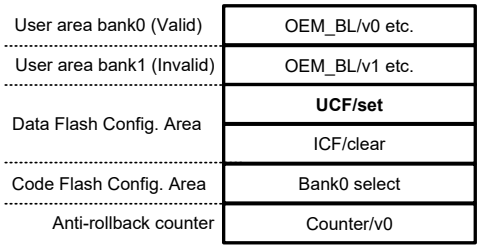


Figure 52.44 Recovery flow of OEM_BL update in Dual mode at power failure (1/3)

Example Operation: Update from Bank0 selected in Dual mode

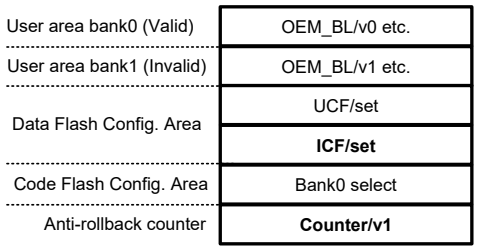
Notation

UCF: Update Complete Flag set/clear by Secure software.
ICF: Increment Complete Flag set/clear by Secure software.
Set status: Indicate that update processing is completed.
Clear status: Indicate that the update processing is not completed.



Step 2 Set UCF
UCF is set by configuration set command.

How to select recovery flow:
Same as step1.
Recovery flow:
Same as step1.



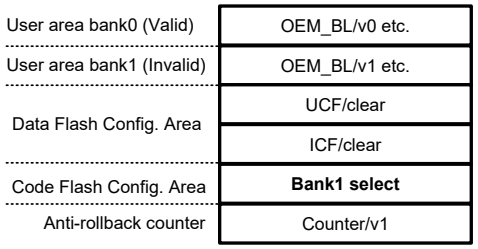
Step 3 Update counter version, Set ICF
OEM_BL version counter is update by increment counter command. Next, ICF is set by configuration set command.

How to select recovery flow(a):
It can check that power failure is occurred by below status.
- UCF is set status.
- ICF is clear status.
- Counter value is old version.

Recovery flow(a):
It should issue refresh counter command. Next, after clearing UCF, it should resume from step1.

How to select recovery flow(b):
It can check that power failure is occurred by below status.
- UCF is set status.
- ICF is clear status.
- Counter value is new version.

Recovery flow(b):
It should issue refresh counter command and set ICF. Next, it should resume from step4.



Step 4 Update bank selection
BANKSWP[2:0] bits are changed to bank1 selection by configuration set command.

How to select recovery flow:
It can check that power failure is occurred by below status.
- UCF is set status.
- ICF is set status.
- Counter value is new version.
- BANKSWP[2:0] bits are bank0 selection.

Recovery flow:
It should resume from step4.

Figure 52.45 Recovery flow of OEM_BL update in Dual mode at power failure (2/3)

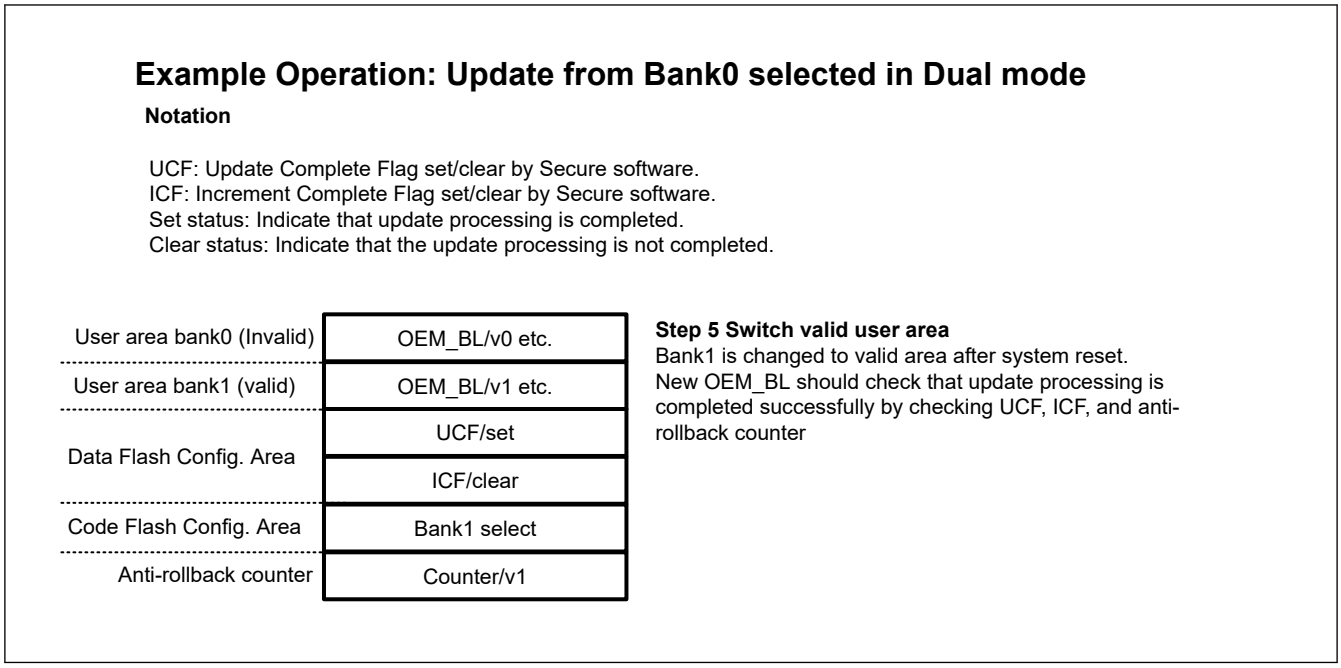


Figure 52.46 Recovery flow of OEM_BL update in Dual mode at power failure (3/3)

52.13 Boot Mode

There are two serial programming modes; the boot mode (for the SCI interface) with SCI9 and the boot mode (for the USB interface) with USBFS. Table 52.37 lists the I/O pins used in boot mode. Table 52.38 lists the available communication interface used in the boot mode.

Table 52.37 I/O Pins Used in Boot Mode

Pin Name	I/O	Mode to be Used	Use
MD	Input	Boot mode (for the SCI interface) Boot mode (for the USB interface) Boot mode (for the JTAG interface) Boot mode (for the SWD interface)	Selection of operating mode
P208/RXD9	Input	Boot mode (for the SCI interface)	For host communication (to receive data through SCI)
P209/TXD9	Output		For host communication (to transmit data through SCI)
USB_DP, USB_DM	I/O	Boot mode (for the USB interface)	Data input/output of USB
USB_VBUS	Input		Detection of connection and disconnection of USB cables
P211/TCK	Input	Boot mode (for the JTAG interface)	Clock pin of JTAG
P210/TMS	Input		Mode Select pin of JTAG
P208/TDI	Input		Data In pin of JTAG
P209TDO	Output		Data Out pin of JTAG
P211/SWCLK	Input	Boot mode (for the SWD interface)	Clock pin of SWD
P210/SWDIO	I/O		Data In-Out pin of SWD

Table 52.38 Available Communication Interface Used in Boot Mode (1 of 2)

Main clock oscillator or external clock is connected	Yes	No	No
Sub clock oscillator is connected*1	Yes or No	Yes	No
Available interface	JTAG, SWD, SCI or USB	JTAG, SWD, SCI or USB	JTAG, SWD or SCI

Table 52.38 Available Communication Interface Used in Boot Mode (2 of 2)

Tool connection time*2	Up to 1 second	Up to 2 seconds	Up to 3 seconds

Note 1. The drive capability of the sub clock oscillator is set to standard by SOMCR.SODRV bit. Note that if you use the crystal corresponding the low drive capability on your board, the crystal may not oscillate in the boot mode.
Note 2. See the boot firmware manual for the detail connection time.

52.13.1 Boot Mode (for the SCI Interface)

In boot mode (for the SCI interface), the host sends control commands and data for programming, and the flash memory is programmed or erased accordingly. An on-chip SCI handles transfer between the host and this MCU in asynchronous mode. Tools for transmission of control commands and the data for programming must be prepared in the host.

When this MCU is activated in boot mode (for the SCI interface), the program on the dedicated area the MCU is executed. The boot program automatically adjusts the bit rate of the SCI and controls programming/erasure by receiving control commands from the host.

Figure 52.47 shows the system configuration for operations in boot mode (for the SCI interface).

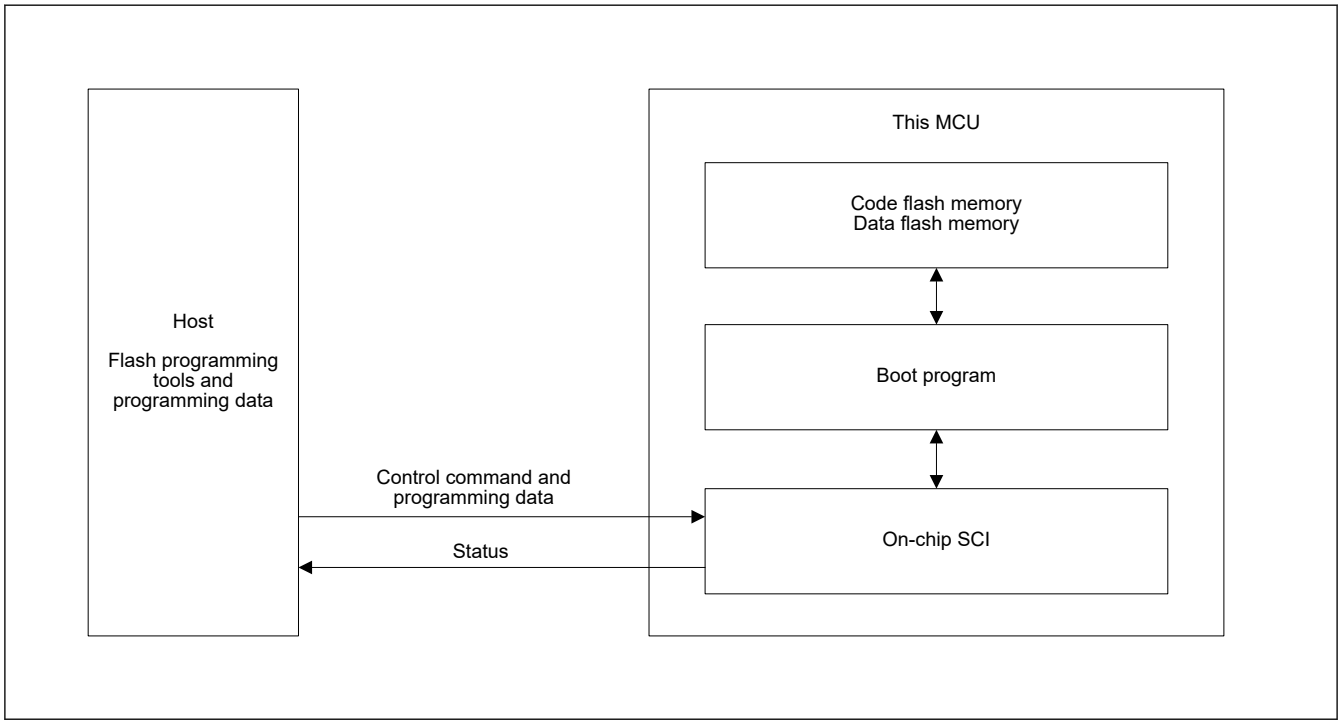


Figure 52.47 System Configuration for Operations in Boot Mode (for the SCI Interface)

52.13.2 Boot Mode (for the USB Interface)

In boot mode (with the USB interface), the flash memory can be programmed or erased by sending control commands and program data from the host. An on-chip USB is used for communications between the host and this MCU. The host requires tools for sending control commands and data for programming. Figure 52.48 shows the configuration of a system for use in boot mode (for the USB interface). The USB cable must be connected on reset release.

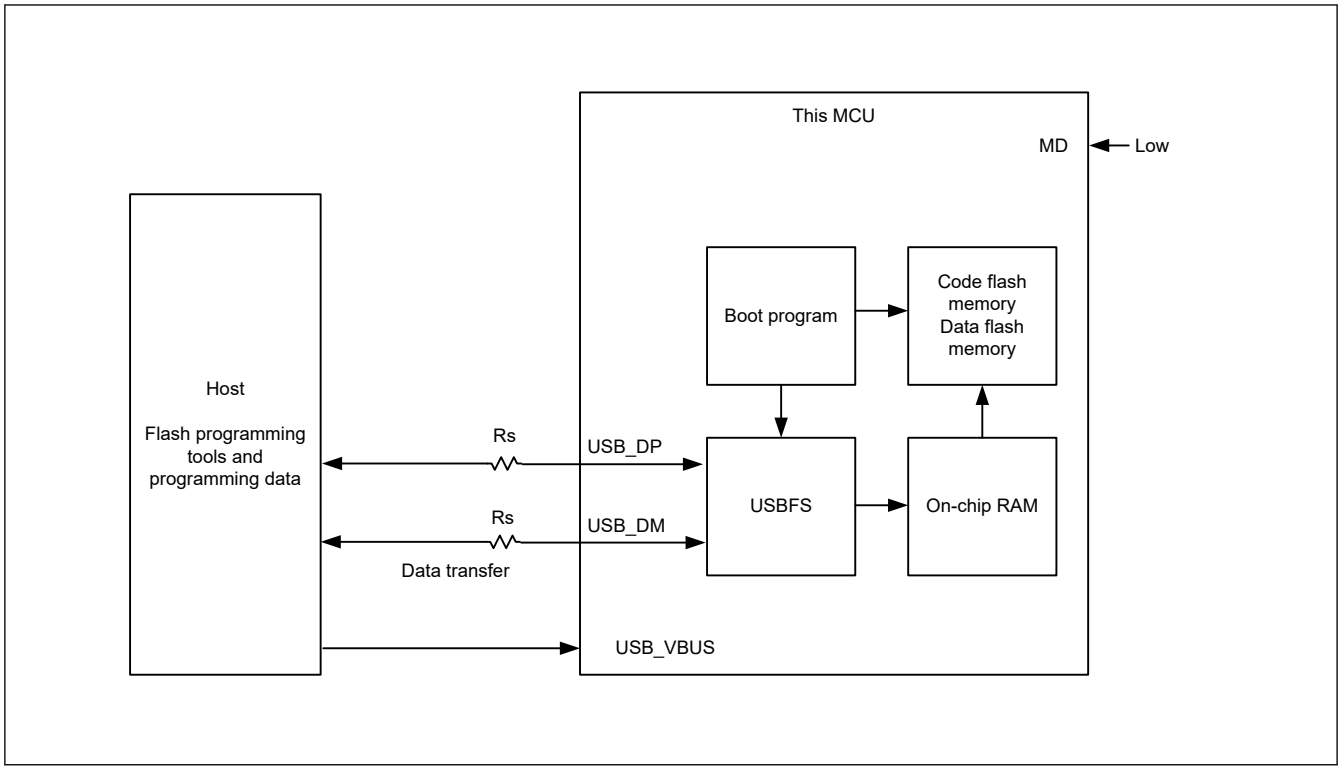


Figure 52.48 System Configuration in Boot Mode (for the USB Interface)

52.14 Using the Serial Programmer for Rewriting

A serial programmer can be used to rewrite flash memory in boot mode.

(1) Serial Programming

This MCU is mounted on the system board at the time of serial programming. Providing a connector to the board enables rewriting of this MCU by the serial programmer to proceed.

52.14.1 Environments for Serial Programming

The recommended environments for rewriting the flash memory of the MCU with data are described below.

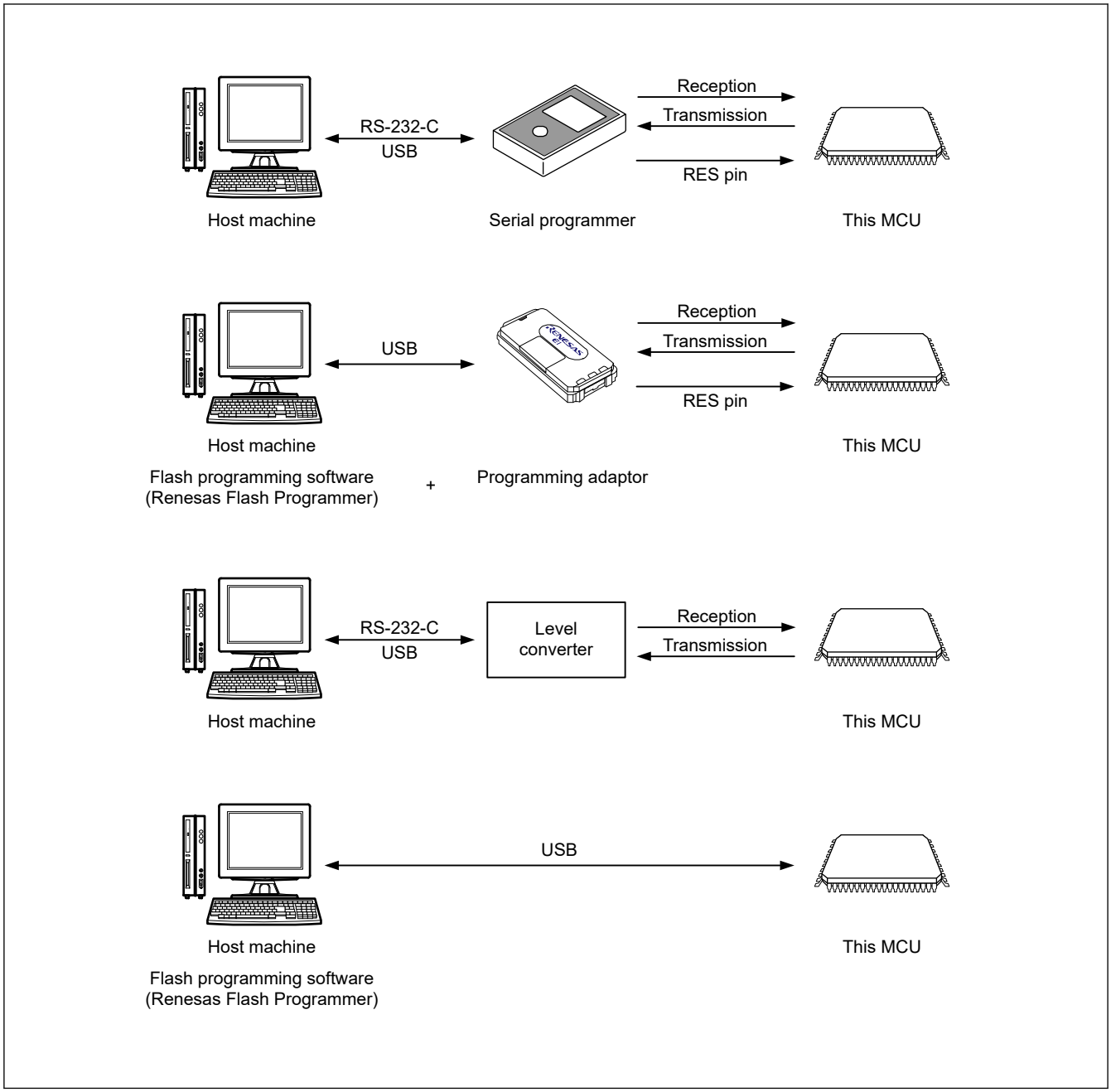


Figure 52.49 Environments for Rewriting the Flash Memory

52.15 Programming through Self-Programming

52.15.1 Overview

This MCU supports programming of the flash memory by the user program itself. The FACI commands can be used with user programs for writing to the flash memory. This allows upgrading of user programs and rewriting of constant data fields.

The program for rewriting must be transferred to the internal RAM or external memory in advance when the BGO is not available or when rewriting the option-setting memory.

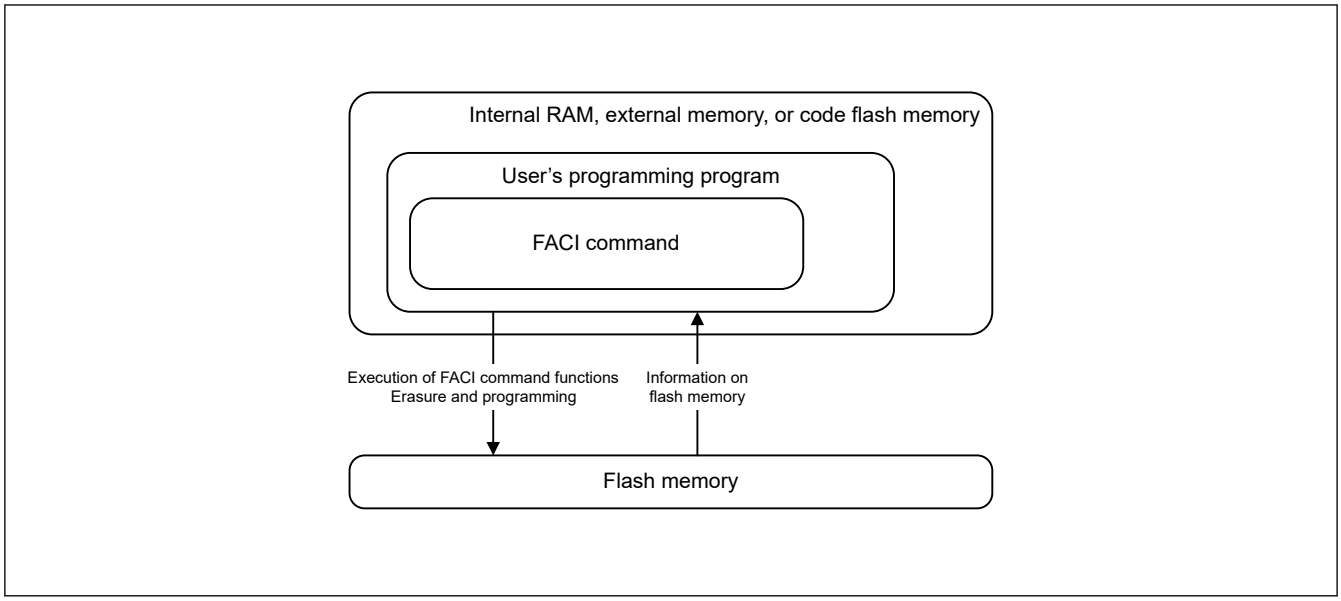


Figure 52.50 Schematic View of Self-Programming

For comprehensive information on the self-programming, see section 52.9. FACI Commands.

52.15.2 Background Operation

The background operation (BGO) can be used to execute the flash rewrite routine on the code flash memory when the data flash memory or other area of code flash memory is rewritten.

Background operations can be used when the combination of the flash memory for rewriting and the flash memory for reading is any of those listed below.

Table 52.39 Conditions under which Background Operation is Usable (1 of 2)

		Range for rewriting	Range for reading
Common to linear and dual modes		Code flash memory	Data flash memory
		Data flash memory	Code flash memory
Liner mode	Products with 2 Mbytes of code flash memory	First half (1 Mbytes) of the code flash memory addresses Secure alias: 0x0200_0000 to 0x020F_FFFF Non-secure alias: 0x1200_0000 to 0x120F_FFFF	Second half (1 Mbytes) of the code flash memory addresses Secure alias: 0x0210_0000 to 0x021F_7FFF Non-secure alias: 0x1210_0000 to 0x121F_7FFF
		Second half (1 Mbytes) of the code flash memory addresses Secure alias: 0x0210_0000 to 0x021F_7FFF Non-secure alias: 0x1210_0000 to 0x121F_7FFF	First half (1 Mbytes) of the code flash memory addresses Secure alias: 0x0200_0000 to 0x020F_FFFF Non-secure alias: 0x1200_0000 to 0x120F_FFFF
	Products with 1 Mbytes of code flash memory	First half (0.5 Mbytes) of the code flash memory addresses Secure alias: 0x0200_0000 to 0x0207_FFFF Non-secure alias: 0x1200_0000 to 0x1207_FFFF	Second half (0.5 Mbytes) of the code flash memory addresses Secure alias: 0x0208_0000 to 0x020F_FFFF Non-secure alias: 0x1208_0000 to 0x120F_FFFF
		Second half (0.5 Mbytes) of the code flash memory addresses Secure alias: 0x0208_0000 to 0x020F_FFFF Non-secure alias: 0x1208_0000 to 0x120F_FFFF	First half (0.5 Mbytes) of the code flash memory addresses Secure alias: 0x0200_0000 to 0x0207_FFFF Non-secure alias: 0x1200_0000 to 0x1207_FFFF

Table 52.39 Conditions under which Background Operation is Usable (2 of 2)

		Range for rewriting	Range for reading
Dual mode	When the BANKSEL_SEC.BANKSWP [2:0] or BANKSEL.BANKSWP[2:0] bits are 111b	Bank 1 area of the code flash memory	Bank 0 area of the code flash memory
	When the BANKSEL_SEC.BANKSWP [2:0] or BANKSEL.BANKSWP[2:0] bits are 000b	Bank 0 area of the code flash memory	Bank 1 area of the code flash memory

52.16 Reading Flash Memory

52.16.1 Reading Code Flash Memory

Special settings are not required to read code flash memory after release from the reset state. Data can simply be read out through access to addresses in the code flash memory.

When reading code flash memory that has been erased but not yet been programming again (i.e. that is in the non-programmed state), all bits are read as 1.

52.16.2 Reading Data Flash Memory

Special settings are not required to read data flash memory after release from the reset state. Data can simply be read out through access to addresses in the data flash memory.

Values read from data flash memory that has been erased but not yet been programming again (i.e. that is in the non-programmed state) are undefined. Use blank checking when you need to confirm that an area is in the non-programmed state.

52.17 Usage Notes

(1) Reading Area Where Programming/Erasure was Interrupted and Area Targeted for Suspension

The data stored in the area where programming or erasure has been suspended or the area where programming or erasure has been suspended by using the suspend command are undefined. To avoid faulty operation caused by reading undefined data, take care not to fetch instructions or read data from areas where programming or erasure was suspended and where programming or erasure was suspended by using the suspend command.

(2) Suspension During Programming/Erasure

When processing of programming/erasure is stopped by issuing the P/E suspend command, the programming/erasure processing can be resumed by issuing the P/E resume command. If the flash sequencer enters the command-locked state for any reason and issues the forced stop command after the suspended processing is normally completed and the ERSSPD flag or PRGSPD flag is set to 1, the suspended processing cannot be resumed. In addition, the values in the area where the processing was suspended are not guaranteed. Erase that area

(3) Prohibition of Additional Programming

Programming a given area of the code flash memory or data flash memory twice is not possible. To program the code flash memory or data flash memory where has been programmed, erase the target area. Programming can be added to the option-setting memory.

(4) Resets During Programming/Erasure, or Blank Checking

In the case of a reset due to the signal on the RES pin during programming/erasure, or blank checking of the flash memory, wait for at least t_{RESW} (see section 60, Electrical Characteristics) of the reset input period once the operating voltage is within the range stipulated in the electrical characteristics, then release the device from the reset state.

(5) Allocation of Vectors for Interrupts and Other Exceptions During Programming/Erasure

Generation of an interrupt or other exception during programming/erasure may lead to fetching of the vector from the code flash memory. Under conditions where BGO cannot be used, set the address of the vector to an address that is not in the code flash memory. Alternatively, make sure that no handling of interrupts or exceptions proceeds during programming/erasure.

(6) Items Prohibited During Programming/Erasure, or Blank Checking

High voltage is applied to the flash memory during programming/erasure, or blank checking. To prevent damage to the flash memory, do not perform the following operations.

- Have the operating voltage from the power supply go beyond the permitted range.
- Change the FWEPROR.FLWE[1:0]bits.
- Change the OPCCR.OPCM[2:0] bits.
- Change the SCKDIVCR.FCK[2:0]bits.
- Change the SCKSCR.CKSEL[2:0]bits.
- Transition to the software standby mode, or Deep Software Standby mode.

(7) Programming/Erasure in Low-Speed Modes

Do not programming/erasure the flash memory when low-speed mode is selected with the operating power control register (OPCCR).

(8) Setting dual bank mode and programming in Boot Mode

The initial mode of the MCU shipped from Renesas is linear mode. In customer's factory when programming dual bank mode and customer's applications on MCU set in linear mode, it is recommended that only area1 in Figure 52.51 is programmed in boot mode and area2 is kept blank. After reset, MCU starts in dual mode and the application is in bank0. Use self-programming when programming area2 for updating in the field. See section 43.8. Field Updating in Dual Mode for details

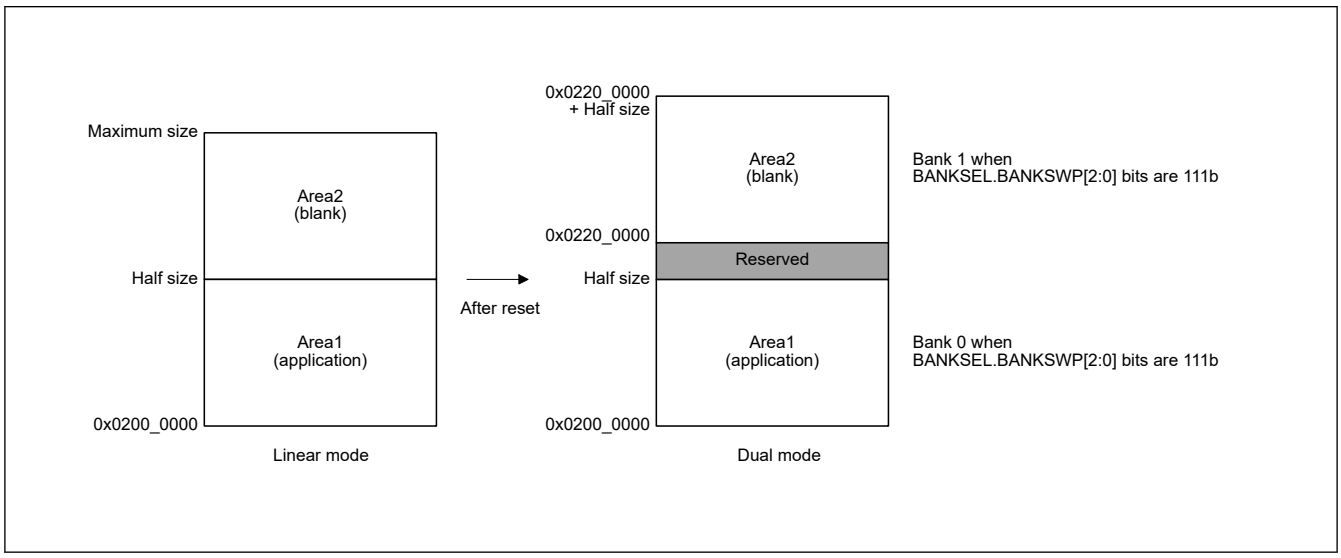


Figure 52.51 Programming the application for the dual mode in the linear mode

(9) Emulator Connection

Renesas provides the emulator which supports both debugging using SWD or JTAG communication and serial programming using SCI communication.

Table 52.40 shows the pinout of 10 pin or 20 pin socket pinouts when using this emulator. The pinout of SWD and JTAG is ARM standard, and MD, TXD, RXD pins are added for the serial programming using SCI communication.

Table 52.40 Pin assign for emulator (1 of 2)

Pin No.	SWD	JTAG	Serial Programming using SCI
1	VCC	VCC	VCC
2	P210/SWDIO	P210/TMS	NC
4	P211/SWCLK	P211/TCK	P201/MD

Table 52.40 Pin assign for emulator (2 of 2)

Pin No.	SWD	JTAG	Serial Programming using SCI
6	P209/SWO	P209/TDO	P209/TXD9
8	P208	P208/TDI	P208/RXD9
9	GNDdetect	GNDdetect	GNDdetect
10	nRESET	nRESET	nRESET
12	P308/TCLK	P308/TCLK	NC
14	P307/TDATA[0]	P307/TDATA[0]	NC
16	P306/TDATA[1]	P306/TDATA[1]	NC
18	P305/TDATA[2]	P305/TDATA[2]	NC
20	P304/TDATA[3]	P304/TDATA[3]	NC
3, 5, 15, 17, 19	GND	GND	GND
7	NC	NC	NC
11, 13	NC	NC	NC

53. Capture Engine Unit (CEU)

53.1 Overview

The Capture Engine Unit (CEU) is a capture module that fetches image data externally input and transfers it to the memory. The CEU is connected to the system bus via bus bridge modules.

Lists the features of CEU as follows.

(1) Image data fetch

- Captures an image output from an external module and writes YCbCr data to the memory with it separated into Y data and CbCr data.
- Fetches image data other than YCbCr data, for example, JPEG data, RGB565, from an externally connected module, such as a camera, and sequentially writes the image data to the memory.
- Fetches an interlace source image in both-field units or one-field units and writes it to the memory. In both-field capture, an image can be stored in the memory as a frame image.

(2) Filter processing

- Performs scale-down and removal of high-frequency components (only in the horizontal direction) for an image using internal filters. Note that the scaled-down image must not exceed VGA. The filter processing can be applied to only YCbCr input data.

(3) Format conversion

- Converts image data input in the YCbCr422 format into the YCbCr420 format and writes it to the memory. Note that the conversion algorithm is simple thinning in which the chrominance component (CbCr) of the evennumbered lines is thinned.

The functional overview of the CEU is shown in Table 53.1 and the main functions and their details are shown in Table 53.2.

Table 53.1 Functional overview of CEU (1 of 2)

Classification	Item	Function	Description	Note
Connectable camera	Size*1	5 megapixels	2,560 pixels × 1,920 lines	Image capture mode or data synchronous fetch mode Horizontal : 8-pixel units Vertical : 4-line units Data enable fetch mode 32-byte units The range of the image size that can be input is as follows. Horizontal: 2,560 pixels to 128 pixels Vertical: 1,920 pixels lines to 96 lines
		3 megapixels	2,048 pixels × 1,536 lines	
		2 megapixels	1,632 pixels × 1,224 lines	
		UXGA	1,600 pixels × 1,200 lines	
		SXGA (1)	1,280 pixels × 1,024 lines	
		SXGA (2)	1,280 pixels × 960 lines	
		WXGA	1,280 pixels × 768 lines	
		XGA	1,024 pixels × 768 lines	
		SVGA	800 pixels × 600 lines	
		WVGA	800 pixels × 480 lines	
		VGA	640 pixels × 480 lines	
		CIF	352 pixels × 288 lines	
		WQVGA	480 pixels × 240 lines	
		QVGA	320 pixels × 240 lines, 240 pixels × 320 lines	
		QCIF	176 pixels × 144 lines	
		QQVGA	160 pixels × 120 lines	
		Sub-QCIF	128 pixels × 96 lines	
	Input format	YCbCr422 8 bits	Cb_0, Y_0, Cr_0, Y_1…	Supports clock ratio of 1:1
			Cr_0, Y_0, Cb_0, Y_1…	
			Y_0, Cb_0, Y_1, Cr_0…	
			Y_0, Cr_0, Y_1, Cb_0…	
		YCbCr422 16 bits	{Y_0, Cb_0}, {Y_1, Cr_0}, …	
			{Y_0, Cr_0}, {Y_1, Cb_0}, …	
		Binary data	Specified amount to be fetched on edges of the sync signal	Written sequentially
			Data is fetched with the horizontal sync signal as an enable signal	
	Horizontal and vertical sync signal polarities	Arbitrary	High-active and low-active	
	Capture start location	Arbitrary	Can be specified in camera input clock units	Horizontal: 1-cycle units Vertical: 1-HD (horizontal sync signal) units
	Number of captured pixels	Arbitrary	Can be specified in 8-pixel units horizontally and in 4-line units vertically	
	Interlace	Both-field capture	Stored as a field image	Capture: 2-VD (vertical sync signal) units
			Stored as a frame image	
		One-field capture	Top field or bottom field can be specified	Capture: 1-VD units
Memory write	Output format	YCbCr422 YCbCr420	YCbCr420 is realized by simple thinning	

Table 53.1 Functional overview of CEU (2 of 2)

Classification	Item	Function	Description	Note
Filter function	No scaling or scale-down	Scale-down of captured display	Desired scaling factor from 1/16 to 1 (scaled-down display must not exceed VGA)	
	Low-pass filter		Removal of high-frequency components	Only in the horizontal direction
Module-stop function	Module-stop state can be set to reduce power consumption.			
TrustZone Filter	Security and Privilege attribution can be set.			

Note 1. The range of the image size that can be input depends on the AC characteristics and the frame rate of the connected device, and the transfer speed to the destination RAM.

Table 53.2 Main functions of CEU and their details

Main function	Detailed description
Image data fetch	• Captures an image output from an external module and writes YCbCr data to the memory with it separated into Y data and CbCr data. • Fetches image data other than YCbCr data, e.g. JPEG data, from an externally connected module, such as a camera, and sequentially writes the image data to the memory. • Fetches an interlace source image in both-field units or one-field units and writes it to the memory. In both-field capture, an image can be stored in the memory as a frame image.
Filter processing	Performs scale-down and removal of high-frequency components (only in the horizontal direction) for an image using internal filters. Note that the scaled-down image must not exceed VGA. The filter processing can be applied to only YCbCr input data.
Format conversion	Converts image data input in the YCbCr422 format into the YCbCr420 format and writes it to the memory. Note that the conversion algorithm is simple thinning in which the chrominance component (CbCr) of the even-numbered lines is thinned.

Figure 53.1 shows a block diagram of the CEU.

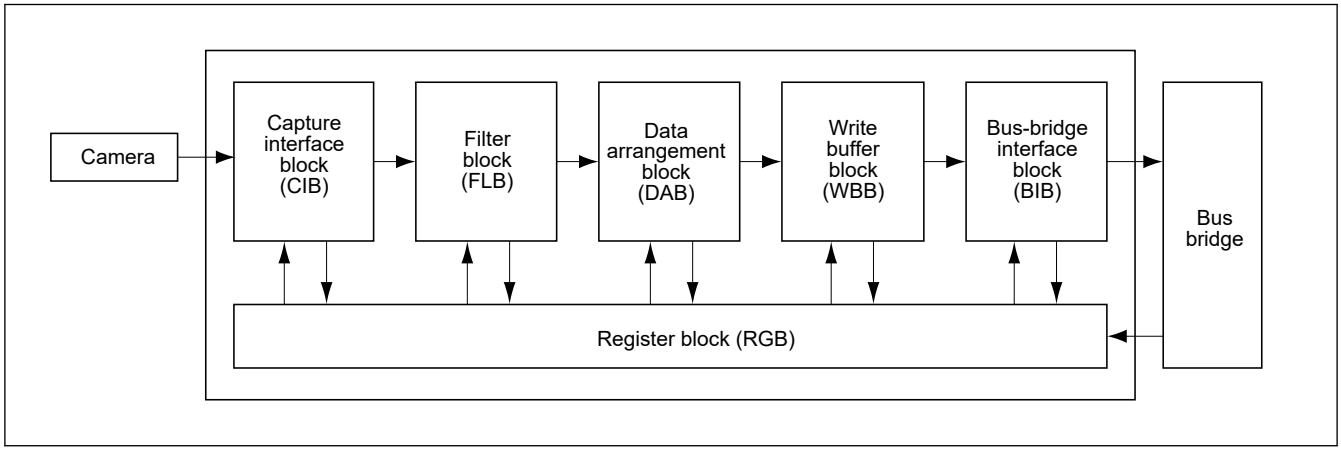


Figure 53.1 Block diagram of CEU

The pin configuration of the CEU is shown in Table 53.3.

Table 53.3 Pin configuration of CEU (1 of 2)

Pin name	Function	I/O	Description
VIO_D15 to VIO_D0*1	CEU data bus	Input	Camera image data input to the CEU
VIO_CLK	CEU clock	Input	Camera clock input to the CEU
VIO_VD	CEU vertical sync	Input	Camera vertical sync signal input to the CEU

Table 53.3 Pin configuration of CEU (2 of 2)

Pin name	Function	I/O	Description
VIO_HD	CEU horizontal sync	Input	Camera horizontal sync signal input to the CEU
VIO_FLD*2	Field signal	Input	Field identification signal to the CEU

Note 1. When the distinction according to the bus width for the data bus is not needed, VIO_D is used in this manual. Otherwise, VIO_D15 to VIO_D0 are used.
Note 2. The VIO_FLD pin is valid only in interlace input mode (CAIFR.IFS=1).

53.2 Register Descriptions

The register configuration of the CEU is shown in Table 53.4.

Most CEU registers have a 2-plane configuration (plane A and plane B). The CEU switches the planes when using these 2-plane registers. A mirror address, which is an address that can always access the register on the unused plane, is provided for each 2-plane register. Figure 53.2 shows the timing to switch the register planes. The CEU switches the register planes at the same time a VD interrupt is asserted.

In the following register descriptions, "during operation" indicates the period that begins when the CEU is activated by the CE bit in the capture start register (CAPSR) and ends when a capture end interrupt (CPE) of the capture event flag clear register (CETCR) occurs. In the read-only bits in each register, the write value should always be 0. If a value other than 0 is written to any of these bits, correct operation cannot be guaranteed.

Table 53.4 Register configuration of CEU (1 of 2)

Register name	Abbr.	R/W	Addresses			Access size
			Address (Plane A)	Address (Plane B)	Mirror Address	
Capture Start Register	CAPSR	R/W	0x4034_8000	—	—	32
Capture Control Register	CAPCR	R/W	0x4034_8004	—	—	32
Capture Interface Control Register*1	CAMCR	R/W	0x4034_8008	—	—	32
Capture Interface Cycle Register*1	CMCYR	R/W	0x4034_800C	—	—	32
Capture Interface Offset Register	CAMOR	R/W	0x4034_8010	0x4034_9010	0x4034_A010	32
Capture Interface Width Register	CAPWR	R/W	0x4034_8014	0x4034_9014	0x4034_A014	32
Capture Interface Input Format Register	CAIFR	R/W	0x4034_8018	—	—	32
CEU Register Control Register	CRCNTR	R/W	0x4034_8028	—	—	32
CEU Register Forcible Control Register	CRCMPR	R/W	0x4034_802C	—	—	32
Capture Filter Control Register	CFLCR	R/W	0x4034_8030	0x4034_9030	0x4034_A030	32
Capture Filter Size Clip Register	CFSZR	R/W	0x4034_8034	0x4034_9034	0x4034_A034	32
Capture Destination Width Register	CDWDR	R/W	0x4034_8038	0x4034_9038	0x4034_A038	32
Capture Data Address Y Register	CDAYR	R/W	0x4034_803C	0x4034_903C	0x4034_A03C	32
Capture Data Address C Register	CDACR	R/W	0x4034_8040	0x4034_9040	0x4034_A040	32
Capture Data Bottom-Field Address Y Register	CDBYR	R/W	0x4034_8044	0x4034_9044	0x4034_A044	32
Capture Data Bottom-Field Address C Register	CDBCR	R/W	0x4034_8048	0x4034_9048	0x4034_A048	32
Capture Bundle Destination Size Register	CBDSR	R/W	0x4034_804C	0x4034_904C	0x4034_A04C	32
Firewall Operation Control Register	CFWCR	R/W	0x4034_805C	—	—	32
Capture Low-Pass Filter Control Register	CLFCR	R/W	0x4034_8060	0x4034_9060	0x4034_A060	32
Capture Data Output Control Register	CDOCR	R/W	0x4034_8064	0x4034_9064	0x4034_A064	32
Capture Event Interrupt Enable Register	CEIER	R/W	0x4034_8070	—	—	32
Capture Event Flag Clear Register	CETCR	R/W	0x4034_8074	—	—	32

Table 53.4 Register configuration of CEU (2 of 2)

Register name	Abbr.	R/W	Addresses			Access size
			Address (Plane A)	Address (Plane B)	Mirror Address	
Capture Status Register	CSTSR	R	0x4034_807C	—	—	32
Capture Data Size Register	CDSSR	R/W	0x4034_8084	—	—	32
Capture Data Address Y Register 2	CDAYR2	R/W	0x4034_8090	0x4034_9090	0x4034_A090	32
Capture Data Address C Register 2	CDACR2	R/W	0x4034_8094	0x4034_9094	0x4034_A094	32
Capture Data Bottom-Field Address Y Register 2	CDBYR2	R/W	0x4034_8098	0x4034_9098	0x4034_A098	32
Capture Data Bottom-Field Address C Register 2	CDBCR2	R/W	0x4034_809C	0x4034_909C	0x4034_A09C	32
CEU Bufferable Write Enable Register	CBWER	R/W	0x4034_80A0	—	—	32

Note: The address shown in this table are Secure region. The address corresponding to the Non-secure region is when b28 of address is set to 1.

Note 1. After changing the setting of a register (CAMCR or CMCYR) that is determined by the external module characteristics, do not start capture until at least 10 external input clock cycles have elapsed.

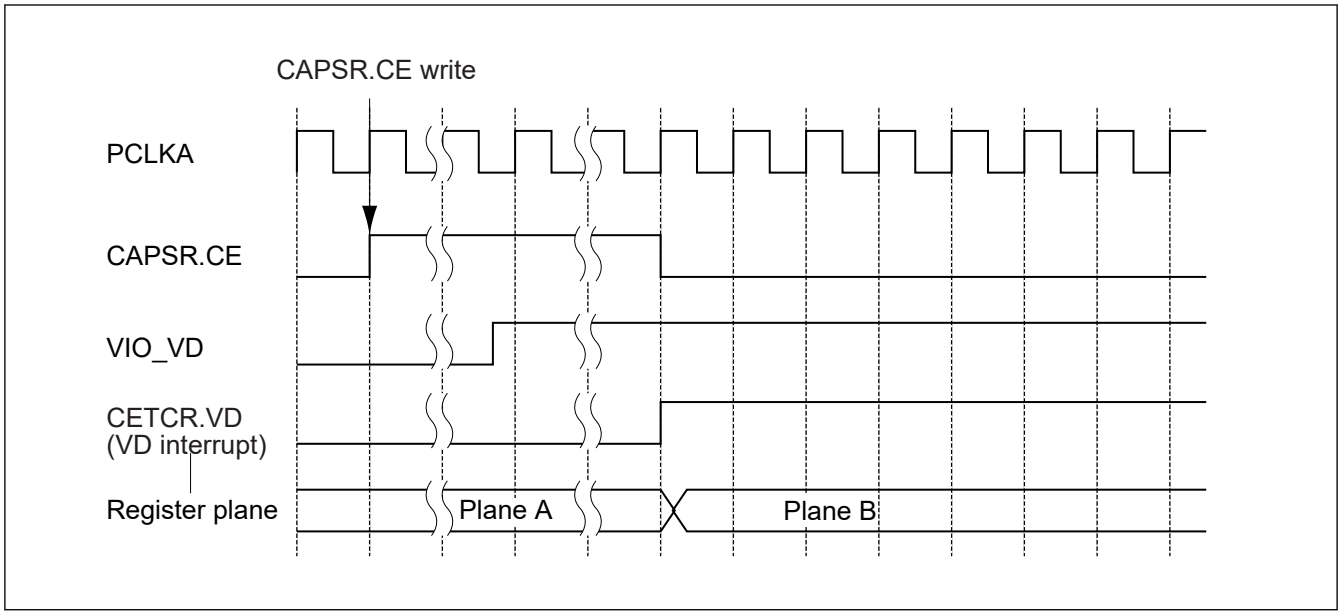


Figure 53.2 Register plane switching timing (VD polarity is high-active in data enable fetch mode)

53.2.1 CAPSR : Capture Start Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPKIL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CE	Capture enable ● Single-capture This bit reserves capturing of the next frame. When 1 is written to this bit, the capture of one frame starts from the next VD input, and stops when the one-frame capture end interrupt (CPE bit in CETCR) is asserted (Figure 53.7). To perform capture again, write 1 to this bit. After the VD or HD polarity is changed, do not write 1 to this bit until the next VD interrupt is asserted. As this bit indicates the capture reserve state, this bit is read as 1 after it is set to 1 and until VD is input. When VD is input, this bit returns to 0 and so is read as 0. The capture end is determined by the one-frame capture end interrupt (CPE bit). This is similar in data fetch mode. Registers should be set before the VD interrupt of the frame where capture starts next. The new register settings take effect at the next VD input. When registers are modified during capturing, the register settings take effect from the capture operations of the next VD input. If a setting register to which writing during capturing is prohibited is modified during capturing, an interrupt source (IGRW bit in CETCR) is generated. For details on the interrupt source, see the description on CETCR. ● Continuous capture When this bit is set to 1 while the CTNCP bit in CAPCR is set to 1, continuous capture starts from the next frame (Figure 53.8). Note that this bit is not cleared to 0 but remains as 1. To stop capturing, clear this bit to 0; capturing stops after the current frame is completed. Continuous capture operations are possible in only image capture mode. The start address of the memory to which the captured data is written to must be set for each frame. 0: Stops capturing 1: Starts capturing	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
16	CPKIL	Software reset of capturing Write 1 to this bit to perform a software reset of capturing. At a software reset, capturing ends immediately without completing capture operation until the end of a frame. Clear the CE bit to 0 when writing 1 to this bit. Processing of the capture software reset is indicated by this bit being set to 1. When CPKIL = 1, do not start capturing since reset processing is in progress. When restarting capture operations, after referring to the CPTON bit in CSTSR to ensure that the CEU is halted (in the idle state), wait until this bit is cleared to 0. The timing of restarting capture operations is shown in Figure 53.6. When a software reset is generated by this bit, a capture end interrupt (CPE bit in CETCR) may be output immediately after the software reset. However, such kind of interrupt should be ignored. Also, even if the capture end interrupt is not output, the interrupt source (CPE bit) must be cleared before capturing of the next frame. 0: Normal state 1: Software reset of capturing	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CAPSR captures data input to the CEU from an external module.

When both the VD (vertical sync signal) and HD (horizontal sync signal) polarities are high-active, one frame is defined as a period from a VD rising edge to the next VD rising edge, and one line as a period from an HD rising edge to the next HD rising edge. Figure 53.3 shows the timing of one frame (when both the VD and HD polarities are high-active).

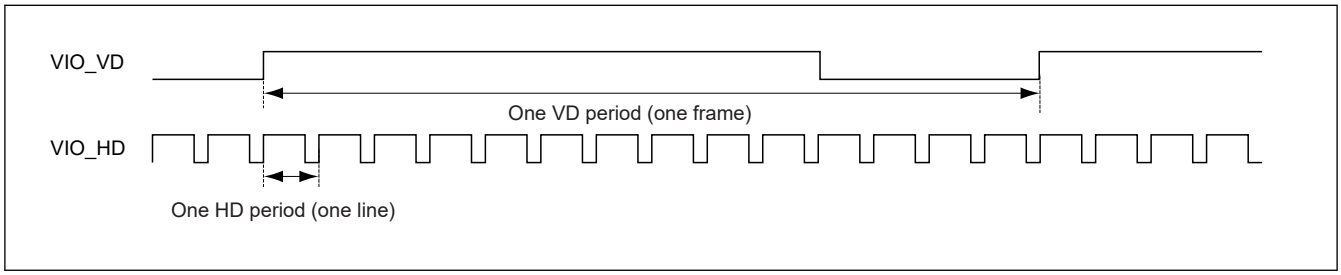


Figure 53.3 Frame timing

When both the VD and HD polarities are high-active, similar to one frame, one field is defined as follows:

- Period from a VD rising edge to the next VD rising edge
- One line is a period from an HD rising edge to the next HD rising edge

The field identification signal FLD should be fixed for at least 1-HD period from a VD input. Figure 53.4 shows the timing of one field (when both the VD and HD polarities are high-active).

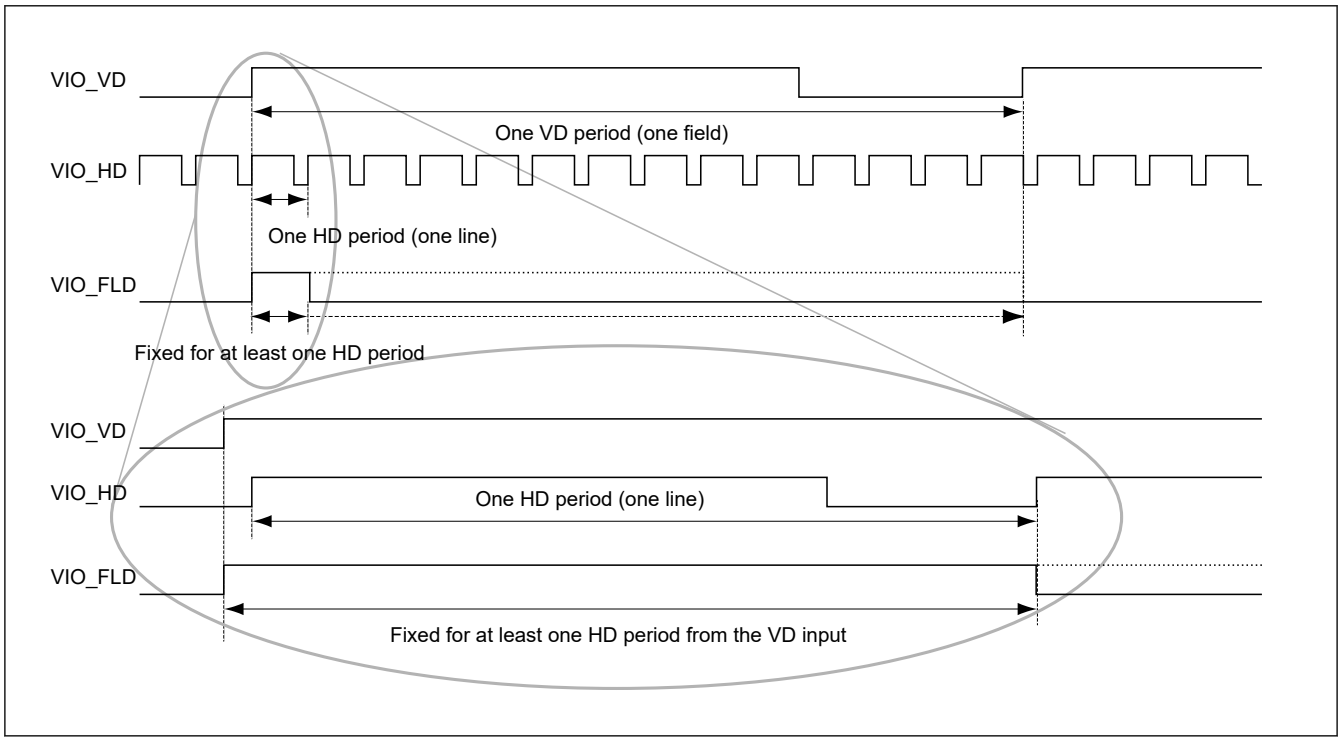


Figure 53.4 One field timing

In data enable fetch, one frame is defined as a period from a VD rising edge to the VD falling edge. With the HD as an enable signal (positive polarity), data of a cycle in which the HD is asserted is fetched while the VD is high. Figure 53.5 shows the timing of one frame for data enable fetch.

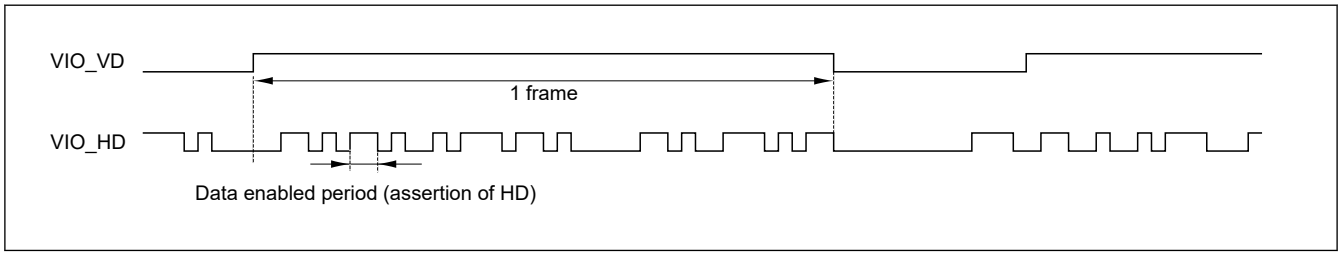


Figure 53.5 Frame timing (data enable fetch)

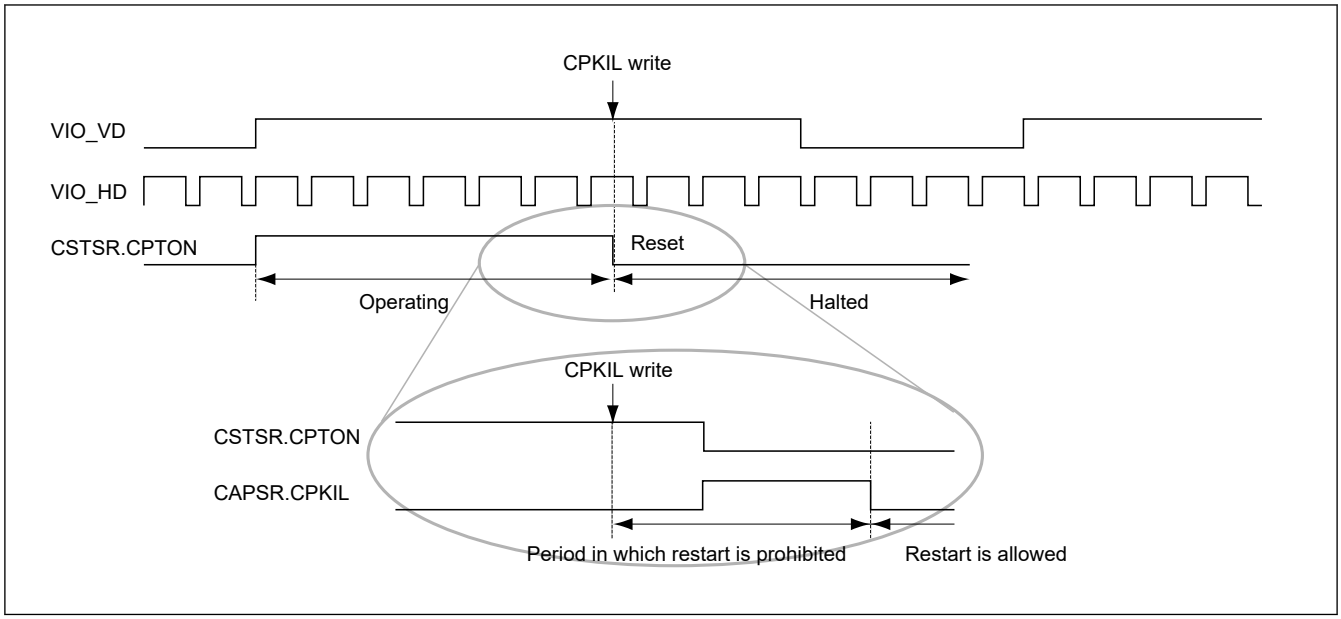


Figure 53.6 Timing of software reset and restart of capturing

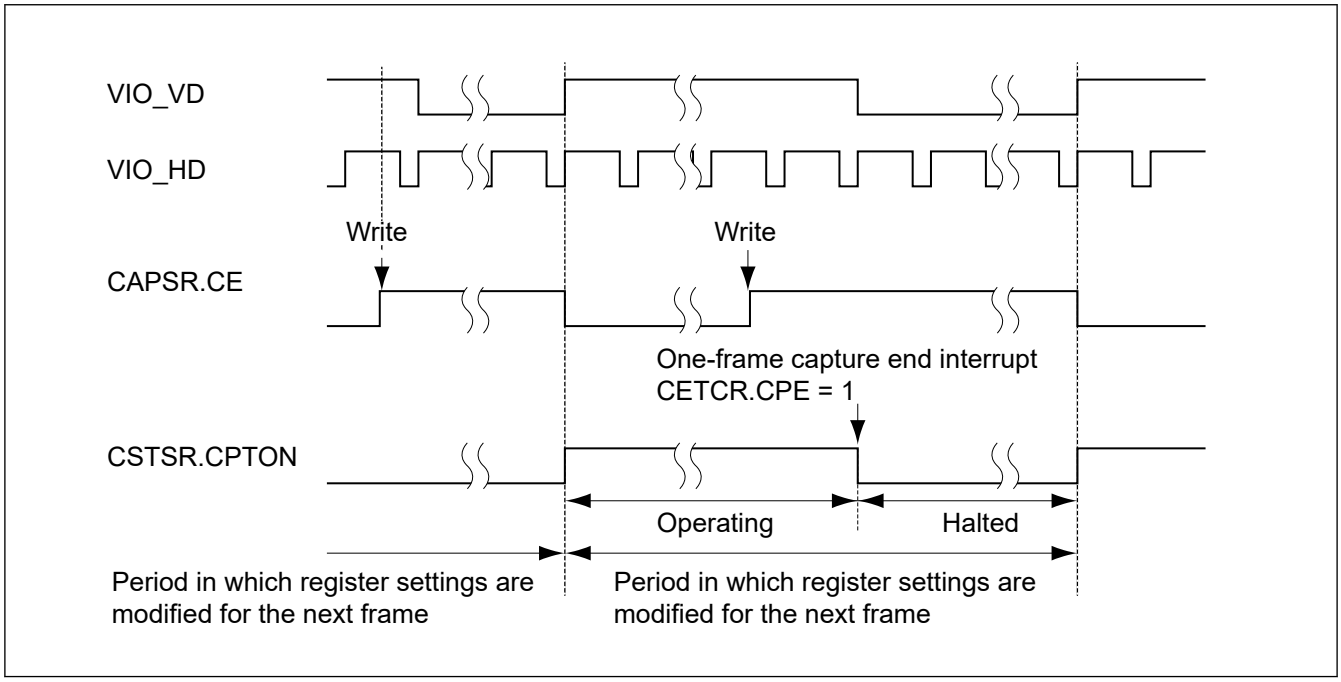


Figure 53.7 Timing of modifying CE bit and register setting in one frame capture

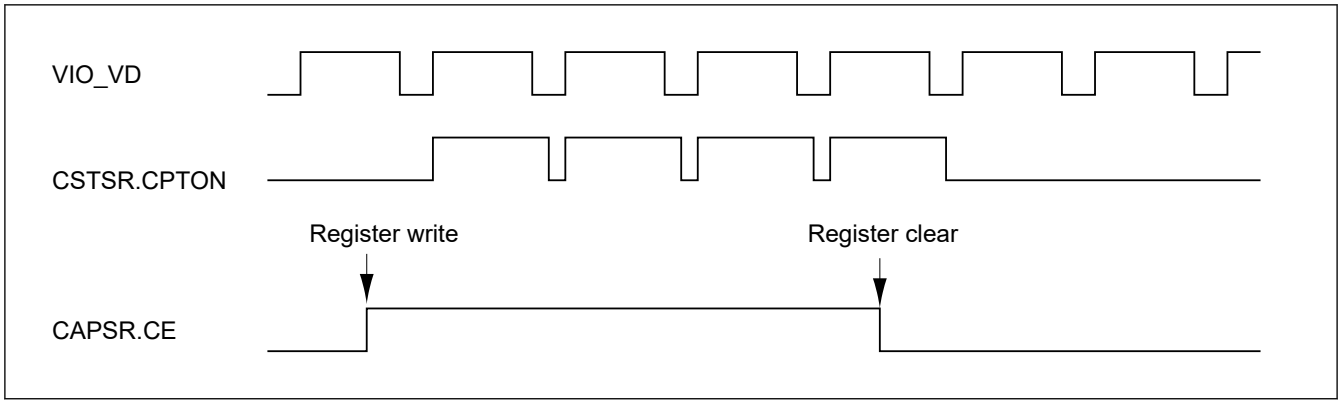


Figure 53.8 Continuous-frame capture

53.2.2 CAPCR : Capture Control Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	FDRP[7:0]								—	—	MTCM[1:0]		—	—	—	CTNCP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	CTNCP	Continuous capture When capturing is started with this bit set to 1, capturing continues until the CE bit in CAPSR is cleared to 0 or a software reset is initiated by the CPKIL bit in CAPSR (see Figure 53.8). Continuous capture must be set before capturing is started. This bit is modified only after 0 is written to the CE bit to stop capturing. If this bit is modified during capturing, correct operation cannot be guaranteed. In data fetch mode, clear this bit to 0. 0: One-frame capture when CAPSR.CE = 1 1: Continuous capture until CAPSR.CE = 0	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
21:20	MTCM[1:0]	Specify the unit for transferring data to a bus bridge module. The efficiency of writing image data can be improved by continuously accessing the addresses. To improve the write efficiency, set these bits to 11. The setting of these bits appear to be unchanged from the outside. 0 0: Image capture: Y data and C data are transferred in 32-byte units Data fetch : Data is transferred in 32-byte units 0 1: Image capture: Y data and C data are transferred in 64-byte units Data fetch : Data is transferred in 64-byte 1 0: Image capture: Y data and C data are transferred in 128-byte units Data fetch : Data is transferred in 128-byte 1 1: Image capture: Y data and C data are transferred in 256-byte units Data fetch : Data is transferred in 256-byte	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
31:24	FDRP[7:0]	Set the frame drop interval in continuous-frame capture. When these bits are cleared to 0, frame drop is not performed, and all frames are captured. Figure 53.9 shows the value set in these bits and the timing of captured frames. The frame drop interval unit differs according to the capture setting. Table 53.5 shows the relationship between the capture setting and frame drop interval unit. The image of the frame drop timing for each capture setting when these bits are set to 2 is shown in Figure 53.10. In both-field capture, capturing is performed continuously for 2-VD periods, regardless of whether the second field is the top field or bottom field. In addition, in both-field capture, the frame drop counter is incremented when the first field has been identified as the top field or bottom field, regardless of whether the second field is the top field or bottom field. When 0 is written to the CE bit in CAPSR, capturing terminates after the current frame has been captured for a capture frame. However, for a drop frame, capturing is forcibly terminated in the CEU so no capture end interrupt (CPE bit in CETCR) is output. While CAPSR.CE = 1, do not change the setting of these bits. Note: Do not change the setting of these bits during continuous capture operations. To change the setting of these bits, stop continuous capture (CAPSR.CE = 0), clear the CTNCP bit (continuous capture) in CAPCR to 0, and then restart continuous capture. Continuous capture is performed during the period of CAPSR.CE = 1 shown in Figure 53.9.	R/W

Note: S-TYPE-3, P-TYPE-3

CAPCR sets continuous-frame capture and the frame drop intervals.

Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to 1.

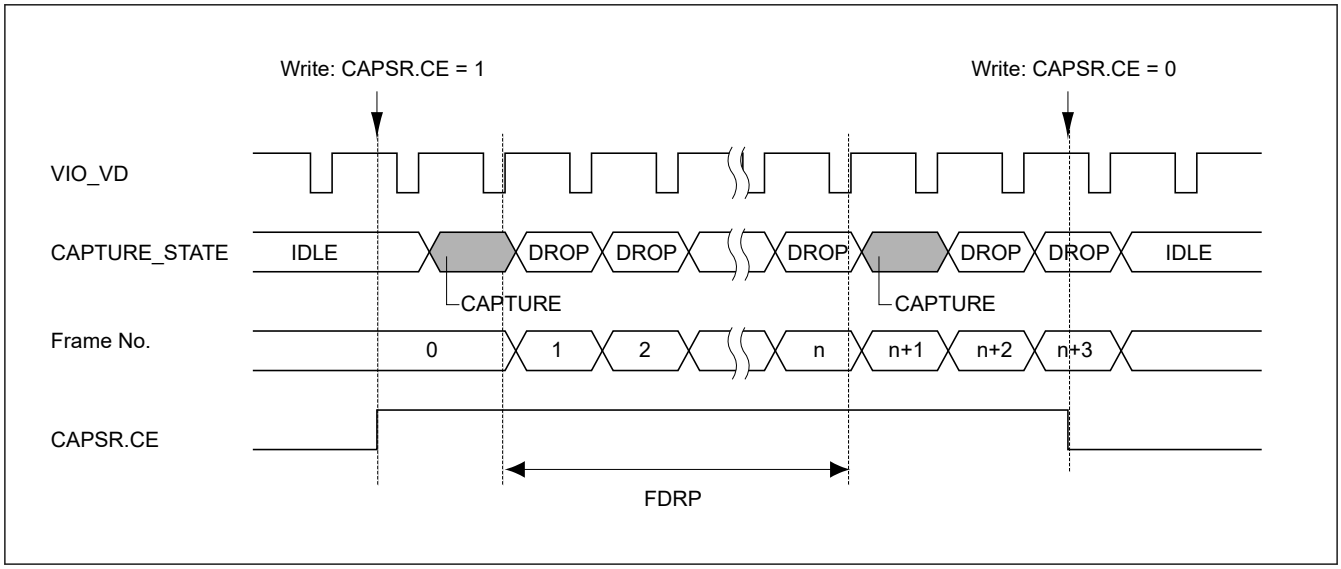


Figure 53.9 Setting of FDRP bits and frame drop timing

Table 53.5 Relationship between capture setting and frame drop interval unit

Input Mode	Captured Image	First Captured Image	Frame Drop Interval Unit	Capture Setting
Progressive	Frame	Frame immediately after capture start	Frame	A
Interlace	Both-field (2-VD capture)	Field immediately after capture start	2 fields (first capture field count)	B
				C
		Top field	2 fields (top-field count)	D
		Bottom field	2 fields (bottom-field count)	E
	One-field (1-VD capture)	Field immediately after capture start	First capture field	F
				G
		Top field	Top field	H
		Bottom field	Bottom field	I

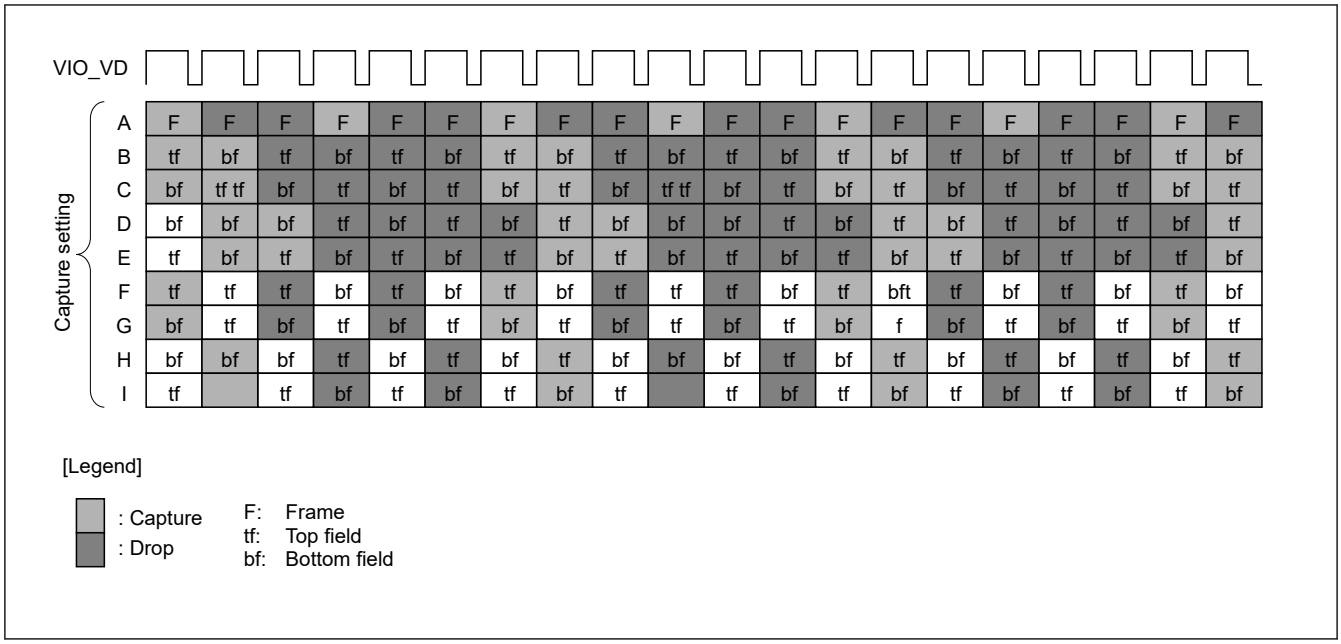


Figure 53.10 Image of frame drop timing for capture settings (FDRP[7:0] = 0x2)

53.2.3 CAMCR : Capture Interface Control Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	VDSEL	HDSEL	FLDSEL	DSEL	—	—	—	—	—	—	—	FLDPOL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DTIF	—	—	DTARY[1:0]		—	—	JPG[1:0]		—	—	VDPOL	HDPOL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	HDPOL	Sets the polarity for detection of the horizontal sync signal (HD) input from an external module. Figure 53.16 shows the relationship between the HD and HD interrupt when high-active is selected. Since an HD interrupt may occur when this bit value is modified, the HD bit in CETCR must always be cleared to 0 when this bit value is changed. In data enable fetch mode, set this bit to 0. 0: Horizontal sync signal (HD) from an external module is detected as high active 1: Horizontal sync signal (HD) from an external module is detected as low active	R/W
1	VDPOL	Sets the polarity for detection of the vertical sync signal (VD) input from an external module. Figure 53.14 and Figure 53.15 show the relationship between the VD, HD and VD interrupt when high-active is selected. Since a VD interrupt may occur when this bit value is modified, the VD bit in CETCR must always be cleared to 0 when this bit value is changed. In data enable fetch mode, this bit is not used and the sense for detection is always active high. 0: Vertical sync signal (VD) from an external module is detected as high active 1: Vertical sync signal (VD) from an external module is detected as low active	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
5:4	JPG[1:0]	These bits select the fetched data type. 0 0: Image capture mode (input data are separated into the luminance component data (Y) and the chrominance component data (CbCr) for output to the memory) 0 1: Data synchronous fetch mode (specified size of input data are output to the specified memory addresses in order of input and in synchronization with the sync signal) 1 0: Data enable fetch mode (input data are fetched with HD as an enable signal and output to the specified addresses in memory in order of input) 1 1: Setting prohibited	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
9:8	DTARY[1:0]	Set the input order of the luminance component and chrominance component. The order in which Y and CbCr are input from an external module differs among modules. The CEU supports the input orders shown in Figure 53.12. Set the corresponding value in these bits. In data fetch mode, set these bits to 00. 0 0: 8-bit interface: Image input data is fetched in the order of Cb_0, Y_0, Cr_0, and Y_1 16-bit interface: Image input data is fetched in the order of {Cb_0, Y_0} and {Cr_0, Y_1} 0 1: 8-bit interface: Image input data is fetched in the order of Cr_0, Y_0, Cb_0, and Y_1 16-bit interface: Image input data is fetched in the order of {Cr_0, Y_0} and {Cb_0, Y_1} 1 0: 8-bit interface: Image input data is fetched in the order of Y_0, Cb_0, Y_1, and Cr_0 16-bit interface: Image input data is fetched in the order of {Y_0, Cb_0} and {Y_1, Cr_0} 1 1: 8-bit interface: Image input data is fetched in the order of Y_0, Cr_0, Y_1, and Cb_0 16-bit interface: Image input data is fetched in the order of {Y_0, Cr_0} and {Y_1, Cb_0}	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	DTIF	Sets the digital image input pins from which data is to be captured. 0: Data input to 8-bit digital image input pins is captured 1: Data input to 16-bit digital image input pins is captured	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	FLDPOL	Sets the polarity of the field identification signal (FLD) from an external module. 0: When the FLD signal is high-active, the field is detected as the top field and when low-active, the field is detected as the bottom field. 1: When the FLD signal is low-active, the field is detected as the top field and when high-active, the field is detected as the bottom field.	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
24	DSEL	Sets the edge for fetching the image data (D15 to D0) from an external module. 0: D15 to D0 are fetched at the rising edge of the camera clock. 1: D15 to D0 are fetched at the falling edge of the camera clock.	R/W
25	FLDSEL	Sets the edge for capturing FLD from an external module. 0: FLD is captured at the rising edge of the camera clock. 1: FLD is captured at the falling edge of the camera clock.	R/W
26	HDSEL	Sets the edge for capturing HD from an external module. 0: HD is captured at the rising edge of the camera clock. 1: HD is captured at the falling edge of the camera clock.	R/W
27	VDSEL	Sets the edge for capturing VD from an external module. 0: VD is captured at the rising edge of the camera clock. 1: VD is captured at the falling edge of the camera clock.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CAMCR sets the capture interface.

The following items are set by CAMCR.

- Selection between the rising edge or falling edge of the camera clock for signal capture or data fetch operation
- Selection between image capture operation or data fetch operation
- Polarities of the vertical and horizontal sync signals
- Input order of image data components (Y, Cb, and Cr) (only for image capture mode)
- Selection of digital image input pins (8 bits or 16 bits)

- Polarity of the field identification signal

CAMCR must be set according to the module connected. In data fetch mode, set the DTARY bits to 0. Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to 1.

Note: After changing the setting of this register, do not start capture until at least 10 external input clock cycles have elapsed.

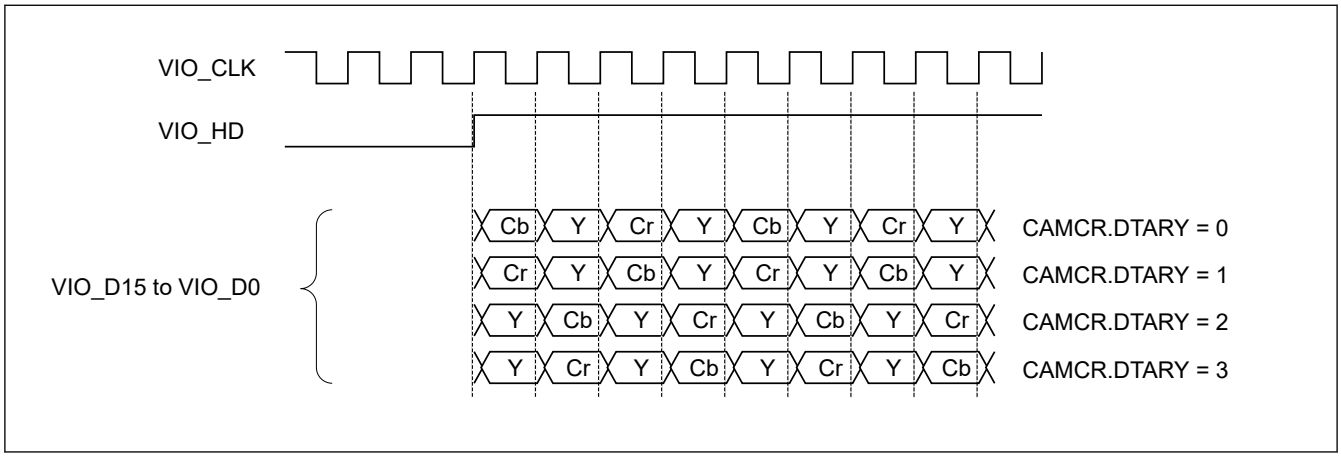


Figure 53.11 Input order of image data

The JPG bit in CAMCR selects whether digital image data is fetched or data such as JPEG is fetched. In addition, when data such as JPEG is fetched, select whether the specified amount of data is continuously fetched in synchronization with the sync signal or data is fetched while the horizontal sync signal is enabled.

In data enable fetch mode, one frame is defined as a period from the rising edge to the falling edge of the vertical sync signal (VD) for data fetching. The horizontal sync signal (HD) is enabled only when the VD is high and treated as an enable signal. Data input in the cycle in which the HD is asserted (high) is fetched and output to the memory continuously.

This module starts fetching data at the rising edge of the VD and stops fetching data at the falling edge of the VD in data enable fetch mode. Thus, if the VD remains high and does not go low, end processing does not start. In addition, if the VD remains high and the HD also remains asserted, data continues to be fetched.

Figure 53.12 and Figure 53.13 show the interface timing in data enable fetch mode.

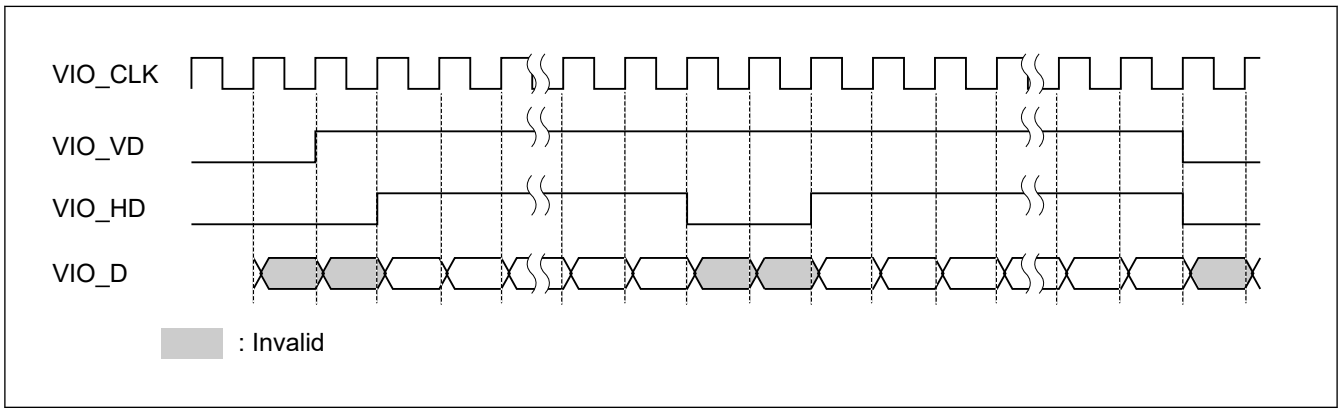


Figure 53.12 Data enable fetch timing (HD asserted (high) while VD is high)

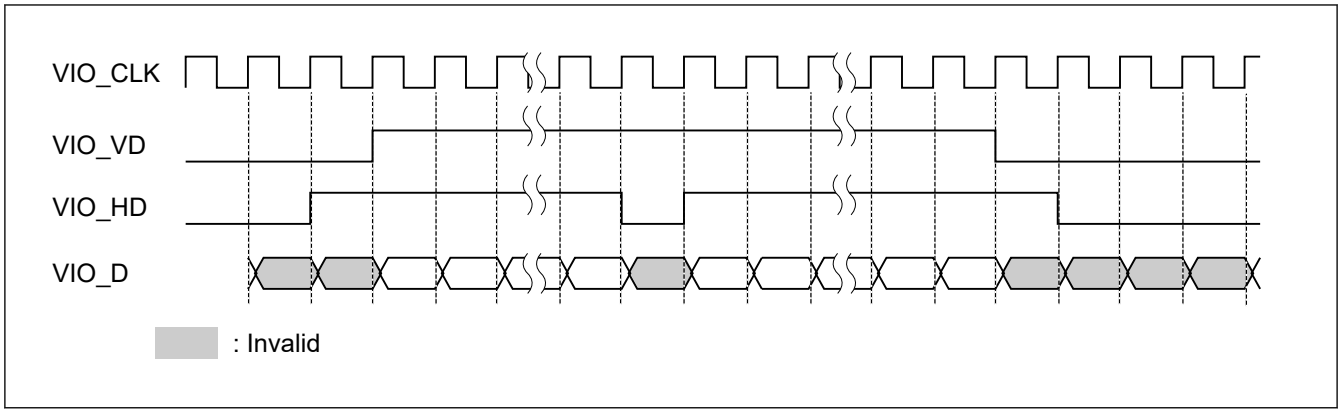


Figure 53.13 Data enable fetch timing (HD asserted (high) when VD is not high)

In data enable fetch mode, this module generates a VD interrupt in response to detection of the active level of VD. In image capture mode and data synchronous fetch mode, this module generates a VD interrupt in response to the first detection of the active level of HD following detection of the active level of VD. Note that, when VD and HD are asserted and detected at the same time, this module generates a VD interrupt at that time.

Figure 53.14 to Figure 53.16 show the relationships between the VIO_VD signal and the VD interrupt, the VIO_VD and VIO_HD signals and the VD interrupt, and the VIO_HD signal and the HD interrupt.

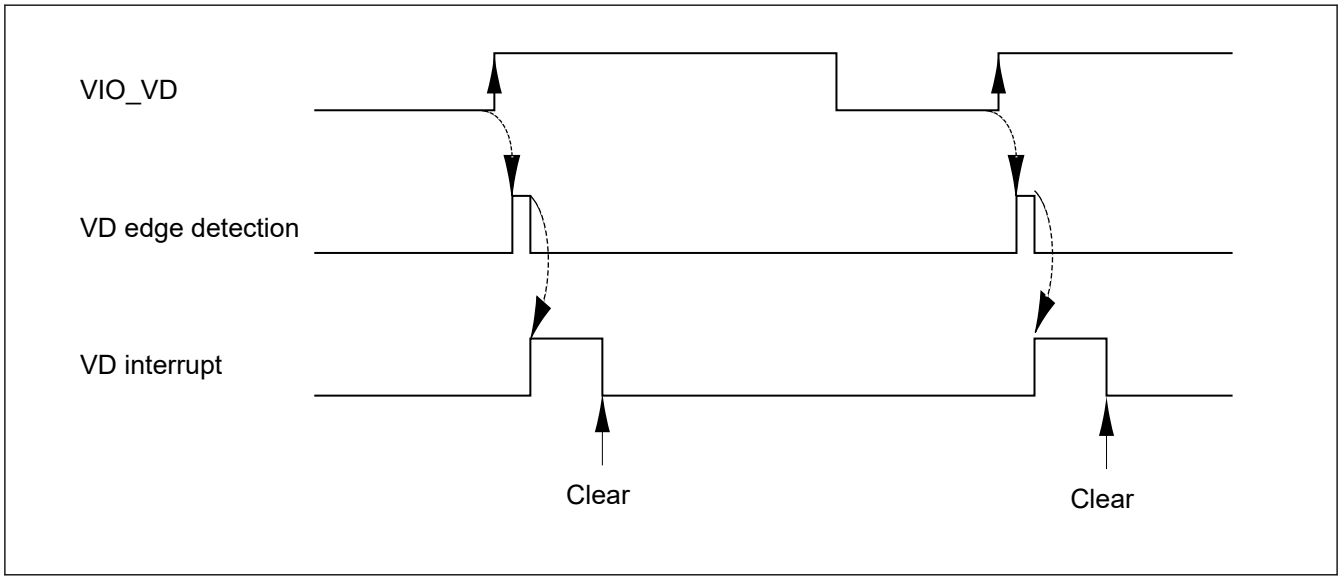


Figure 53.14 Relationship between VIO_VD and VD interrupt when VD is high-active (in data enable fetch mode)

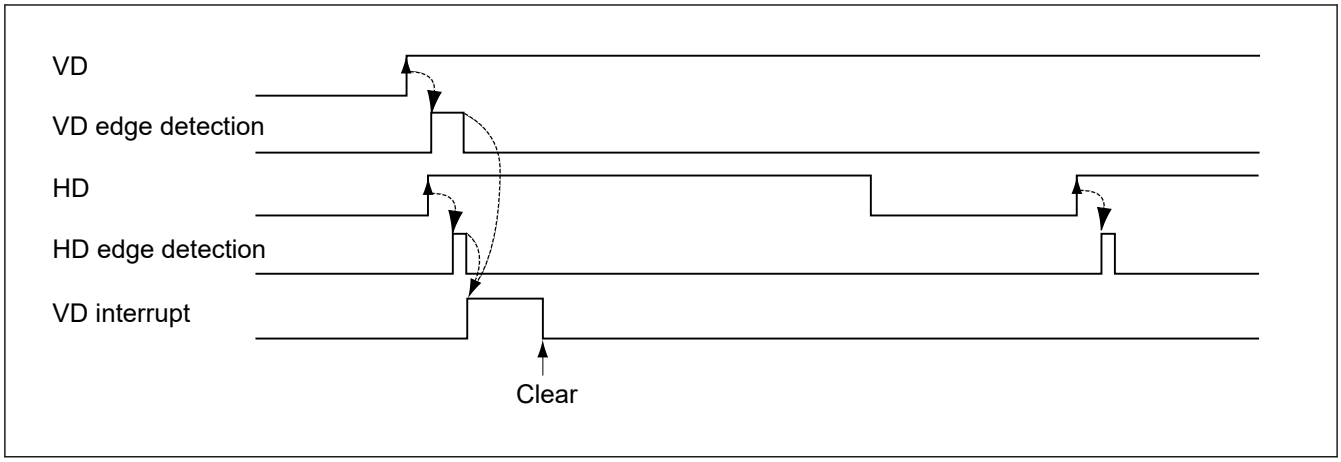


Figure 53.15 Relationship between the VIO_VD and VIO_HD signals and the VD Interrupt when VD and HD are high-active (in Image capture mode or data synchronous fetch mode)

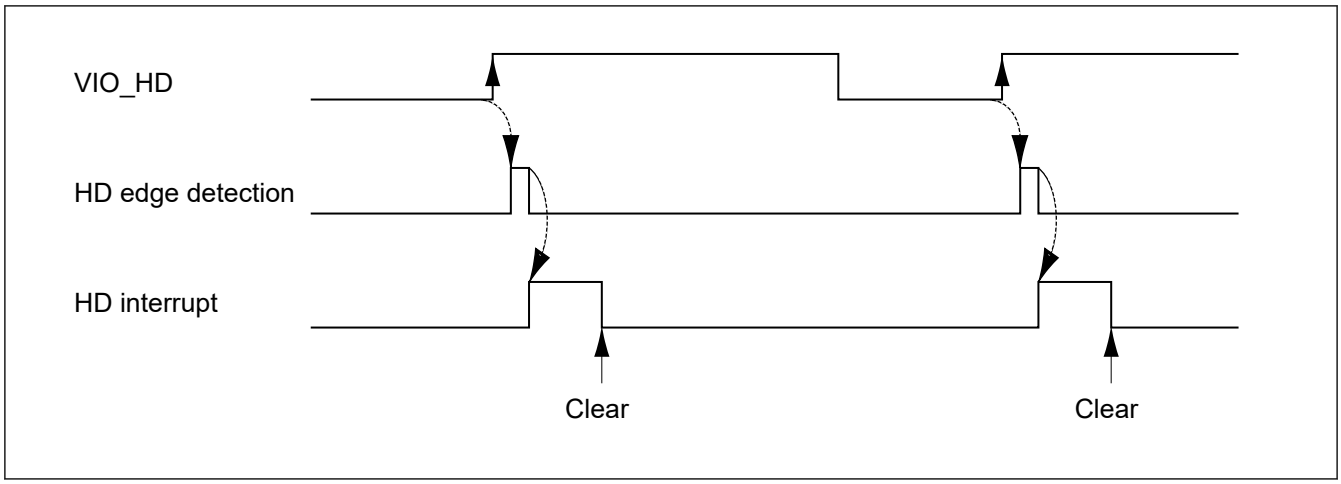


Figure 53.16 Relationship between VIO_HD and HD interrupt when HD is high-active

53.2.4 CMCYR : Capture Interface Cycle Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x000C

Bit position:	31	29														16	13													0
Bit field:	—	—	VCYL[13:0]														—	—	HCYL[13:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	HCYL[13:0]	Horizontal Cycle Count of External Module These bits set the number of HD cycles of an external module. The interrupt source bit IGHS in CETCR is set to 1 when the actual number of HD cycles input from the external module differs from this setting. Set these bits for detecting an illegal HD. When these bits are all cleared to 0, the interrupt source bit IGHS in CETCR is not set to 1. Though the interrupt source bit IGHS in CETCR may be set to 1 after the HDPOL bit (HD polarity) in CAMCR is changed, this interrupt should be ignored.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
29:16	VCYL[13:0]	Vertical HD Count of External Module These bits set the number of VD cycles of an external module with the number of HD inputs. The interrupt source bit IGVS in CETCR is set to 1 when the actual number of VD cycles input from the external module differs from this setting. Set these bits for detecting an illegal VD. When these bits are all cleared to 0, the interrupt source bit IGVS in CETCR is not set to 1. Though the interrupt source bit IGVS in CETCR may be set to 1 after the VDPOL bit (VD polarity) in CAMCR is changed, this interrupt should be ignored.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CMCYR is used to detect an illegal VD and an illegal HD. For HD, the number of cycles from a rising edge of HD to the next rising edge is set (falling edges when low-active is selected for HD). For VD, the number of HD inputs from a rising edge of VD to the next rising edge is set (falling edges when low-active is selected for VD).

Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to 1.

Set 0 in all bits of this register, during data enable fetch mode.

Note: After changing the setting of this register, do not start capture until at least 10 external input clock cycles have elapsed.

53.2.5 CAMOR, CAMOR_x : Capture Interface Offset Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0010 (CAMOR)
0x1010 (CAMOR_B)
0x2010 (CAMOR_M)

Bit position:	31				27											16				12												0
Bit field:	—	—	—	—	VOFST[11:0]												—	—	—	HOFST[12:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
12:0	HOFST[12:0]	Specify the capture start location in terms of the number of clock cycles from a horizontal sync signal (1-cycle units). The blanking period from a horizontal sync signal differs among external modules. Therefore, the horizontal capture start location must be specified by these bits in terms of external input clock cycles from a horizontal sync signal so that an image can be captured from the valid image area. This is similar in data synchronous fetch mode (see Figure 53.19). Some external modules output a horizontal sync signal as a data enable signal. In this case, there is no blanking period so these bits must be cleared to 0 (see Figure 53.20). Note: The first HD (horizontal sync signal) being input simultaneously or after the first VD (vertical sync signal) is the operating condition of the CEU. These inputs are affected by the polarities (set by the VDPOL and HDPOL bits in CAMCR).	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
27:16	VOFST[11:0]	Specify the capture start location in terms of the HD count from a vertical sync signal (1-HD units). The blanking period from a vertical sync signal differs among external modules. Therefore, the vertical capture start location must be specified by these bits in terms of the HD count from a vertical sync signal so that an image can be captured from the valid image area (see Figure 53.17). Some external modules output a vertical sync signal as a data enable signal. In this case, there is no blanking period so these bits must be cleared to 0 (see Figure 53.18).	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CAMOR sets the location to start capturing when capturing images.

Since the number of HD (horizontal sync signal) inputs from a VD (vertical sync signal) input to the start of a valid image period, and the number of clock cycles from an HD input to the start of a valid image period differ among external modules, these must be set in CAMOR. By setting a value greater than the valid image area, part of the image can be clipped for capture. In data synchronous fetch mode, the setting of this register becomes the number of cycles (HD count) up to the start of a valid data period.

This register is not used, during data enable fetch mode.

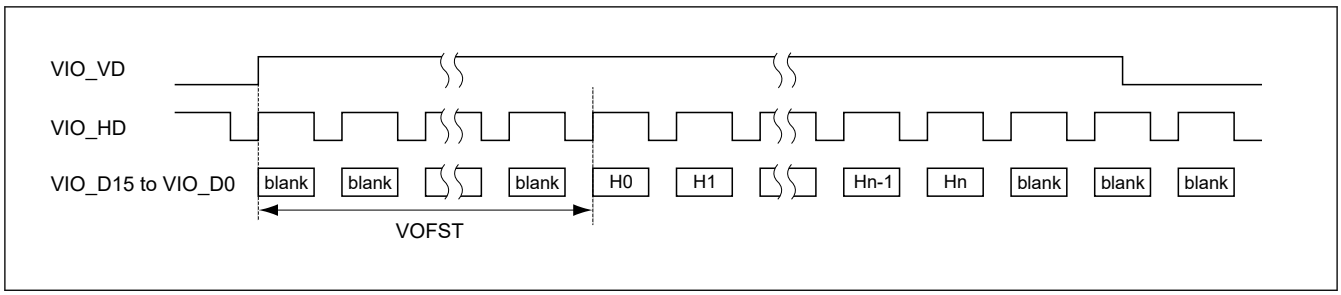


Figure 53.17 Vertical offset

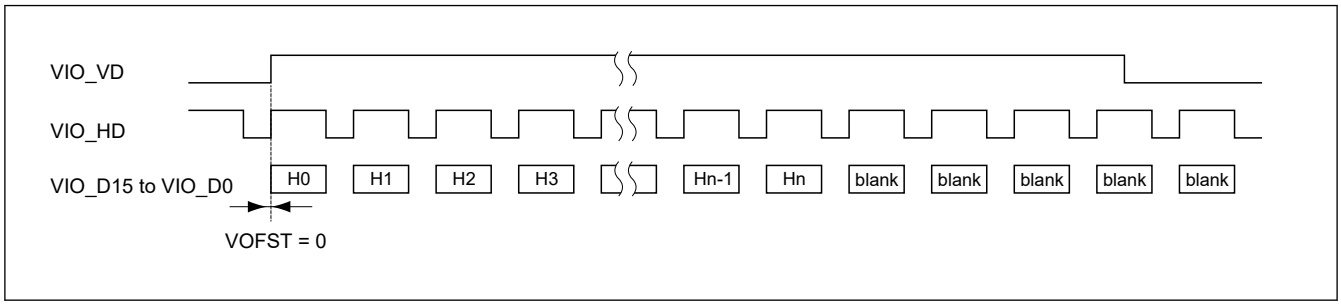


Figure 53.18 Timing when VD is data enable signal

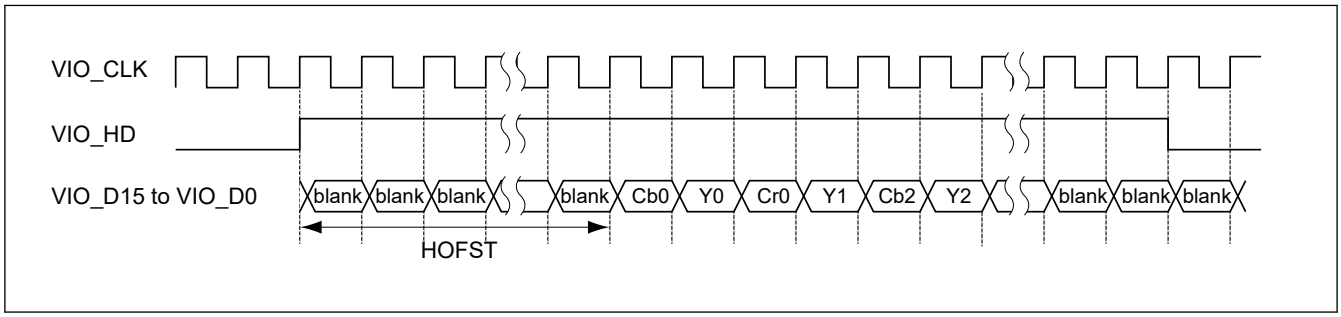


Figure 53.19 Horizontal offset

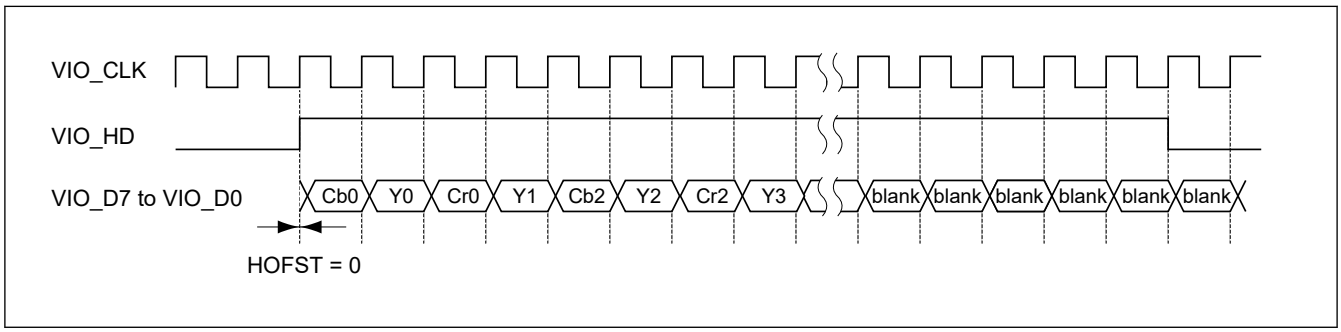
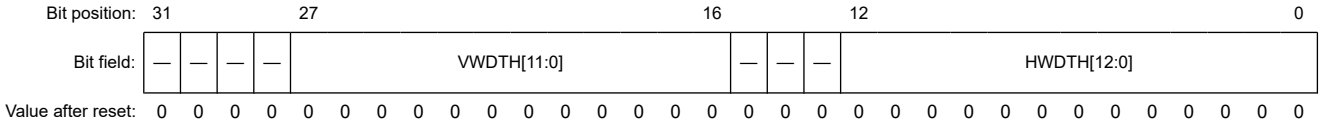


Figure 53.20 Timing when HD is data enable signal (8-bit interface)

53.2.6 CAPWR, CAPWR_x : Capture Interface Width Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0014 (CAPWR)
0x1014 (CAPWR_B)
0x2014 (CAPWR_M)



Bit	Symbol	Function	R/W
12:0	HWDTH[12:0]	Specify the horizontal capture period. Specify the number of cycles to be captured from the location specified by the HOFST bits. Figure 53.22 shows the timing when the horizontal blanking period is 0. The CEU captures for only the number of cycles specified by these bits in the horizontal direction. Make a similar setting for data synchronous fetch. The maximum value to be set is as follows: ● 8-bit interface Image capture (16-cycle units) : 5,120 cycles (2,560 pixels) Data synchronous fetch (8-cycle units) : 2,560 cycles (2,560 bytes) ● 16-bit interface Image capture (8-cycle units) : 2,560 cycles (2,560 pixels) Data synchronous fetch (4-cycle units) : 1,280 cycles (2,560 bytes) The lower 2 bits should be written 0. Note: In data synchronous fetch mode, set CFSZR and CDWDR according to the values set in this register. For details, see the descriptions on CFSZR and CDWDR.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
27:16	VWDTH[11:0]	Specify the vertical capture period (4-HD units). Specify the number of lines (HD count) to be captured from the location specified by the VOFST bits in CAMOR. Figure 53.21 shows the timing when the vertical blanking period is 0. The CEU captures only the number of lines (HD count) specified by these bits in the vertical direction. Make a setting in the same way to obtain data synchronization. The maximum value to be set is 1,920 HD (5 megapixels). The value of writing are ignored on the lower 2 bits, so write 0 to the lower 2 bits.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CAPWR sets the fetch (capture) cycle width when capturing images.

The cycle width unit differs according to the interface and the data type to be captured. For each setting unit, see Table 53.6.

This register is not used, during data enable fetch mode.

Table 53.6 Unit for setting fetch (capture) cycle width

	Vertical Direction		Horizontal Direction	
Interface	Image Capture	Data Synchronous Fetch	Image Capture	Data Synchronous Fetch
8-bit interface	4 HD	4 HD	16 cycles	8 cycles
16-bit interface	4 HD	4 HD	8 cycles	4 cycles

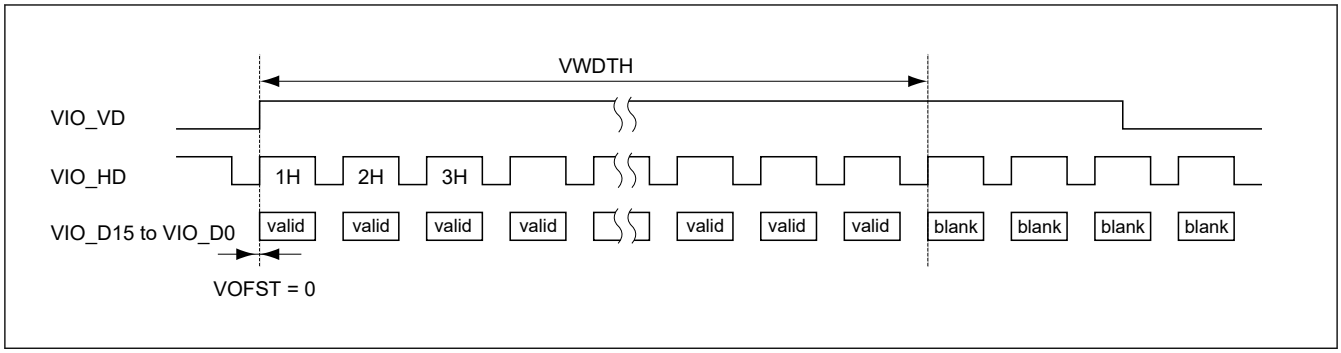


Figure 53.21 Vertical capture timing

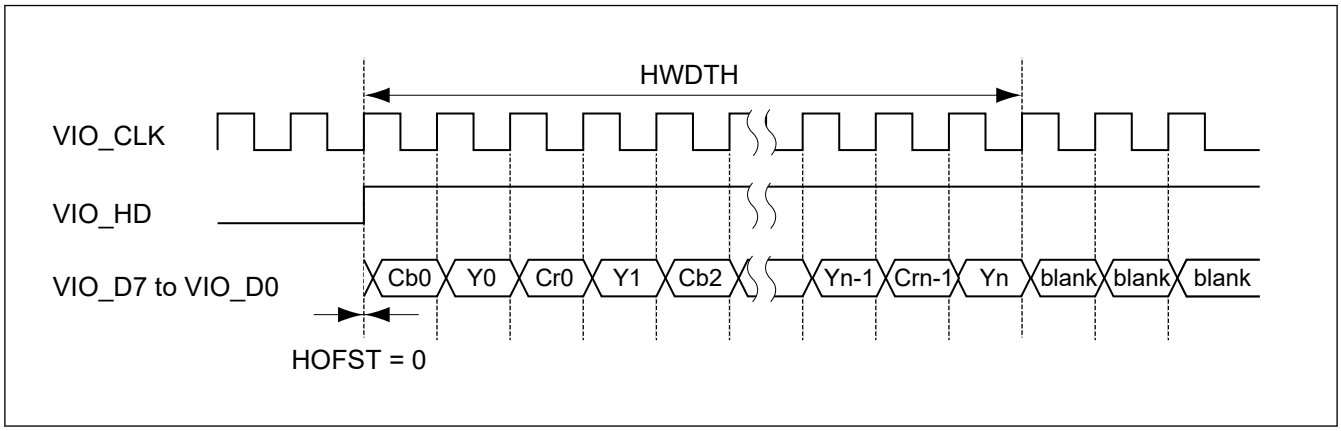


Figure 53.22 Horizontal capture timing (image capture with 8-bit interface)

53.2.7 CAIFR : Capture Interface Input Format Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	IFS	—	—	—	CIM	—	—	FCI[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	FCI[1:0]	Set the timing to start capturing. The timing to start capturing is set by specifying the image to be captured first. Set these bits to 00 when the input mode is progressive. 0 0: Capture starts from the VD input immediately after the CEU activation regardless of it being a top or bottom field 0 1: After the CEU activation, input of a top-field image is waited, and then capture starts from the top field 1 0: After the CEU activation, input of a bottom-field image is waited, and then capture starts from the bottom field 1 1: Setting prohibited	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
4	CIM	Sets the images to be captured. Clear this bit to 0 when the input mode for image capture is progressive (frame image) or when the input mode for image capture is interlace for continuous capture of both the top and bottom fields. Set this bit to 1 when the input mode for image capture is interlace for capture of only a one-field image. 0: Capture of frame image (1 VD) or both-field image (2 VD) 1: Capture of one-field image (1 VD)	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	IFS	Sets the input mode for capturing images. 0: Progressive 1: Interlace	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CAIFR sets the input mode (progressive or interlace) for capturing images, the images to be captured (frame, both-field, or one-field), the image from which capturing starts (top field or bottom field), etc. CAIFR is not used in data fetch mode.

Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to 1.

The items set by CAIFR are listed in Table 53.7.

Table 53.7 CAIFR Setting Items

Input Mode	IFS Bit	Captured Image	CIM Bit	Image to Start Capture	FCI Bits
Progressive	0	Frame	0	Frame immediately after activation	00
Interlace	1	Both-field (2-VD capture)	0	Field immediately after activation	00
				Top field	01
				Bottom field	10
				Setting prohibited	11
		One-field (1-VD capture)	1	Field immediately after activation	00
				Top field	01
				Bottom field	10
				Setting prohibited	11

In frame image capture and one-field image capture, a one-frame capture end interrupt occurs when capture for 1 VD finishes. In both-field image capture, a one-field capture end interrupt occurs when capture for 1 VD finishes and an oneframe capture end interrupt occurs when capture for 2 VD finishes. At this time, a one-field capture end interrupt occurs simultaneously with a one-frame capture end interrupt. Figure 53.23 shows the timing of a one-frame capture end interrupt and one-field capture end interrupt in both-field image capture.

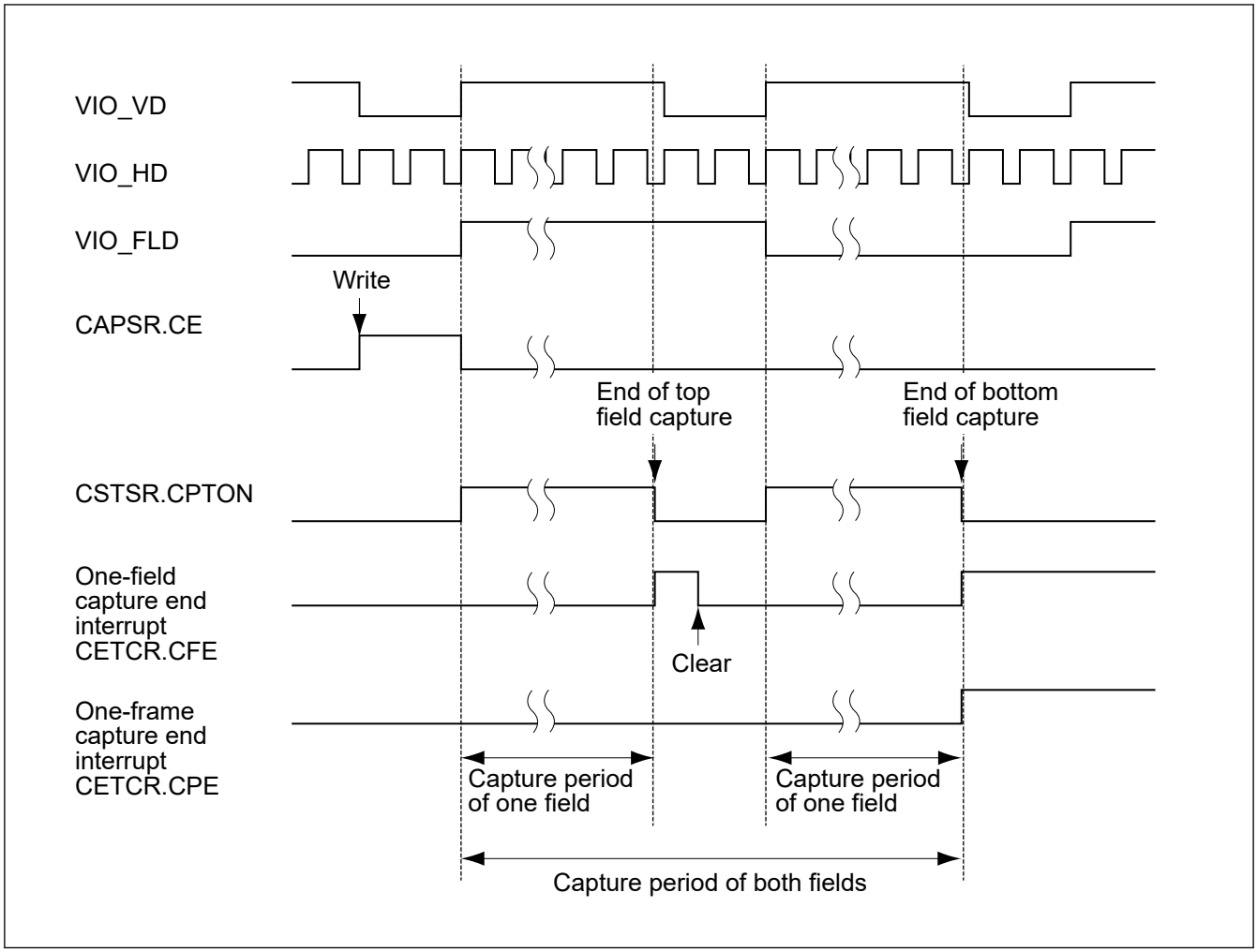


Figure 53.23 One-frame capture end interrupt and one-field capture end interrupt in both-field image capture

A captured frame image or captured one-field image is stored in the memory from the addresses set in CDAYR and CDACR (Figure 53.24). Captured both-field images are stored in different memory areas depending on whether it is a top-field or bottom-field image. A top-field image is stored in the memory from the addresses set in CDAYR and CDACR whereas a bottom-field image is stored in the memory from the addresses set in CDBYR and CDBCR (Figure 53.25).

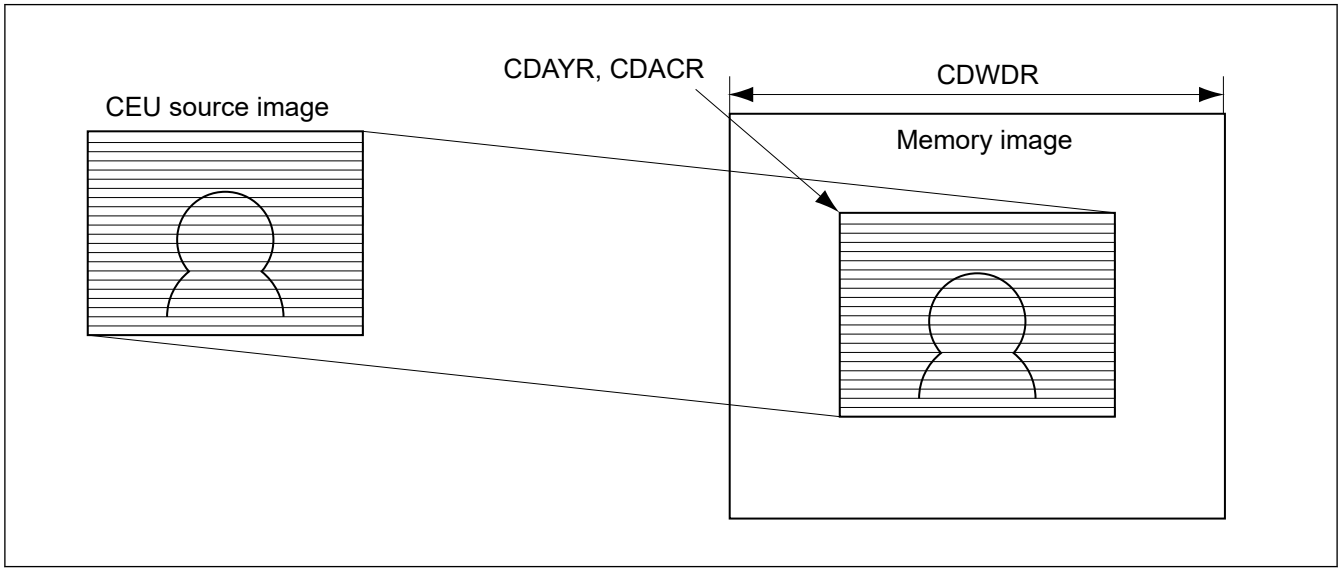


Figure 53.24 Image of storing captured frame image or captured one-field image in memory

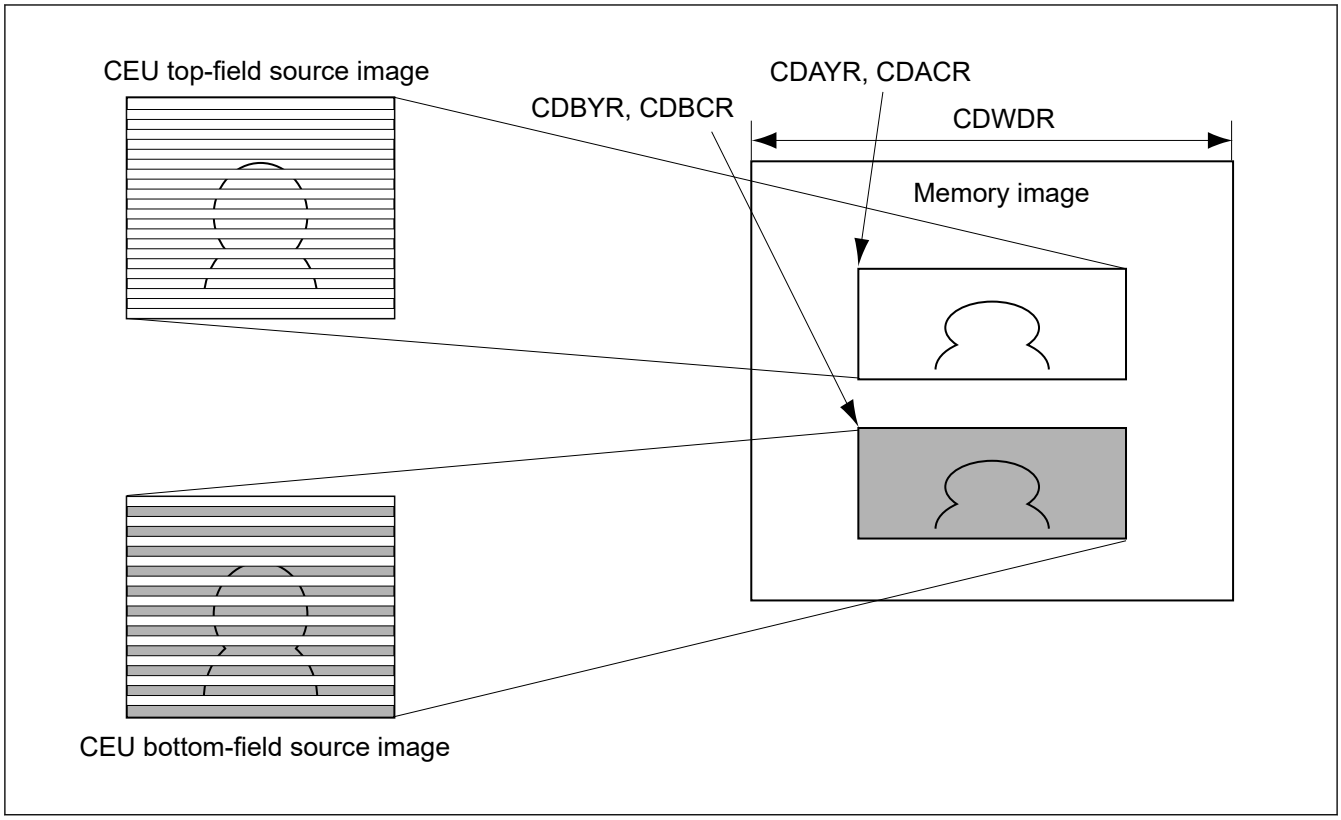


Figure 53.25 Image of storing captured both-field images in memory

If the FCI bits are set to 00b for continuous capture in interlace input mode, images are continuously captured for 2 VD with the first captured field as the reference in both-field image capture (Figure 53.26). In one-field image capture, only the first captured field is continuously captured for 1 VD (Figure 53.27).

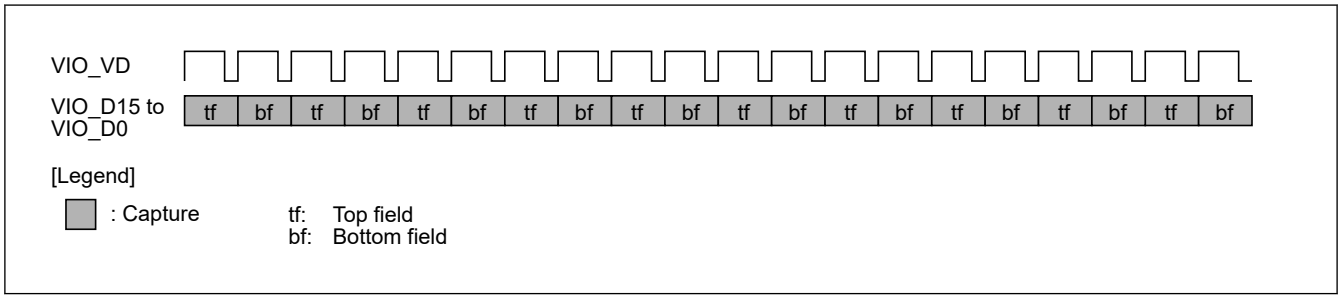


Figure 53.26 Continuous both-field capture in interlace mode (Image Immediately after Activation is Top Field (FCI[1:0] = 00b))

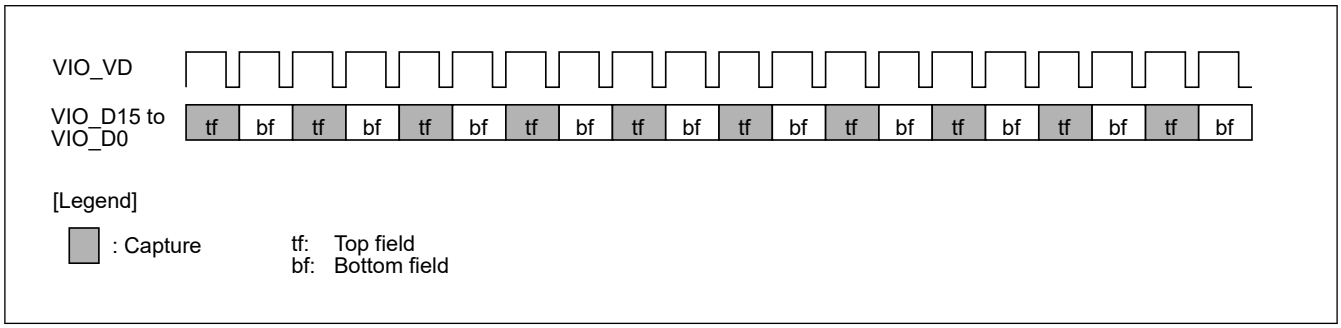


Figure 53.27 Continuous one-field capture in interlace mode (Image Immediately after Activation is Top Field (FCI[1:0] = 00b))

(1) Storage of Interlace Input as Frame Image

The CEU can store an interlace source image in the memory as a frame image. To store an interlace source image as a frame image, make the following register settings:

Input mode: Interlace (IFS = 1)

Capture image: Both-field (CIM = 0)

Image to start capture: Any setting other than the prohibited setting (FCI[1:0] = as desired)

Figure 53.28 shows a memory image of capturing both fields of an interlace input and storing it as a frame image in the memory. Set the start addresses of the memory destination for the captured top-field image in CDAYR and CDACR, and the start addresses of the memory destination for the captured bottom-field image in CDBYR and CDBCR. When storing an interlace image as a frame image in the memory, set the horizontal image size of the memory area in CDWDR with the top-field image and bottom-field image placed next to each other as shown in Figure 53.28. In addition, set the number of captured lines of the field image in the VWDTH bits in CAPWR.

A memory image of folding the horizontal image size of the memory area in Figure 53.28 at CDWDR/2 is shown in Figure 53.29. Setting the registers to form the image in Figure 53.28 enables an interlace image to be stored as a frame image in the memory as shown in Figure 53.29.

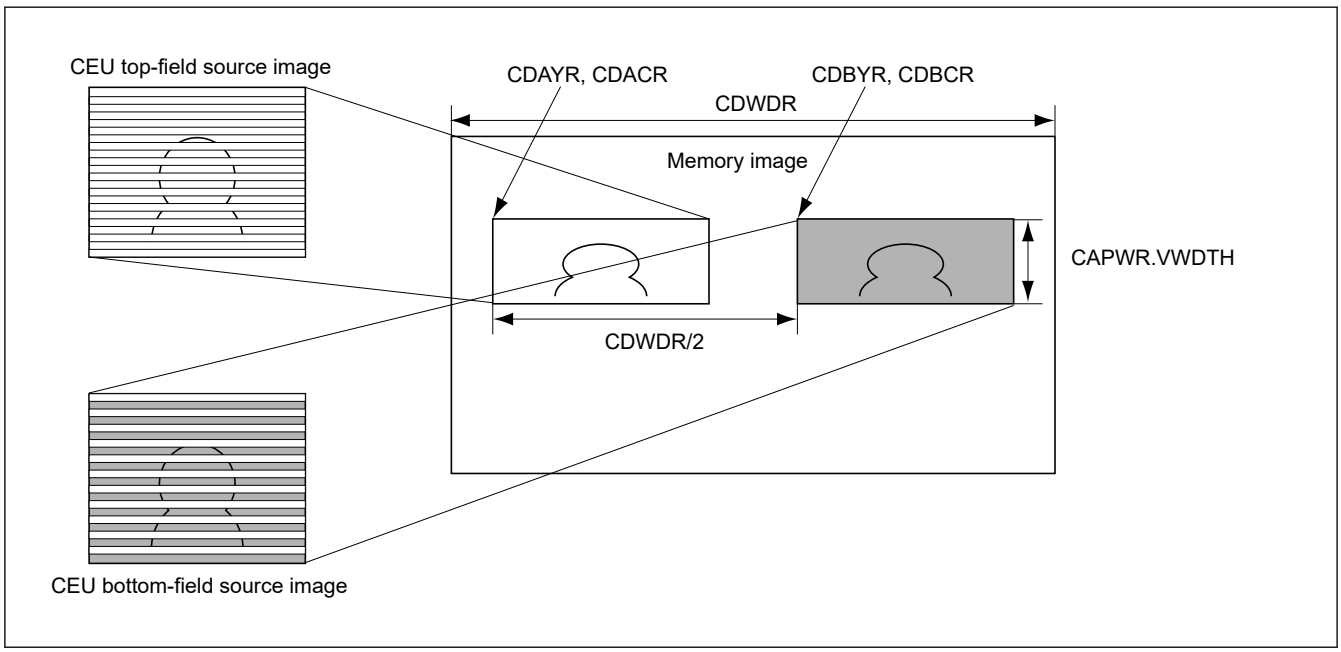


Figure 53.28 Image of storing captured both-fields of interlace input in memory

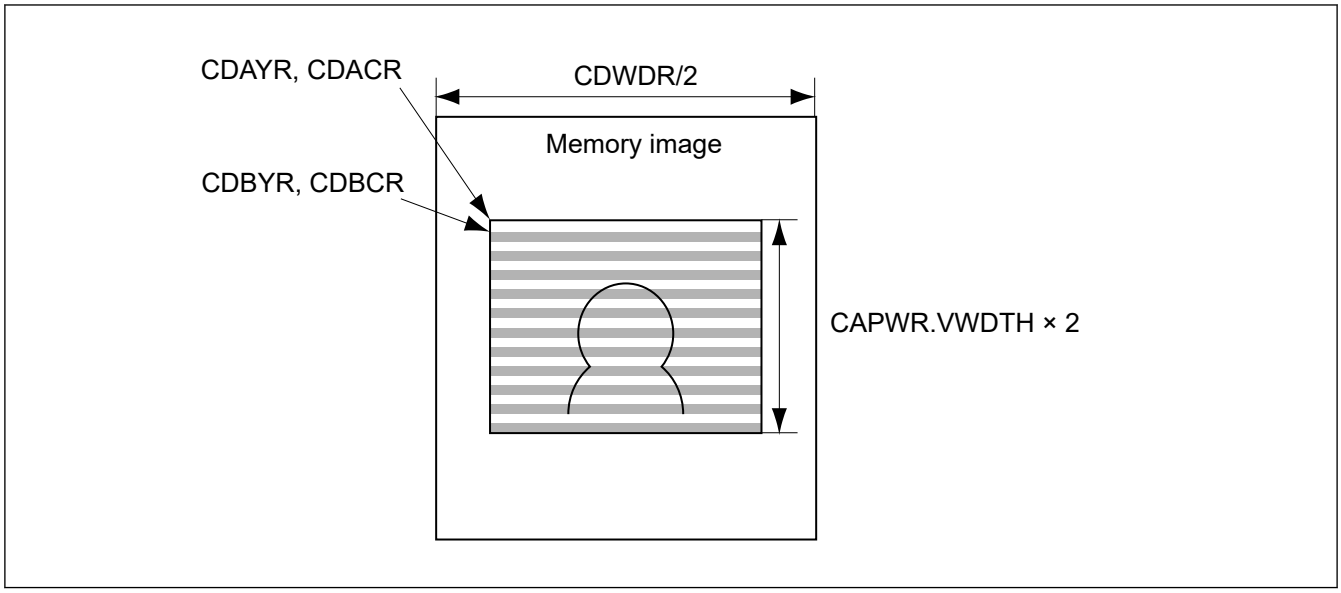


Figure 53.29 Image of storing Interlace input as frame image in memory

53.2.8 CRCNTR : CEU Register Control Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0028

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	RVS	—	—	RS	RC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RC	Specifies switching of the register plane used by the CEU in synchronization with VD. If the register plane is not switched, the register plane specified by the RS bit is used. 0: Uses the specified register plane in synchronization with VD 1: Switches the register plane in synchronization with VD	R/W
1	RS	Specifies which register plane is used by the CEU in synchronization with VD. The setting of this bit is valid only when RC = 0. 0: Uses plane A of the register 1: Uses plane B of the register	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	RVS	Sets the timing to switch the register plane in both-field capture. The setting of this bit is valid only when RC = 1 in both-field capture. 0: Switches the register plane every 2 VD 1: Switches the register plane every 1 VD	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CRCNTR controls switching of the planes of registers with a 2-plane configuration.

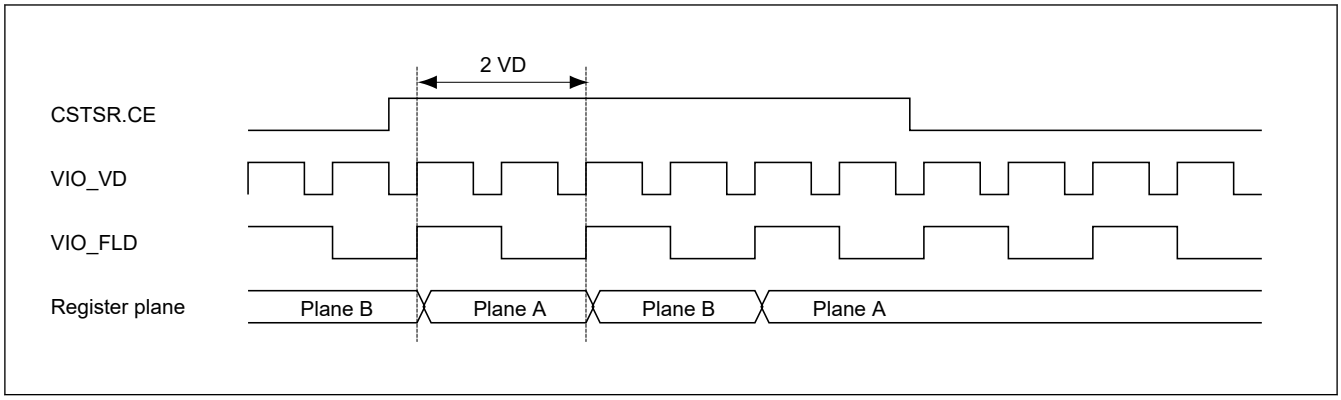


Figure 53.30 Timing for register plane switching when RVS = 0

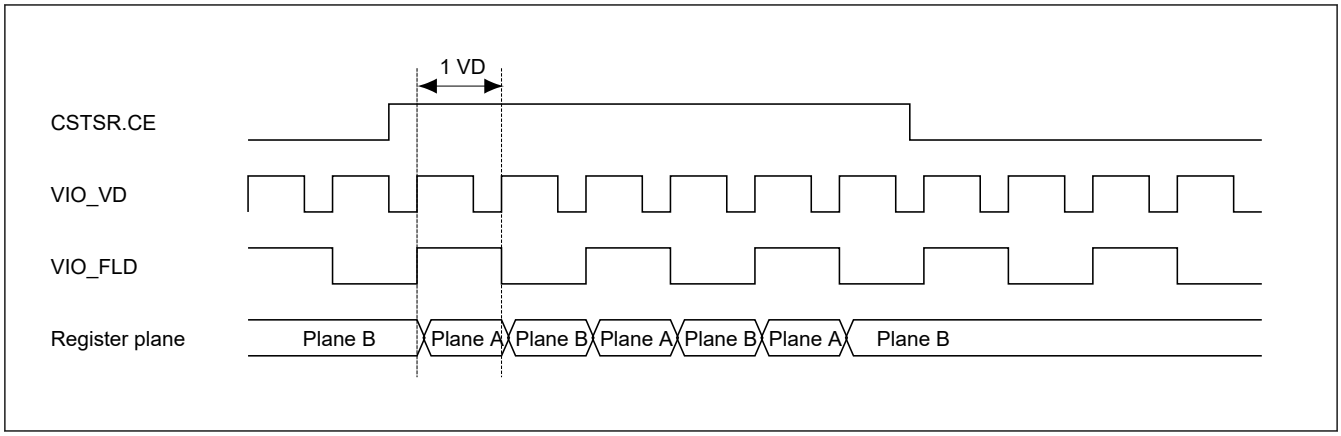


Figure 53.31 Timing for register plane switching when RVS = 1

53.2.9 CRCMPR : CEU Register Forcible Control Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x002C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RA	Indicates the register plane currently specified. This register value automatically changes in synchronization with VD for starting capture. To start capture with plane A of the register when a setting to switch the register plane in synchronization with VD has been made (CRCNTR.RC = 1), specify plane B of the register using this bit. 0: Specifies plane A of the register 1: Specifies plane B of the register	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CRCMPR forcibly controls switching of the planes of registers with a 2-plane configuration. Setting this register enables direct control of register plane switching.

Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to 1.

53.2.10 CFLCR, CFLCR_x : Capture Filter Control Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0030 (CFLCR)
0x1030 (CFLCR_B)
0x2030 (CFLCR_M)

Bit position:	31 28	27 16	15 12	11 0
Bit field:	VMANT[3:0]	VFRAC[11:0]	HMANT[3:0]	HFRAC[11:0]
Value after reset:	0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
11:0	HFRAC[11:0]	Fraction Part of Horizontal Scale-Down Factor The specifiable range is 0x000 to 0xFF8. The fraction of the scale-down factor that cannot be set with only the HMANT bits must be specified with these bits. The value of writing are ignored on the lower 3 bits, so write 0 to the lower 3 bits.	R/W
15:12	HMANT[3:0]	Mantissa Part of Horizontal Scale-Down Factor The specifiable range is 0x0 to 0xF. When 0x0 is set for the HMANT bits and 0x000 is set for the HFRAC bits, the scale-down filter is not used.	R/W
27:16	VFRAC[11:0]	Fraction Part of Vertical Scale-Down Factor The specifiable range is 0x000 to 0xFF8. The fraction of the scale-down factor that cannot be set with only the VMANT bits must be set with these bits. The value of writing are ignored on the lower 3 bits, so write 0 to the lower 3 bits.	R/W
31:28	VMANT[3:0]	Mantissa Part of Vertical Scale-Down Factor The specifiable range is 0x0 to 0xF. When 0x0 is set for the VMANT bits and 0x000 is set for the VFRAC bits, the scale-down filter is not used.	R/W

Note: S-TYPE-3, P-TYPE-3

CFLCR sets the scale-down factor for the filter to scale images down.

The CEU has an image scale-down filter which can be used to scale down the captured images before storing them in the memory. Set CFLCR to 0 when not performing scale-down (same size output). If a value other than 0 is set in CFLCR, scale-down is performed. In data fetch mode, set CFLCR to 0.

When handling an interlace source image as a frame image, set CFLCR to 0 not to use the filter.

An image scale-down filter is installed in the CEU, and the captured images can be scaled down and stored in the memory.

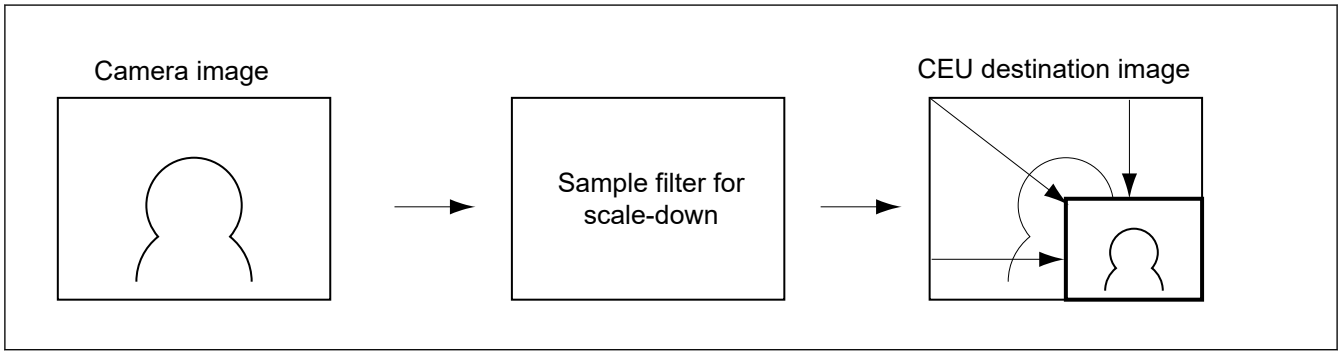


Figure 53.32 Scale-down of captured image

The formulas for obtaining the MANT (VMANT or HMANT) and FRAC (VFRAC or HFRAC) values from the input pixel count and output pixel count of the filter are shown below. Set the MANT and FRAC bits in order to obtain the desired output pixel count from the number of pixels input to the CEU.

First, calculate preliminary MANT and FRAC values. The parameters needed for calculation are defined as follows:

$$\alpha = \mathrm{MANT} \times 4096 + \mathrm{FRAC} \quad \text{...Formula 1}$$

$$\mathrm{SCL\ (scaling\ factor)} = \frac{4096}{\alpha} \quad \text{...Formula 2}$$

Assuming an operator ⌊x⌋ which discards fractions of an integer x, the MANT and FRAC values can be temporarily set as follows, according to formula 1 and formula 2.

$$\mathrm{MANT} = \left\lfloor \frac{1}{\mathrm{SCL}} \right\rfloor, \mathrm{FRAC} = \left\lfloor 512 \times \left(\frac{1}{\mathrm{SCL}} - \mathrm{MANT} \right) \right\rfloor \times 8$$

Here, the scaled-down filter output size ($SIZE_D$) can be calculated using the input image size S_{in} (8-bit interface: half of the CAPWR setting, 16-bit interface: CAPWR setting) in the following formula.

$$\mathrm{SIZE}_D = \left\lfloor 1 + \left(\left\lfloor \frac{1}{2} + \frac{S_{\mathrm{in}} - 1}{\mathrm{MANT}_{\mathrm{pre}}} \right\rfloor - 1 \right) \times \frac{\mathrm{MANT}_{\mathrm{pre}} \times 4096}{\alpha} \right\rfloor \quad \text{...Formula 3}$$

$$\begin{bmatrix} \mathrm{MANT}_{\mathrm{pre}} = 1 \rightarrow (0 \leq \mathrm{MANT} < 2) \\ \mathrm{MANT}_{\mathrm{pre}} = 2 \rightarrow (2 \leq \mathrm{MANT} < 4) \\ \mathrm{MANT}_{\mathrm{pre}} = 4 \rightarrow (4 \leq \mathrm{MANT} < 8) \\ \mathrm{MANT}_{\mathrm{pre}} = 8 \rightarrow (8 \leq \mathrm{MANT}) \end{bmatrix}$$

The number of output pixels can be obtained by substituting the temporarily calculated MANT, FRAC, and input image size into these formulas. If the calculated number of output pixels is smaller than the number of output pixels used to obtain the preliminary MANT and FRAC values, recalculate with a smaller FRAC (α) value, and set the MANT and FRAC values in this register so that a pixel value greater than the desired number of output pixels can be obtained.

Example: Scale down 640 pixels to 480 pixels

SCL = 480/640 = 3/4, and the preliminary settings of MANT = 1, MANTpre = 1, and FRAC = 0x550 are made. Substituting these in the following formula results in an output pixel count of 479.

$$\mathrm{SIZE}_D = \left\lfloor 1 + \left(\left\lfloor \frac{1}{2} + \frac{S_{\mathrm{in}} - 1}{\mathrm{MANT}_{\mathrm{pre}}} \right\rfloor - 1 \right) \times \frac{\mathrm{MANT}_{\mathrm{pre}} \times 4096}{\alpha} \right\rfloor \quad \text{...Formula 3}$$

Since this output pixel count is smaller than the desired output pixel count of 480, the formula is recalculated with a FRAC value of 0x548, a value eight less than the previous time. The obtained result of output pixel count = 480 is equal to the desired output pixel count of 480, so this register is set as MANT = 1 and FRAC = 0x548.

Table 53.8 Setting examples for each scale-down filter factor

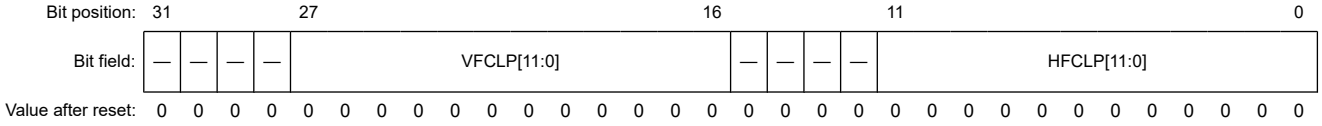
Scale-down factor	FRAC		MANT	Input pixel count	Output pixel count	Clipping size (CFSZR)
	Decimal	Hexadecimal				
7/8	576	0x240	1	640	560	560
3/4	1352	0x548	1	640	480	480
5/8	2448	0x990	1	640	400	400
1/2	0	0x0	2	640	320	320
3/8	2728	0xAA8	2	640	240	240
1/3	0.0	0x0	3	640	213	212
1/4	0.0	0x0	4	640	160	160
1/5	0.0	0x0	5	640	128	128
1/6	0.0	0x0	6	640	107	104
1/7	0.0	0x0	7	640	91	88
1/8	0.0	0x0	8	640	80	80
1/16	4088	0xFF8	15	640	40	40

Note: This scale-down filter uses a VGA-size line memory for scale-down. Therefore, when an image larger than the VGA size is input for scale-down, settings must be made so that the output image size is equal to or larger than the SubQCIF size and equal to or smaller than the VGA size. When an image is not scaled down (same size output), this restriction does not apply.

53.2.11 CFSZR, CFSZR_x : Capture Filter Size Clip Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0034 (CFSZR)
0x1034 (CFSZR_B)
0x2034 (CFSZR_M)



Bit	Symbol	Function	R/W
11:0	HFCLP[11:0]	Specify the horizontal clipping value of the filter output size (8-pixel units). The lower 3 bits should be written 0.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
27:16	VFCLP[11:0]	Specify the vertical clipping value of the filter output size (4-pixel units). The value of writing are ignored on the lower 2 bits, so write 0 to the lower 2 bits.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CFSZR sets the clipping size for fine adjustment of the image size output from the filter, and must be set in combination with CFLCR. When clipping the output size of the filter, set the clipping size as a number of pixels, and the setting unit should be four pixels. CFSZR must be set even when scale-down is not performed (same size output).

This register is not used, during data enable fetch mode.

In data synchronous fetch mode, set CFSZR according to the setting of CAPWR.

The scale-down filter in the CEU may output an odd number of pixels or lines depending on the settings. To adjust the output size of the filter, the CEU clips the destination image by using the number of pixels specified in CFSZR, as shown in Figure 53.33. The clipping size must be specified vertically in 4-pixel units and horizontally in 8-pixel units.

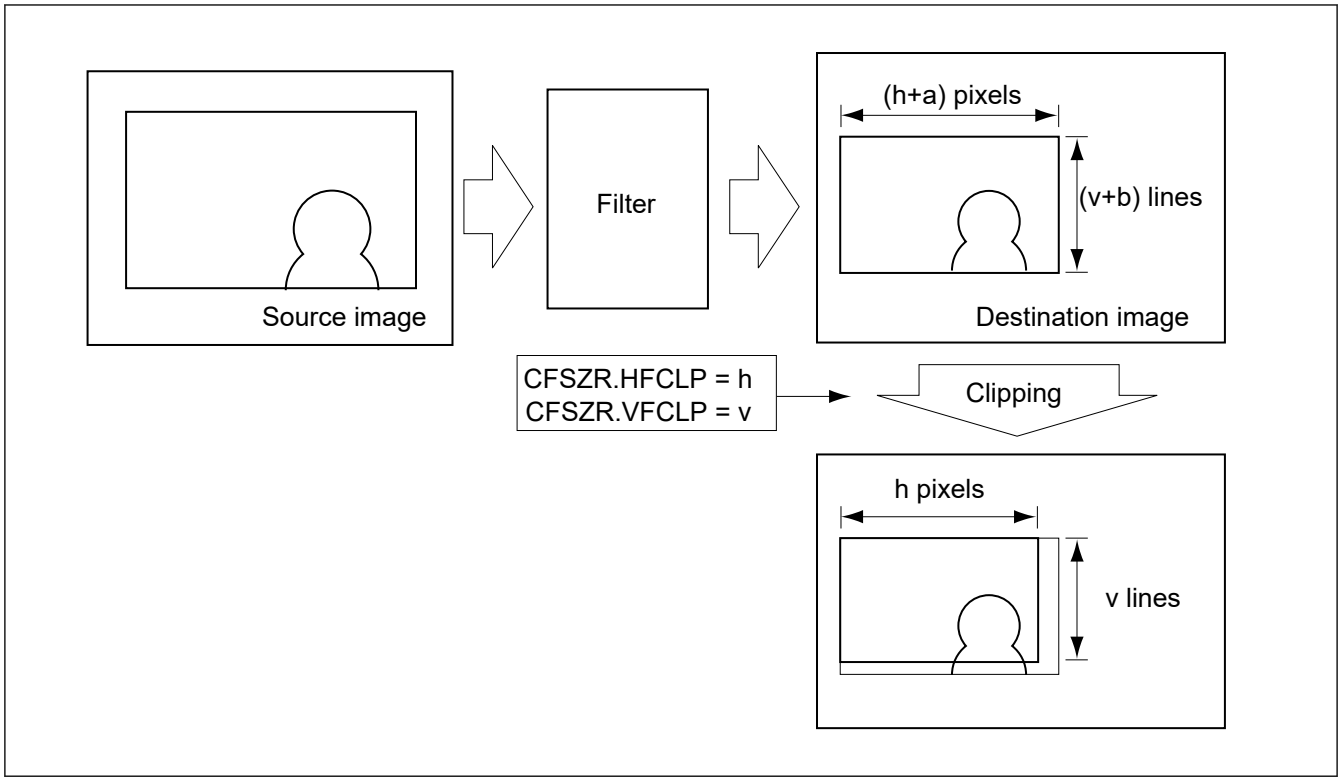


Figure 53.33 Clipping of image output from filter

The pixels to be clipped are counted from the top-left corner of a display. The pixels located to the right of the specified number of pixels or below the specified number of lines are discarded by the clipping function. If the number of pixels specified in CFSZR is larger than that output from the filter, correct operation cannot be guaranteed. To avoid this, the clipping size specified in CFSZR must be equal to or smaller than the number of pixels output from the filter.

Note: In data synchronous fetch mode, the following settings are required. Data cannot be fetched correctly unless the following settings are made.

- 8-bit interface:
  VFCLP = CAPWR.VWDTH
  HFCLP = CAPWR.HWDTH/2
- 16-bit interface:
  VFCLP = CAPWR.VWDTH
  HFCLP = CAPWR.HWDTH

53.2.12 CDWDR, CDWDR_x : Capture Destination Width Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0038 (CDWDR)
0x1038 (CDWDR_B)
0x2038 (CDWDR_M)

Bit position:	31												12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CHDW[12:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
12:0	CHDW[12:0]	Specify the horizontal image size in the memory area where the captured image is to be stored (8-byte units). The image data captured by the CEU is stored in the memory. In data synchronous fetch mode, set as follows: 8-bit interface: CHDW = CAPWR.HWDTH 16-bit interface: CHDW = CAPWR.HWDTH × 2 The lower 3 bits should be written 0.	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDWDR sets the horizontal image size in the memory area where the captured image is to be output in 8-byte units (8-pixel units).

In data synchronous fetch mode, set CDWDR according to the setting of CAPWR.

This register is not used, during data enable fetch mode.

53.2.13 CDAYR, CDAYR_x : Capture Data Address Y Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x003C (CDAYR)
0x103C (CDAYR_B)
0x203C (CDAYR_M)

Bit position:	31																												0
Bit field:	CAYR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CAYR[31:0]	Capture Data Address Y ● Frame image capture: These bits set the address for storing the luminance component data of the captured data (8-pixel units). ● One-field image capture: These bits set the address for storing the luminance component data of the captured data (8-pixel units). ● Both-field image capture: These bits set the address for storing the luminance component data of the captured top-field data (8-pixel units). ● Data fetch: These bits set the address for storing data (8-byte units). ● Data enable fetch bundle write: These bits set the address for storing data (32-byte units). The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDAYR specifies the address where the luminance component of the captured data is to be stored in frame image capture or one-field image capture, the address where the luminance component of the captured top field is to be stored in both-field image capture, and the address where the fetched data is to be stored in data fetch. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. In frame image capture or one-field image capture, set the start address of the memory area where the luminance component of the captured data is to be stored by CDAYR. In both-field image capture, set the start address of the memory area where the luminance component of the captured top-field image is to be stored by CDAYR. In data fetch, set the start address of the memory area where data is to be stored by CDAYR.

Because the address must be specified in 32 bits, the address set by CDAYR must be in longword units.

Set the address of the starting point of the memory area where the fetched data is to be stored in this register, as shown in Figure 53.34.

- Frame image capture: Set the address of the starting point of the memory area where the luminance component of the captured image is to be stored.
- One-field image capture: Set the address of the starting point of the memory area where the luminance component of the captured image is to be stored.

- Both-field image capture: Set the address of the starting point of the memory area where the luminance component of the captured top-field image is to be stored.
- Data fetch: Set the address of the starting point of the memory area where the fetched data is to be stored. In data fetch mode, the data is simply stuffed in order from the start address so the end address becomes as follows:
  End address = CDAYR + number of fetched bytes
- Data enable fetch bundle write: Set the address in 32-byte units.

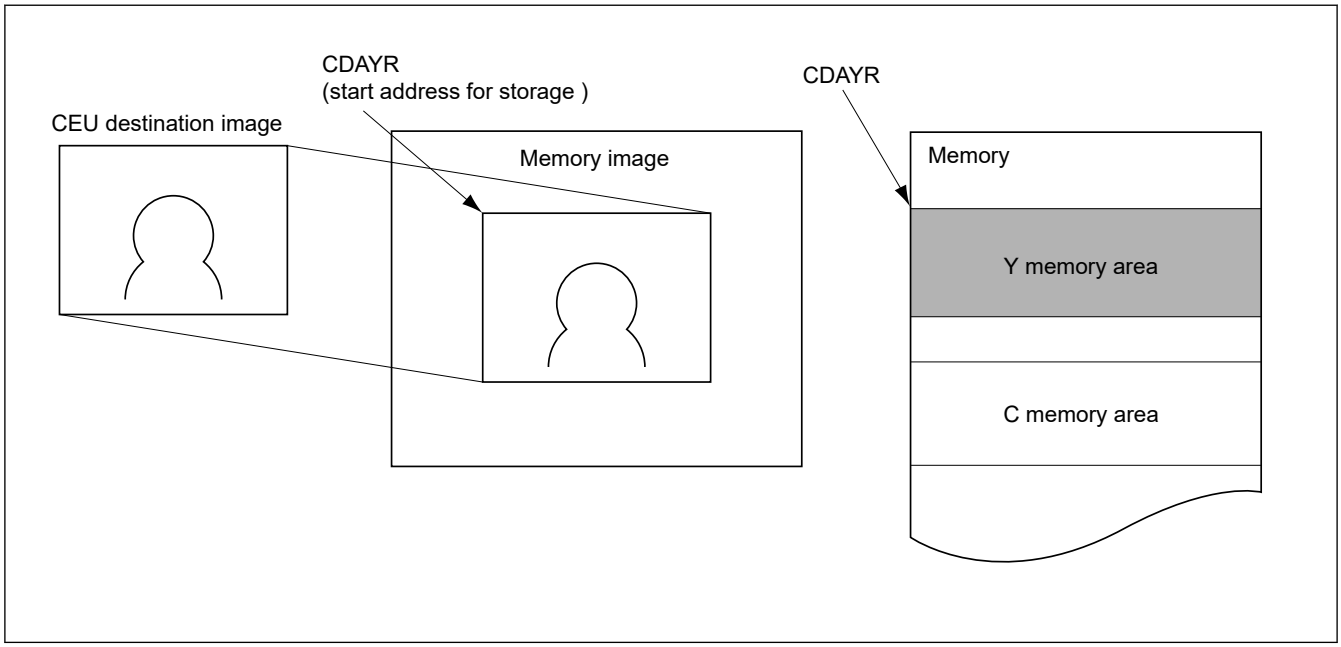


Figure 53.34 Relationship between captured image and luminance component memory area

53.2.14 CDACR, CDACR_x : Capture Data Address C Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0040 (CDACR)
0x1040 (CDACR_B)
0x2040 (CDACR_M)

Bit position:	31																													0
Bit field:	CACR[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CACR[31:0]	Capture Data Address C ● Frame image capture: These bits set the address for storing the chrominance component data of the captured data (8-pixel units). ● One-field image capture: These bits set the address for storing the chrominance component data of the captured data (8-pixel units). ● Both-field image capture: These bits set the address for storing the chrominance component data of the captured top-field data (8-pixel units). The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDACR specifies the address where the chrominance component of the captured data is to be stored in frame image capture or one-field image capture, and the address where the chrominance component of the captured top field is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. In frame image capture or onefield image capture, set the start address of the memory area where the chrominance component of the captured data is to be stored by

CDACR. In both-field image capture, set the start address of the memory area where the chrominance component of the captured top-field image is to be stored by CDACR. CDACR is not used in data fetch.

Because the address must be specified in 32 bits, the address set by CDACR must be in longword units.

Set the address of the starting point of the memory area where the chrominance component of the captured image is to be stored in this register, as shown in Figure 53.35. The chrominance component has an output data format like that in Figure 53.36, and is saved in the memory in this format.

- Frame image capture: Set the address of the starting point of the memory area where the chrominance component of the captured image is to be stored.
- One-field image capture: Set the address of the starting point of the memory area where the chrominance component of the captured image is to be stored.
- Both-field image capture: Set the address of the starting point of the memory area where the chrominance component of the captured top-field image is to be stored.

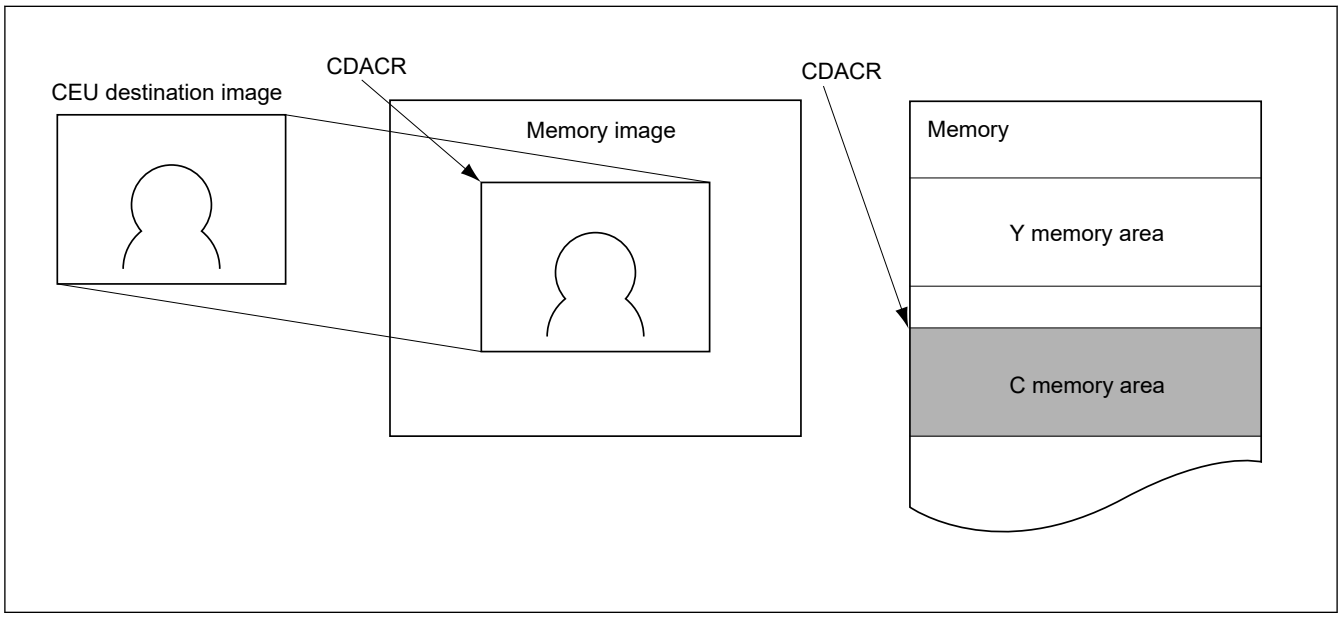


Figure 53.35 Relationship between captured image and chrominance component memory area

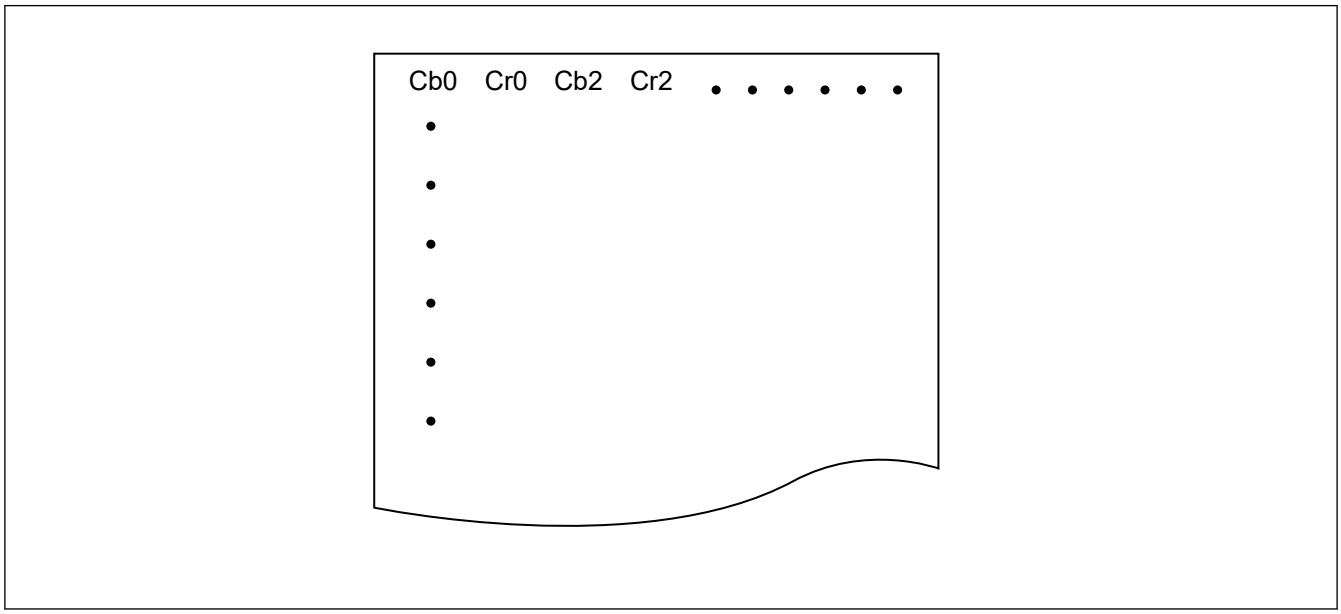


Figure 53.36 Image of storing chrominance components in memory

53.2.15 CDBYR, CDBYR_x : Capture Data Bottom-Field Address Y Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0044 (CDBYR)
0x1044 (CDBYR_B)
0x2044 (CDBYR_M)

Bit position:	31																												0
Bit field:	CBYR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CBYR[31:0]	Set the address for storing the luminance component data of the captured bottom-field data (8-pixel units). The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDBYR specifies the address where the luminance component of the captured bottom-field data is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. Set the start address of the memory area where the luminance component of the captured bottom-field image is to be stored by CDBYR. CDBYR is not used in frame image capture, one-field image capture, or data fetch.

Because the address must be specified in 32 bits, the address set by CDBYR must be in longword units.

Set the address of the starting point of the memory area where the luminance component of the captured bottom-field image is to be stored in this register, as shown in Figure 53.37.

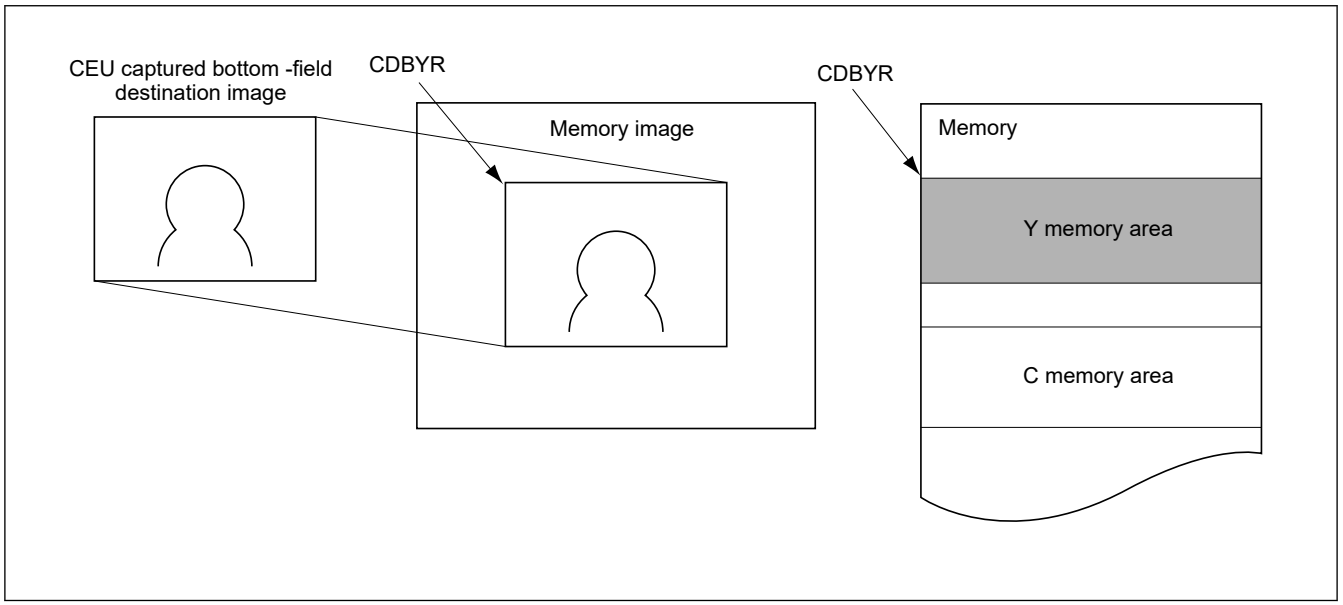


Figure 53.37 Relationship between captured bottom-field image and luminance component memory area

53.2.16 CDBCR, CDBCR_x : Capture Data Bottom-Field Address C Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0048 (CDBCR)
0x1048 (CDBCR_B)
0x2048 (CDBCR_M)

Bit position:	31																												0
Bit field:	CBCR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CBCR[31:0]	Set the address for storing the chrominance component data of the captured bottom-field data (8-pixel units). The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDBCR specifies the address where the chrominance component of the captured bottom-field data is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. Set the start address of the memory area where the chrominance component of the captured bottom-field image is to be stored by CDBCR. CDBCR is not used in frame image capture, one-field image capture, or data fetch.

Because the address must be specified in 32 bits, the address set by CDBCR must be in longword units.

Set the address of the starting point of the memory area where the chrominance component of the captured bottom-field image is to be stored in this register, as shown in Figure 53.38. The chrominance component has an output data format like that in Figure 53.39, and is saved in the memory in this format.

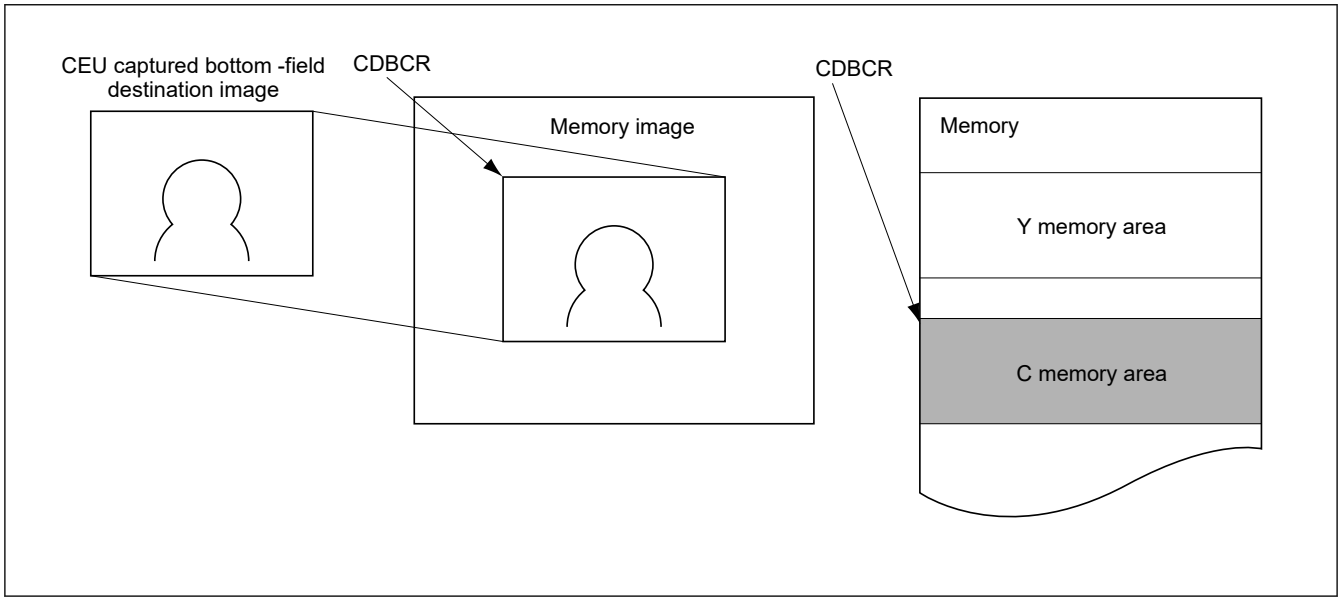


Figure 53.38 Relationship between captured bottom-field image and chrominance component memory area

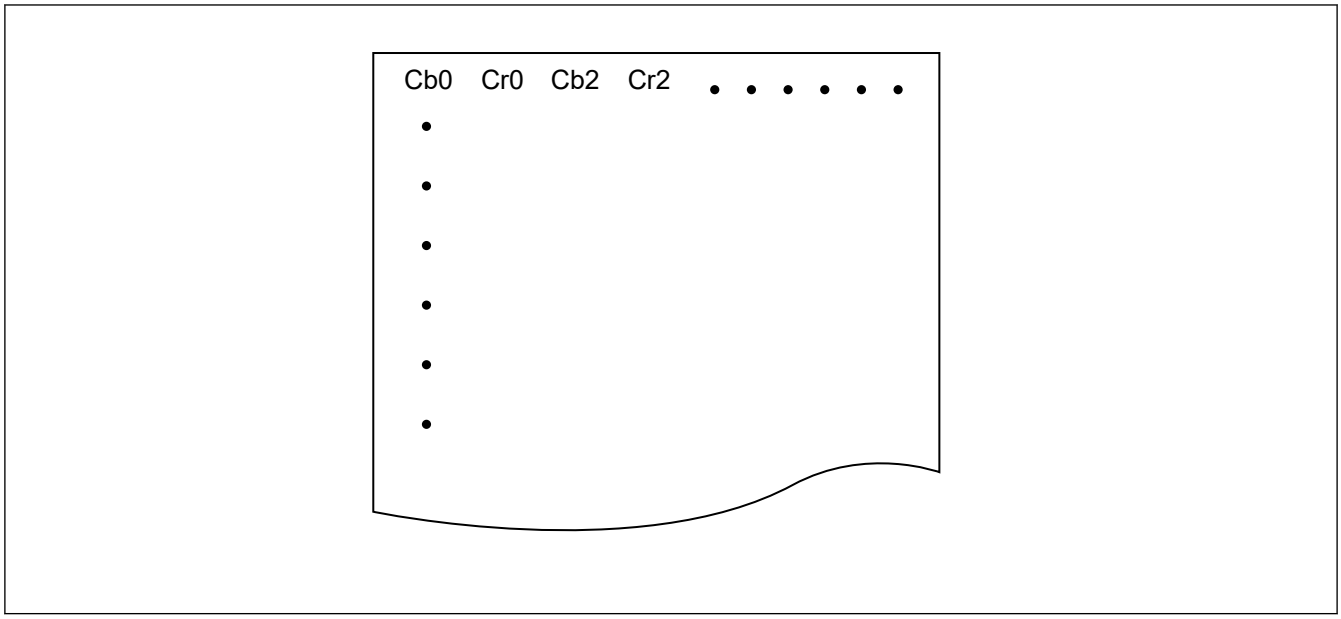


Figure 53.39 Image of storing chrominance components in memory

53.2.17 CBDSR, CBDSR_x : Capture Bundle Destination Size Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x004C (CBDSR)
0x104C (CBDSR_B)
0x204C (CBDSR_M)

Bit position:	31									22																				0
Bit field:	—	—	—	—	—	—	—	—	—	CBVS[22:0]																				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
22:0	CBVS[22:0]	Select the number of lines or number of bytes for output to the memory in a bundle write. Image capture and data synchronous fetch: Number of lines for output to the memory in a bundle write. Unit: 8 lines, min.: 8 lines, max.: 1,920 lines (0x780) Data enable fetch: Number of bytes for output to the memory in a bundle write. Unit: 32 bytes, min.: 512 bytes, max.: 6291456 bytes (0x600000) The value of writing are ignored on the lower 3 bits, so write 0 to the lower 3 bits.	R/W
31:23	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CBDSR sets the size of output to memory in a bundle write. The number of output lines should be specified for image capture or data synchronous fetch. The number of bytes should be specified for data enable fetch.

(a) Image capture and data synchronous fetch

Set the number of lines of captured data to be written to the memory by a bundle write as a multiple of eight. This register is valid only when CDOCR.CBE = 1. When CDOCR.CBE = 1 and this register cleared to 0, this module operates with the number of lines of captured data to be written to the memory as eight. The maximum number of lines that can be set is 1,920 (0x780). Only bits CBVS[11:3] are valid.

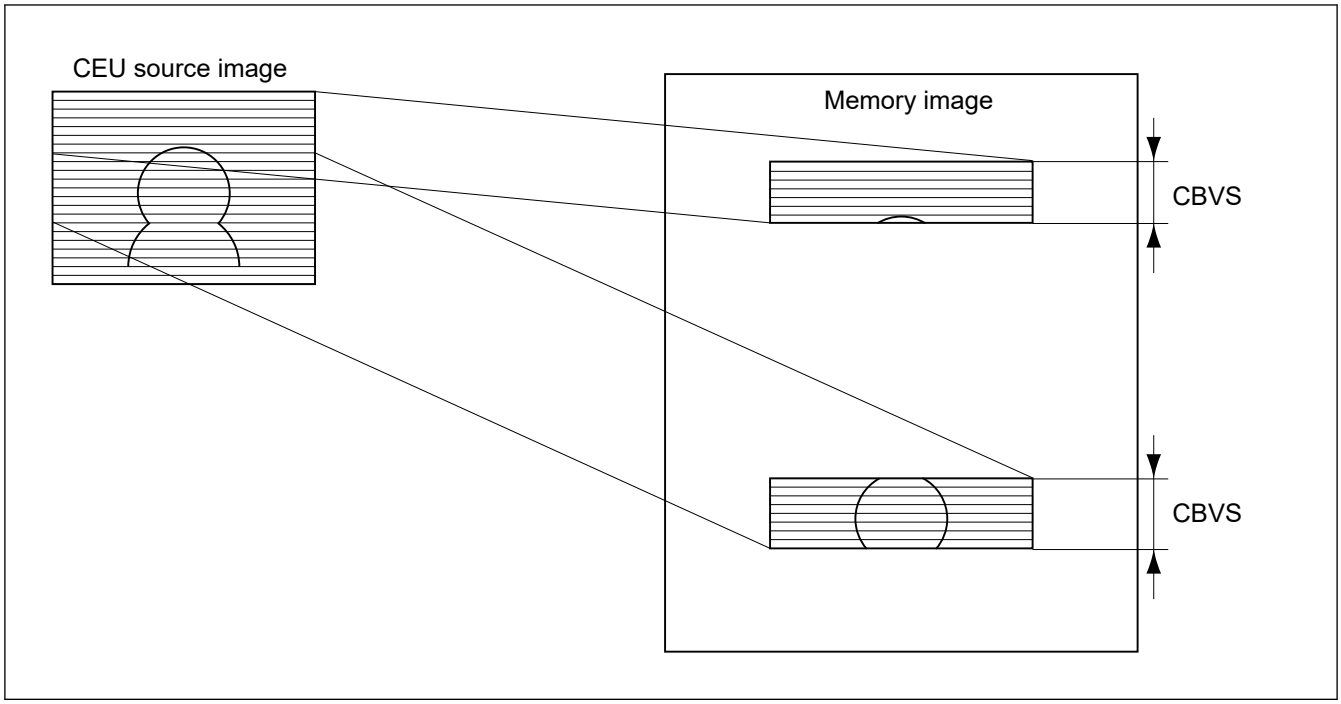


Figure 53.40 Image of storing captured image in memory by bundle write

(b) Data enable fetch

Set the number of bytes of captured data to be written to the memory by a bundle write as a multiple of 32. This register is valid only when CDOCR.CBE = 1. The minimum settable size is 512 bytes. When a number smaller than 512 bytes is specified, operation is not guaranteed.

53.2.18 CFWCR : Firewall Operation Control Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x005C

Bit position:	31																															5							0
Bit field:	FWV[26:0]																																—	—	—	—	FWE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FWE	Firewall Operation With the setting of FWE = 1, when a write address exceeds the value set with FWV, the address is retained and an interrupt source FWF is set. After this, the address is not incremented and data is overwritten on the upper limit address. 0: Firewall is not activated. 1: Firewall is activated.	R/W
4:1	—	These bits are read as 0. The write value should be 0.	R/W
31:5	FWV[26:0]	Specify the upper limit of a write address. Specify the upper 27 bits of the 32-bit address. The upper limit of an address is FWV[26:0]<<5 + 0x1F.	R/W

Note: S-TYPE-3, P-TYPE-3

CFWCR specifies the upper limit of a write address in data enable fetch. When the VD input from an external module dose not go low and end notification is not given, this register can prevent writing to memory from being out of control.

This register is enabled only in data enable fetch.

53.2.19 CLFCR, CLFCR_x : Capture Low-Pass Filter Control Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0060 (CLFCR)
0x1060 (CLFCR_B)
0x2060 (CLFCR_M)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LPF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LPF	Enables or disables operation of the low-pass filter. The low-pass filter removes high-frequency components from the destination image in the horizontal direction. Clear this bit to 0 in data fetch mode. 0: Low-pass filter not used 1: Low-pass filter used (only in the horizontal direction)	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CLFCR specifies whether or not to operate the low-pass filter. In data fetch mode, clear the LPF bit to 0.

The characteristic of the low-pass filter installed in the CEU causes the phase location of the image processed by the low-pass filter to be shifted right by one pixel compared to the raw image.

53.2.20 CDOCR, CDOCR_x : Capture Data Output Control Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0064 (CDOCR)
0x1064 (CDOCR_B)
0x2064 (CDOCR_M)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CBE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	CDS	—	COLS	COWS	COBS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	COBS	Controls swapping in 8-bit units for data output from the CEU. 0: Data is not swapped in 8-bit units 1: Data is swapped in 8-bit units	R/W
1	COWS	Controls swapping in 16-bit units for data output from the CEU. 0: Data is not swapped in 16-bit units 1: Data is swapped in 16-bit units	R/W
2	COLS	Controls swapping in 32-bit units for data output from the CEU. 0: Data is not swapped in 32-bit units 1: Data is swapped in 32-bit units	R/W

Bit	Symbol	Function	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	CDS	Sets the image format when outputting the image data captured in the YCbCr422 format to the memory. When 0 is written to this bit, only the luminance component (Y) is output and no chrominance components (Cb and Cr) are output for the odd-numbered lines. With an interlace source image, similarly only the luminance component (Y) is output and no chrominance components (Cb and Cr) are output for the odd-numbered lines of the field. In data fetch mode, set this bit to 1. 0: Converts the YCbCr422 format to the YCbCr420 format before outputting data to the memory 1: Outputs data in the YCbCr422 format to the memory without conversion	R/W
15:5	—	These bits are read as 0. The write value should be 0.	R/W
16	CBE	Controls the number of lines of captured data to be written to the memory. ● Image capture This bit controls the number of lines of captured data to be written to the memory. When bundle write is set by this register, captured data is written in line units specified by CBDSR to the addresses specified by CDAYR and CDACR, and CDAYR2 and CDACR2 (CDBYR and CDBCR, and CDBYR2 and CDBCR2 for the bottom field in both-field capture) alternately (Figure 53.41). When captured data has been written for the number of lines set by CBDSR, a write end interrupt corresponding to each address setting register occurs. However, after write for one-frame (one-field) capture ends, a bundle write end interrupt does not occur even when bundle write has finished. ● Data synchronous fetch This bit controls the number of lines of captured data to be written to the memory. When bundle write is set by this register, captured data is written in line units specified by CBDSR to the addresses specified by CDAYR and CDAYR2 alternately. When captured data has been written for the number of lines set by CBDSR, a write end interrupt corresponding to each address setting register occurs. However, after write for one-frame capture ends, a bundle write end interrupt does not occur even when bundle write has finished. ● Data enable fetch This bit controls the number of bytes of captured data to be written to the memory. When bundle write is set by this register, captured data is written in byte units specified by CBDSR to the addresses specified by CDAYR and CDAYR2 alternately. When captured data has been written for the number of bytes set by CBDSR, a write end interrupt corresponding to each address setting register occurs. Also, only in data enable fetch, a bundle write end interrupt occurs when bundle write has finished after write for one-frame capture ends. Table 53.9 shows the correspondence between address setting registers and write end interrupt sources. Figure 53.42 shows the timing of write end interrupts in image capture and data synchronous fetch. Figure 53.43 shows the timing of write end interrupts in data enable fetch. 0: Normal write 1: Bundle write	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDOCR sets the format for outputting captured data to the memory. In data fetch mode, set the CDS bit to 1.

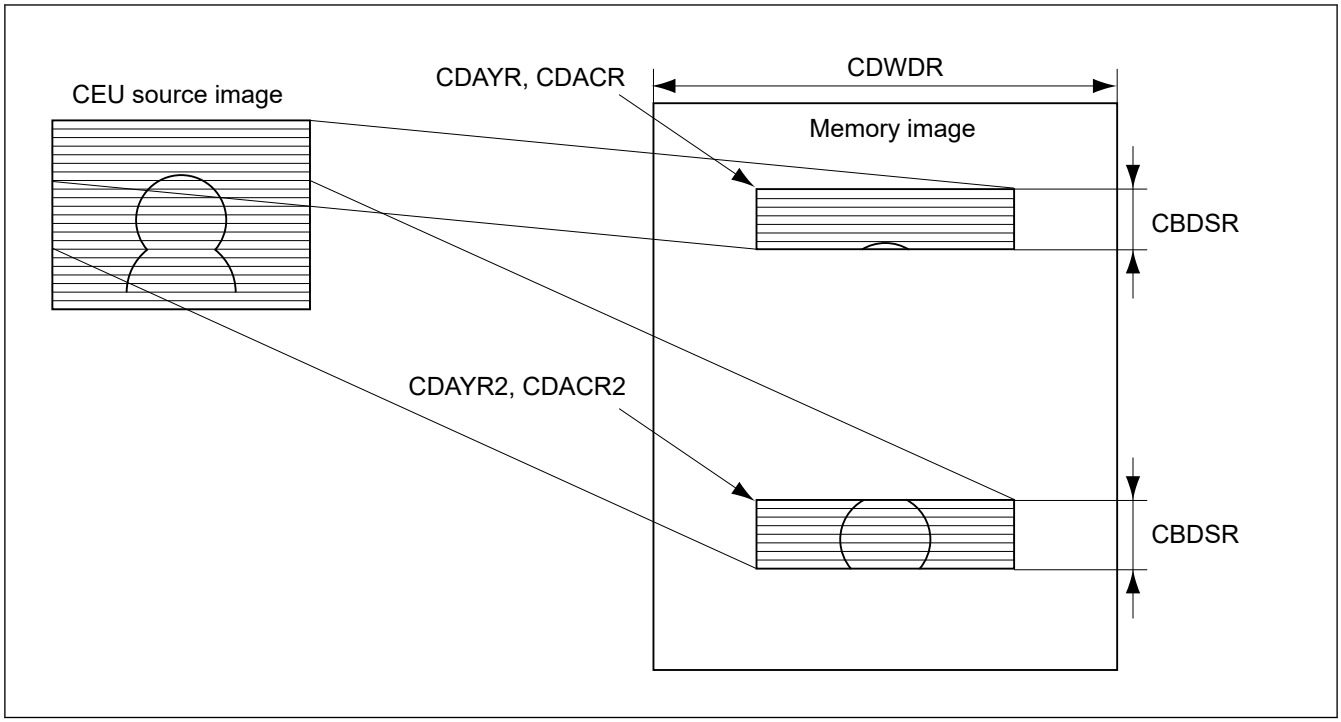


Figure 53.41 Image of bundle write to memory

Table 53.9 Correspondence between address setting registers and write end interrupt sources

Address setting registers	Bundle write end interrupt source
CDAYR, CDACR	CPBE1 bit in CETCR
CDAYR2, CDACR2	CPBE2 bit in CETCR
CDBYR, CDBCR	CPBE3 bit in CETCR
CDBYR2, CDBCR2	CPBE4 bit in CETCR

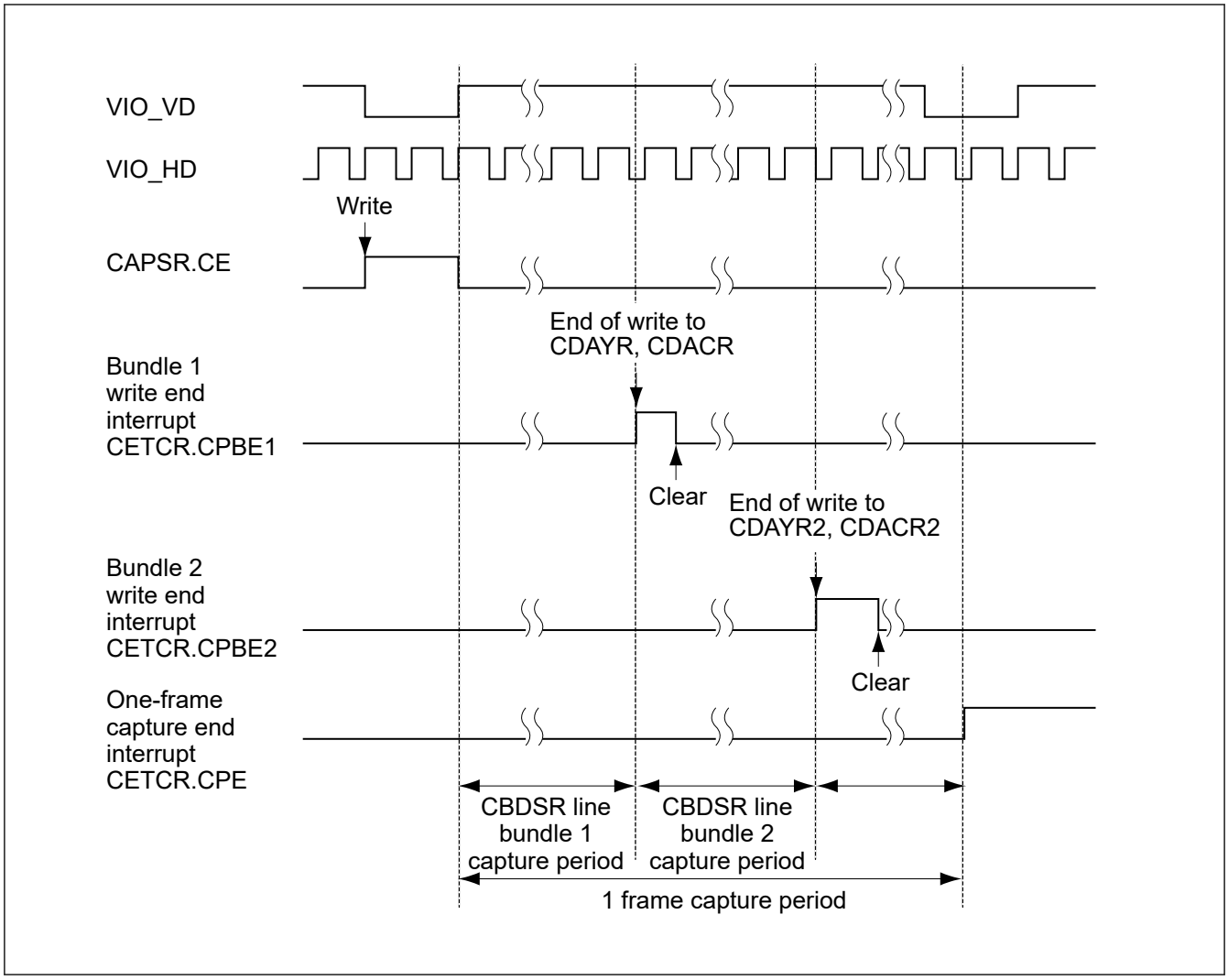


Figure 53.42 Timing of write end interrupts (Image capture, data synchronous fetch)

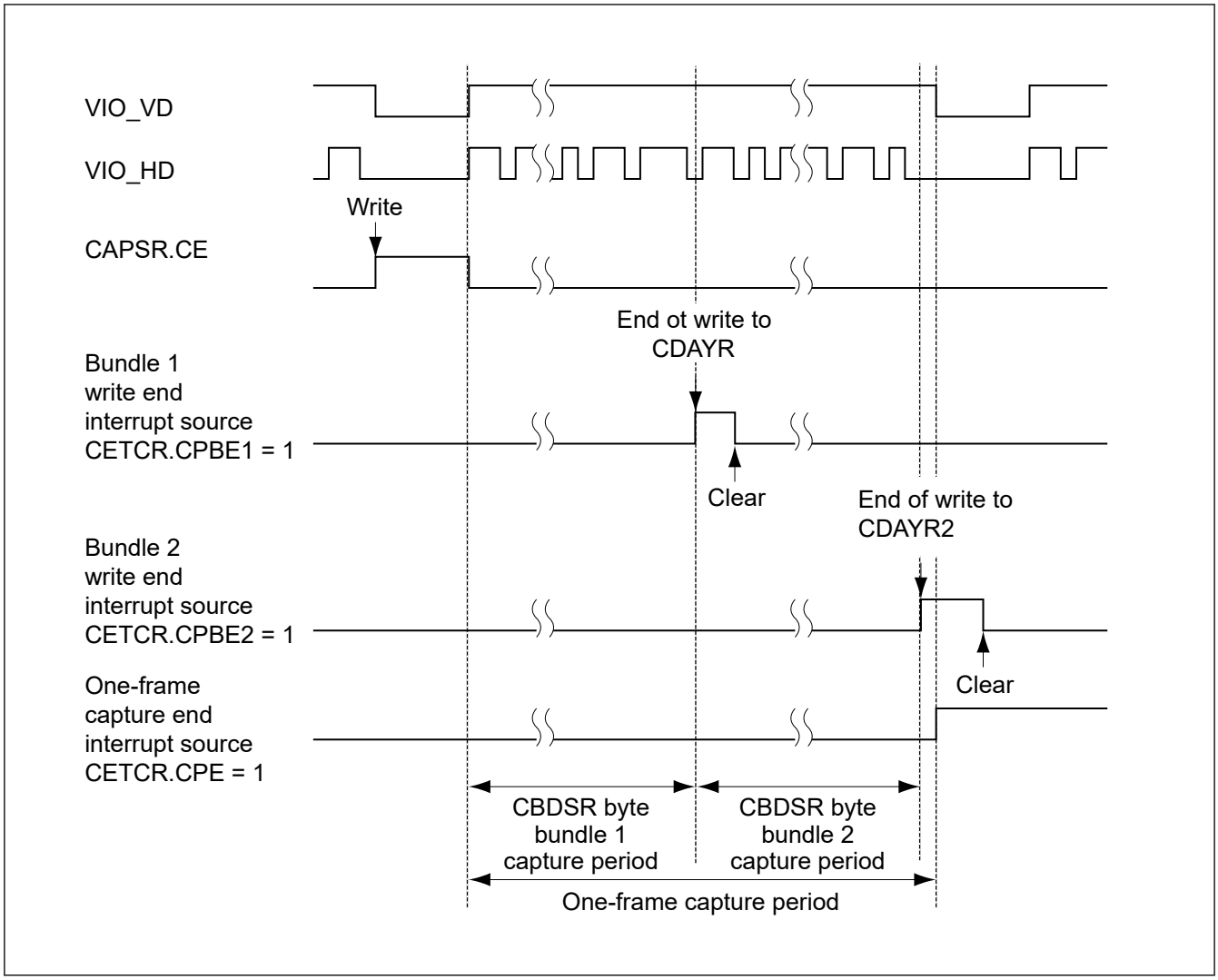


Figure 53.43 Timing of write end interrupts (data enable fetch)

For data output from the CEU, the COLS, COWS, and COBS bits control swapping in 32-bit, 16-bit, and 8-bit units, respectively. Set these bits when data is misaligned because of a difference in endian. The data swapping bits are shown below. These bits can be set similarly in data fetch mode.

Data can be swapped in 8-bit, 16-bit, 32-bit units, or in 32 bits, 16 bits and 8 bits, as shown in Figure 53.44. To enable data swapping, set the corresponding control bit to 1.

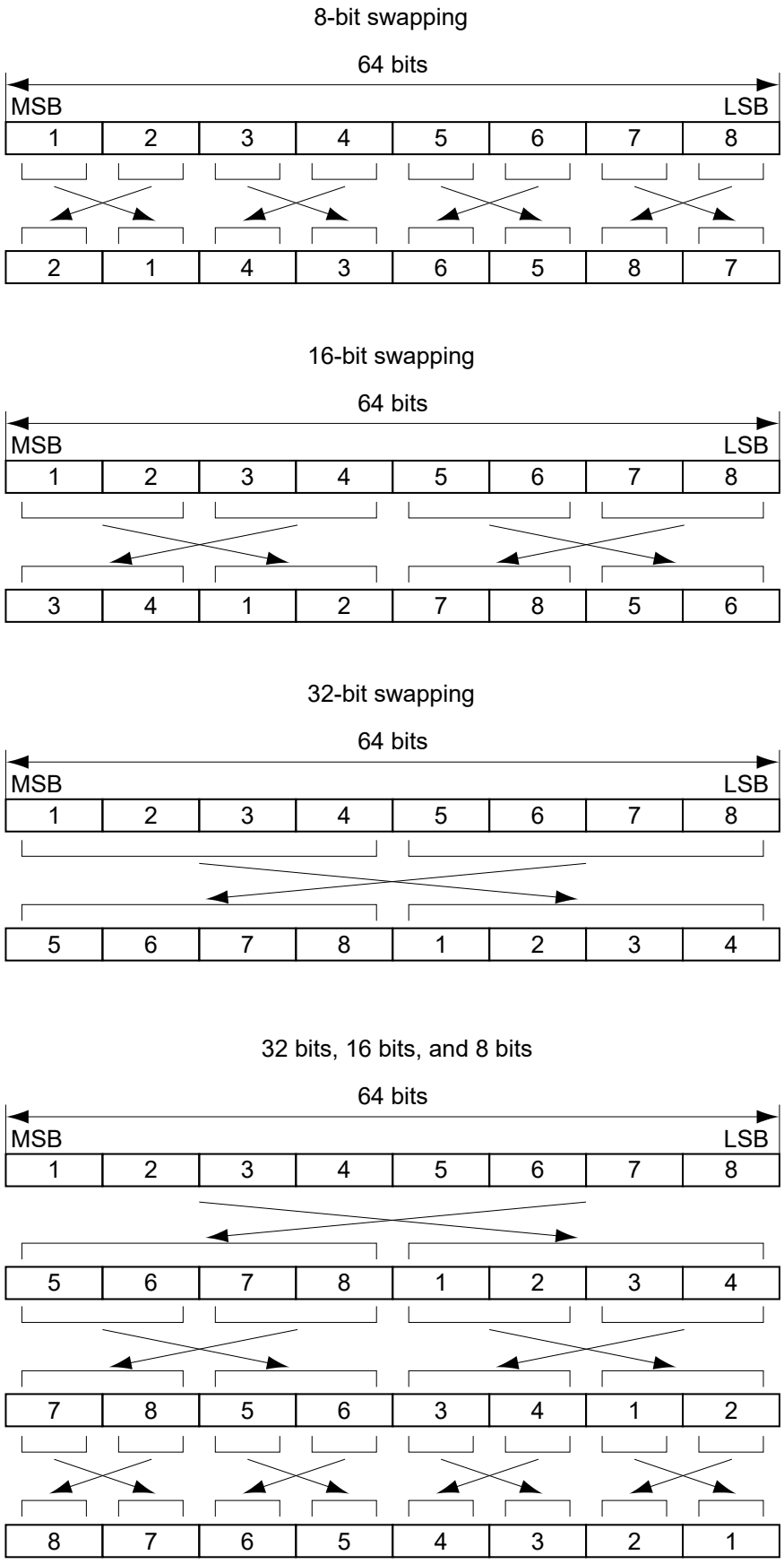


Figure 53.44 Data swapping by data aligner

53.2.21 CEIER : Capture Event Interrupt Enable Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0070

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	NVDIE	NHDIE	FWFIE	—	—	VBPIE	—	IGVSIE	IGHSIE	CDTOFIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CPBE4IE	CPBE3IE	CPBE2IE	CPBE1IE	—	—	VDIE	HDIE	—	—	—	IGRWIE	—	—	CFEIE	CPEIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPEIE	One-Frame Capture End Interrupt Enable 0: Disables a one-frame capture end interrupt 1: Enables a one-frame capture end interrupt	R/W
1	CFEIE	CFE Interrupt Enable 0: Disables a CFE interrupt 1: Enables a CFE interrupt	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	IGRWIE	Register-Access-During-Capture Interrupt Enable 0: Disables a register-access-during-capture interrupt 1: Enables a register-access-during-capture interrupt	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	HDIE	HD Interrupt Enable 0: Disables an HD interrupt 1: Enables an HD interrupt	R/W
9	VDIE	VD Interrupt Enable 0: Disables a VD interrupt 1: Enables a VD interrupt	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	CPBE1IE	CPBE1 Interrupt Enable 0: Disables a CPBE1 interrupt 1: Enables a CPBE1 interrupt	R/W
13	CPBE2IE	CPBE2 Interrupt Enable 0: Disables a CPBE2 interrupt 1: Enables a CPBE2 interrupt	R/W
14	CPBE3IE	CPBE3 Interrupt Enable 0: Disables a CPBE3 interrupt 1: Enables a CPBE3 interrupt	R/W
15	CPBE4IE	CPBE4 Interrupt Enable 0: Disables a CPBE4 interrupt 1: Enables a CPBE4 interrupt	R/W
16	CDTOFIE	CDTOF Interrupt Enable 0: Disables a CDTOF interrupt 1: Enables a CDTOF interrupt	R/W
17	IGHSIE	IGHS Interrupt Enable 0: Disables an IGHS interrupt 1: Enables an IGHS interrupt	R/W
18	IGVSIE	IGVS Interrupt Enable 0: Disables an IGVS interrupt 1: Enables an IGVS interrupt	R/W

Bit	Symbol	Function	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
20	VBPIE	VBP Interrupt Enable 0: Disables a VBP interrupt 1: Enables a VBP interrupt	R/W
22:21	—	These bits are read as 0. The write value should be 0.	R/W
23	FWFIE	FWF Interrupt Enable 0: Disables a FWF interrupt 1: Enables a FWF interrupt	R/W
24	NHDIE	Non-HD Interrupt Enable Disable this interrupt (NHDIE = 0) for data enable fetch. 0: Disables a non-HD interrupt 1: Enables a non-HD interrupt	R/W
25	NVDIE	Non-VD Interrupt Enable Disable this interrupt (NVDIE = 0) for data enable fetch. 0: Disables a non-VD interrupt 1: Enables a non-VD interrupt	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CEIER enables or disables interrupts of the event flag register that generates CEU interrupts.

53.2.22 CETCR : Capture Event Flag Clear Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0074

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	NVD	NHD	FWF	—	—	VBP	—	IGVS	IGHS	CDTOF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CPBE4	CPBE3	CPBE2	CPBE1	—	—	VD	HD	—	—	—	IGRW	—	—	CFE	CPE
Value after reset:	0	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPE	An interrupt indicating that capturing of one frame from an external module has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next VD input. This interrupt indicates that capturing of one frame has finished. This bit is set to 1 when the image of the size set in CAPWR is captured and the last data transfer to the bus finished (see Figure 53.46).	R/W
1	CFE	An interrupt indicating that capturing of one field from an external module has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next VD input (see Figure 53.45). This interrupt occurs only in both-field capture mode.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	IGRW	An interrupt indicating that during capturing, access was attempted to a register to which writing during operation is prohibited. Among the CEU registers, writing during capturing is prohibited for some registers. Table 53.10 shows which registers can/cannot be written to during capturing. This bit is set to 1 when a register to which writing during capturing is prohibited has been written to.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	HD	An interrupt indicating that HD (horizontal sync signal) was input from an external module. This bit is set to 1 when an HD input from an external module is detected. Immediately after the HDPOL bit in CAMCR is modified, a pseudo HD is input and this bit is set to 1. The HD interrupt after the HDPOL bit is modified should be ignored.	R/W
9	VD	An interrupt indicating that VD (vertical sync signal) was input from an external module. In data enable fetch mode, this bit is set to 1 when a VD input from an external module is detected. In image capture mode and data synchronous fetch mode, this module generates a VD interrupt in response to the first detection of the active level of HD following detection of the active level of VD from an external module. Note that, when VD and HD are asserted and detected at the same time, this module generates a VD interrupt at that time. Immediately after the VDPOL bit in CAMCR is modified, a pseudo VD is input and this bit is set to 1. The VD interrupt after the VDPOL bit is modified should be ignored.	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	CPBE1	An interrupt indicating that writing to CDAYR and CDACR in a bundle write has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next HD input. This bit is set to 1 when data for the number of lines (number of bytes in data enable fetch) set in CBDSR has been captured and the last data transfer to the bus has completed. However, in image capture or data synchronous fetch, this interrupt does not occur when the last captured data in a bundle write is the last captured data of a frame (field).	R/W
13	CPBE2	An interrupt indicating that writing to CDAYR2 and CDACR2 in a bundle write has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next HD input. This bit is set to 1 when data for the number of lines (number of bytes in data enable fetch) set in CBDSR has been captured and the last data transfer to the bus has completed. However, in image capture or data synchronous fetch, this interrupt does not occur when the last captured data in a bundle write is the last captured data of a frame (field).	R/W
14	CPBE3	An interrupt indicating that writing to CDBYR and CDBCR in a bundle write has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next HD input. This bit is set to 1 when data for the number of lines set in CBDSR has been captured and the last data transfer to the bus has completed. However, this interrupt does not occur when the last captured data in a bundle write is the last captured data of a frame (field).	R/W
15	CPBE4	An interrupt indicating that writing to CDBYR2 and CDBCR2 in a bundle write has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next HD input. This bit is set to 1 when data for the number of lines set in CBDSR has been captured and the last data transfer to the bus has completed. However, this interrupt does not occur when the last captured data in a bundle write is the last captured data of a frame (field).	R/W
16	CDTOF	An interrupt indicating that data overflowed in the CRAM of the write buffer Since data is input at realtime from an external module in capture operations, the frame image is overwritten unless the captured data is transferred from the CEU internal buffer to the memory at a certain or higher transfer rate. This bit is set to 1 when writing the data in the CRAM of the CEU internal write buffer to the bus is not performed within time and data has overflowed.	R/W
17	IGHS	An interrupt generated when the number of HD cycles set in CMCYR differ from the number of HD cycles input from an external module. This bit is set to 1 when there is an illegal HD input from an external module. This bit is set to 1 when the number of clock cycles for the HD input to the CEU differs from the value set in the HCYL bits in CMCYR. Note however that when the HCYL bits are cleared to 0, this interrupt does not occur.	R/W
18	IGVS	An interrupt generated when the number of VD cycles set in CMCYR differ from the number of VD cycles input from an external module. This bit is set to 1 when there is an illegal VD input from an external module. This bit is set to 1 when the number of HD cycles for the VD input to the CEU differs from the value set in the VCYL bits in CMCYR. Note however that when the VCYL bits are cleared to 0, this interrupt does not occur.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
20	VBP	An interrupt indicating that VD has been input while the CEU holds data (insufficient vertical-sync front porch). The conditions for a VBP interrupt to occur are as follows: ● Condition 1 VD is input when there is captured data within the CEU ● Condition 2 The last transfer data cannot be internally detected due to a write buffer overflow or an illegal HD so that the end timing is unclear until the next VD (By generating a VBP interrupt at the VD input timing, capture fail can be announced.) When a VBP interrupt occurs, a capture end interrupt (CPE bit in CETCR) does not occur and the image of that frame is not captured correctly. Though a capture end interrupt (CPE bit) will occur on rare occasions, it should be ignored in this case. Capturing cannot be performed until the next VD (even if the CE bit (capture reservation signal) in CAPSR is 1, capturing does not start). In the case of condition 2, instead of waiting for a VBP interrupt to occur, execute a software reset (CPKIL bit in CAPSR) to stop capturing and then restart capturing. In this case, since capture operation is terminated without waiting for the next VD, a VBP interrupt does not occur and capturing can be performed from the next VD.	R/W
22:21	—	These bits are read as 0. The write value should be 0.	R/W
23	FWF	The interrupt is generated when data is written to the address that exceeds the value specified with CFWCR.FMV. This bit is set to 1 when data is written to the address that exceeds the value specified with CFWCR.FMV while CFWCR.FWE = 1.	R/W
24	NHD	An interrupt indicating that no HD was input. The timing for a non-HD interrupt to occur differs depending on the bit width of the digital image input pins. 8-bit digital image input pins: Occurs when the 11-bit internal counter that is incremented every eight cycles becomes full. Accordingly, this bit is set to 1 when no HD has been input for at least 16,376 cycles since the last HD was input. 16-bit digital image input pins: Occurs when the 12-bit internal counter that is incremented every four cycles becomes full. Accordingly, this bit is set to 1 when no HD has been input for at least 16,380 cycles since the last HD was input. When connecting a camera whose HD is fixed low when VD is low, this bit may be set to 1. Ignore this interrupt during data enable fetch.	R/W
25	NVD	An interrupt indicating that no VD was input. A non-VD interrupt occurs when the 14-bit internal counter becomes full. Accordingly, this bit is set to 1 when no VD has been input for at least 16,383 lines since the last VD was input.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CETCR notifies the CPU of the source of an interrupt that is generated in the CEU. The flags of CETCR can be used as interrupt signals. When the corresponding interrupt is enabled, an interrupt (CEU_CEUI) is generated. To clear the interrupt, clear the bit corresponding to the interrupt source to 0. After several cycles have passed after modifying the bit, the interrupt is cleared.

To clear the bit corresponding to the interrupt source to be cleared to 0 and retain that state, write 1 to that bit. For example, to clear only the CPE bit to 0, write 0xFFFFFFFE to CETCR.

In CETCR, only bits to which 0 is written are cleared. Bits to which 1 is written retain their current values. To clear an interrupt source, write 0 only to the bit corresponding to the interrupt source to be cleared, and 1 to the other bits.

Note: Since the CETCR value becomes undefined in the following cases, clear all bits in CETCR to 0.
- VD and HD bits immediately after system reset or Software Standby mode is entered.
- VD and HD bits after the polarities of the capture interface sync signals are changed.

Table 53.10 Registers that can/cannot be modified during capturing (1 of 2)

Register name	Modification during capturing
CAPSR	Possible
CAPCR	Prohibited
CAMCR	Prohibited

Table 53.10 Registers that can/cannot be modified during capturing (2 of 2)

Register name	Modification during capturing
CMCYR	Prohibited
CAMOR	Possible
CAPWR	Possible
CAIFR	Prohibited
CRCNTR	Possible
CRCMPR	Prohibited
CFLCR	Possible
CFSZR	Possible
CDWDR	Possible
CDAYR	Possible
CDACR	Possible
CDBYR	Possible
CDBCR	Possible
CBDSR	Possible
CFWCR	Possible
CLFCR	Possible
CDOCR	Possible
CEIER	Possible
CETCR	Possible
CSTSR	Prohibited
CDSSR	Prohibited
CDAYR2	Possible
CDACR2	Possible
CDBYR2	Possible
CDBCR2	Possible

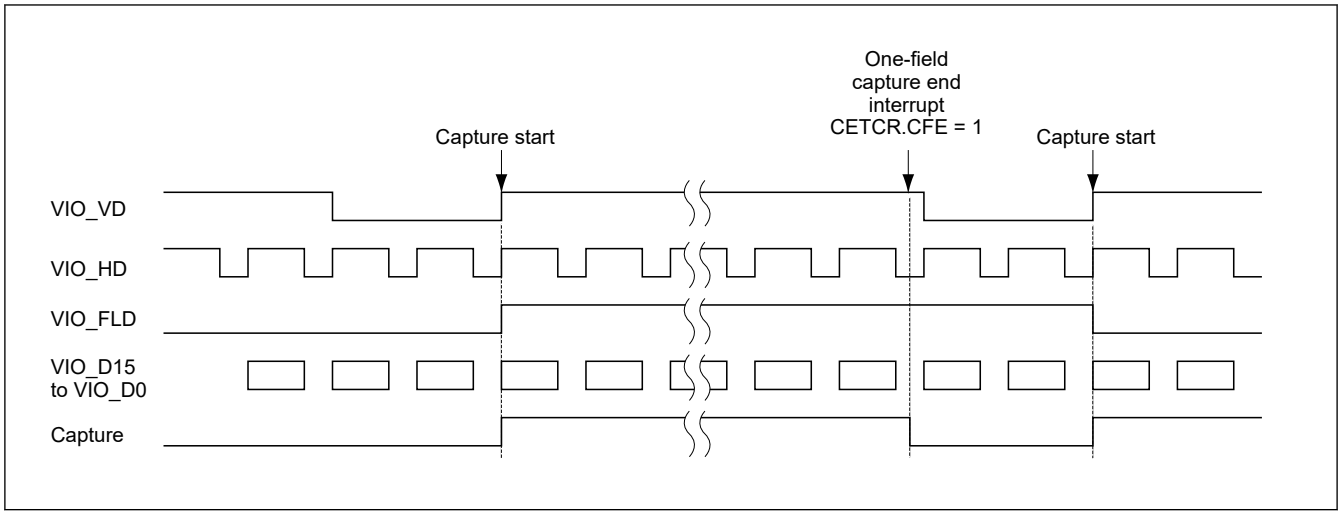


Figure 53.45 CFE generation timing

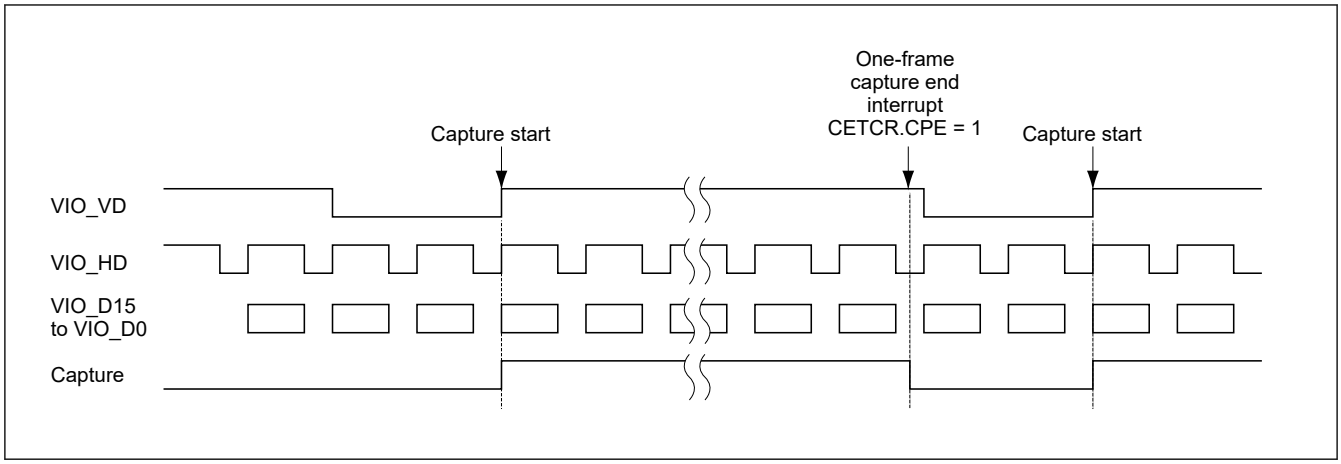


Figure 53.46 CPE generation timing

53.2.23 CSTSR : Capture Status Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x007C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	CRST	—	—	—	—	—	—	—	CPFLD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPTON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPTON	Indicates that the CEU is operating. This bit retains 1 during the period that starts from the internal VD at capture start and ends when a one-frame capture end interrupt occurs. Figure 53.47 shows the CEU operating period.	R
15:1	—	These bits are read as 0.	R
16	CPFLD	Indicates which field is being captured. 0: Bottom field is being captured 1: Top field is being captured	R
23:17	—	These bits are read as 0.	R
24	CRST	Indicates which register plane is currently used. 0: Plane A of the register is being used 1: Plane B of the register is being used	R
31:25	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

CSTSR indicates the internal status of the CEU. CSTSR differs from CETCR in that no interrupt is generated for the events indicated in CSTSR.

The CEU operating/halt state can be confirmed using CSTSR. To confirm the halt state of the CEU, make sure that the status bit (bit 0) indicating that the CEU is operating is cleared to 0 for sure.

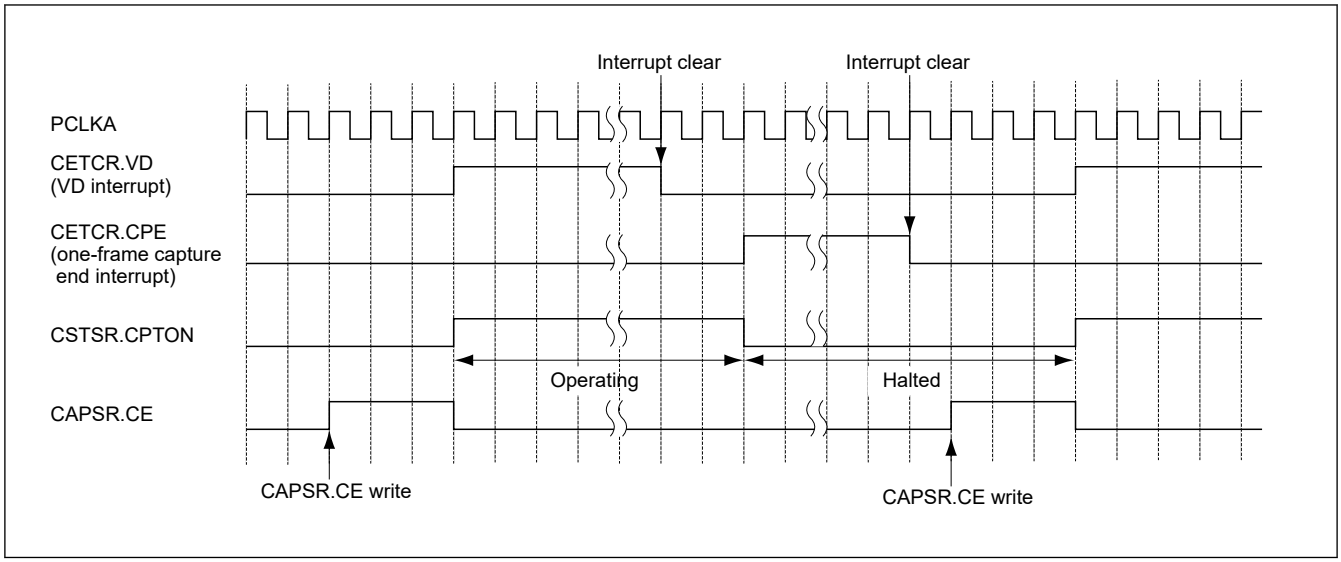


Figure 53.47 Operating status during capturing

53.2.24 CDSSR : Capture Data Size Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0084

Bit position:	31																												0
Bit field:	CDSS[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CDSS[31:0]	Indicate the size of data written to the memory in data enable fetch. In a bundle write, size of data written to the selected address at the end of one-frame capture is indicated. In a bundle write, as soon as the number of bytes specified by CBDSR is transferred to the bus, address to which data is written is switched. Therefore, if one-frame capture is completed at the same time as a bundle write is completed, this register indicates 0x00000000. Figure 53.48 and Figure 53.49 show the overall timing of the CDSSR operation in a bundle write.	R

Note: S-TYPE-3, P-TYPE-3

CDSSR indicates the size of data written to the memory in data enable fetch. As this register indicates a correct value at the end of capture, confirm this register when capture is completed.

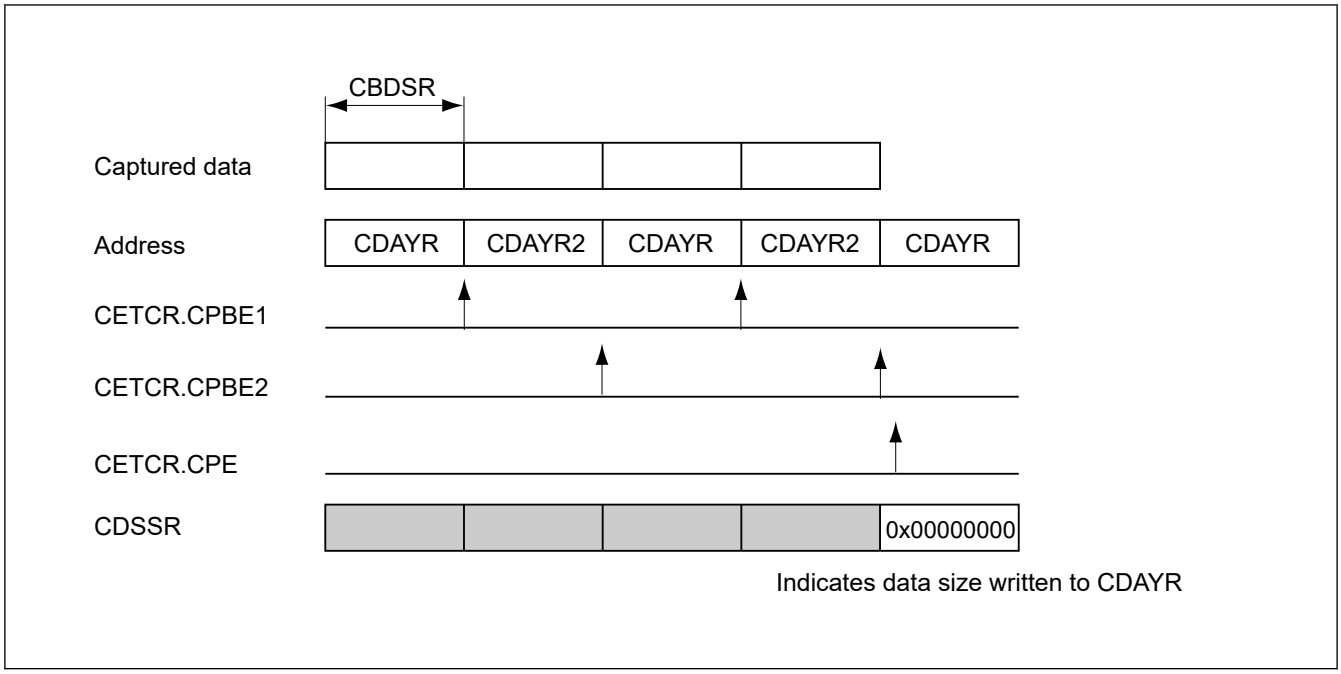


Figure 53.48 Overall timing of CDSSR operation in bundle write (when bundle write end and capture end coincide)

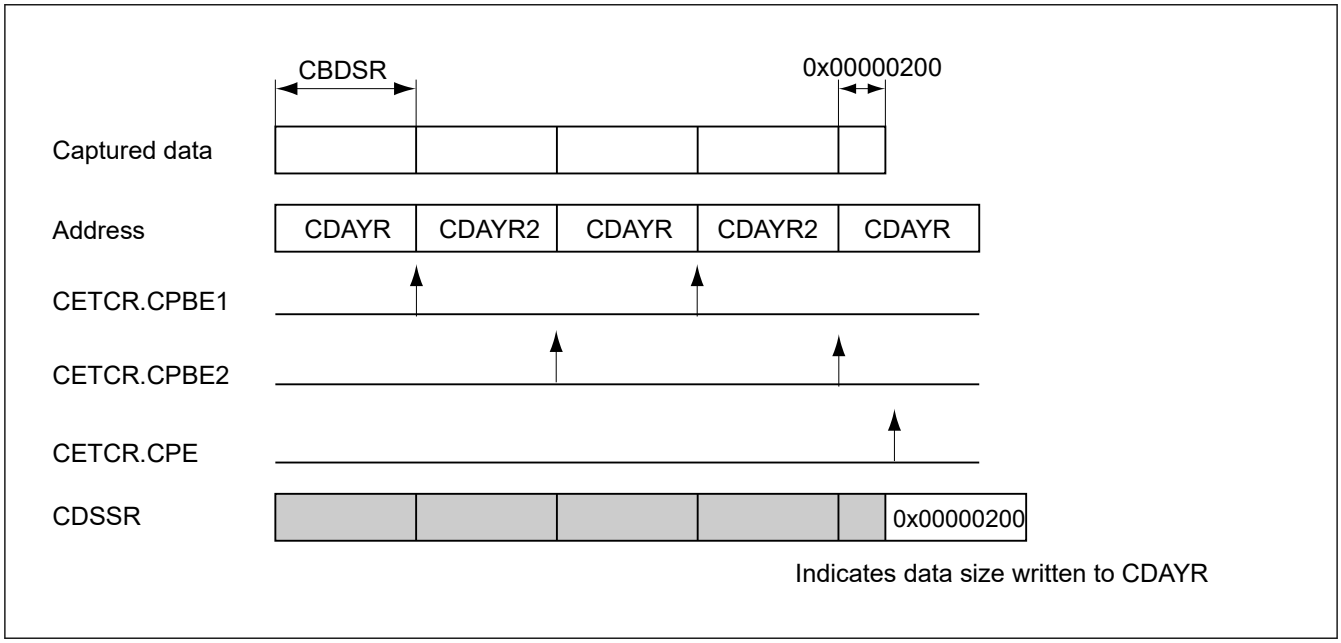


Figure 53.49 Overall timing of CDSSR operation in bundle write (when bundle write end and capture end do not coincide)

53.2.25 CDAYR2, CDAYR2_x : Capture Data Address Y Register 2 (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0090 (CDAYR2)
0x1090 (CDAYR2_B)
0x2090 (CDAYR2_M)

Bit position:	31																												0
Bit field:	CAYR2[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CAYR2[31:0]	Capture Data Address Y Frame image capture: These bits set the address for storing the luminance component data of the captured data (8-pixel units). ● One-field image capture: These bits set the address for storing the luminance component data of the captured data (8-pixel units). ● Both-field image capture: These bits set the address for storing the luminance component data of the captured top-field data (8-pixel units). ● Data synchronous fetch: These bits set the address for storing data (8-byte units). ● Data enable fetch: These bits set the address for storing data (32-byte units). The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDAYR2 specifies the address for the luminance component used in a bundle write and the address for data storage in a bundle write in data fetch. CDAYR2 is used only in a bundle write.

CDAYR2 specifies the address where the luminance component of the captured data is to be stored in frame image capture or one-field image capture, the address where the luminance component of the captured top field is to be stored in both-field image capture, and the address where data is to be stored in data fetch. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. In frame image capture or one-field image capture, set the start address of the memory area where the luminance component of the captured data is to be stored by CDAYR2. In both-field image capture, set the start address of the memory area where the luminance component of the captured top-field image is to be stored by CDAYR2. In data fetch, set the start address of the memory area to be used for data storage.

The address specified by this register must be in 32-bit units.

Set the address of the starting point of the memory area where the luminance component of the captured image is to be stored in this register, as shown in Figure 53.50.

- Frame image capture: Set the address of the starting point of the memory area where the luminance component of the captured image is to be stored.
- One-field image capture: Set the address of the starting point of the memory area where the luminance component of the captured image is to be stored.
- Both-field image capture: Set the address of the starting point of the memory area where the luminance component of the captured top-field image is to be stored.
- Data synchronous fetch: Set the address of the starting point of the memory area where the captured data is to be stored.
- Data enable fetch: Set the address of the starting point of the memory area where the captured data is to be stored in 32-byte units.

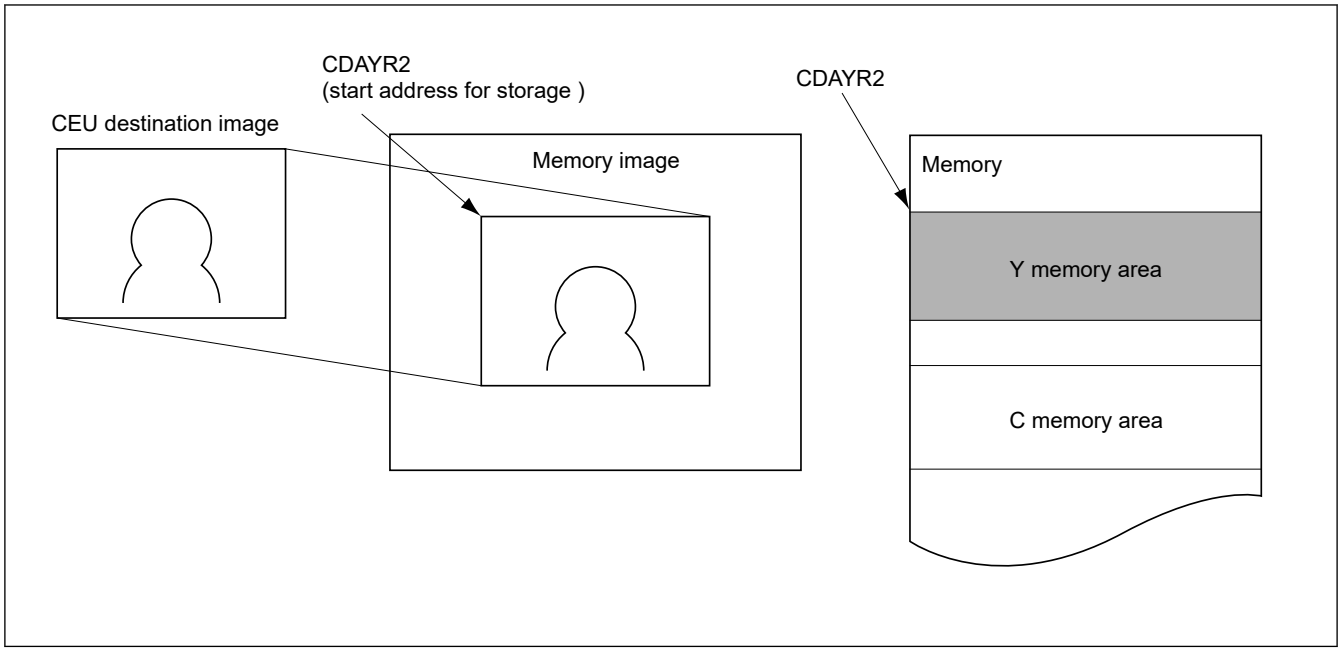


Figure 53.50 Relationship between captured image and luminance component memory area

53.2.26 CDACR2, CDACR2_x : Capture Data Address C Register 2 (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0094 (CDACR2)
0x1094 (CDACR2_B)
0x2094 (CDACR2_M)

Bit position:	31																												0
Bit field:	CACR2[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CACR2[31:0]	Capture Data Address C ● Frame image capture: These bits set the address for storing the chrominance component data of the captured data (8-pixel units). ● One-field image capture: These bits set the address for storing the chrominance component data of the captured data (8-pixel units). ● Both-field image capture: These bits set the address for storing the chrominance component data of the captured top-field data (8-pixel units). The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDACR2 specifies the address for the chrominance component used in a bundle write. CDACR2 is used only in a bundle write.

CDACR2 specifies the address where the chrominance component of the captured data is to be stored in frame image capture or one-field image capture, and the address where the chrominance component of the captured top field is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. In frame image capture or one-field image capture, set the start address of the memory area where the chrominance component of the captured data is to be stored by CDACR2. In bothfield image capture, set the start address of the memory area where the chrominance component of the captured top-field image is to be stored by CDACR2. CDACR2 is not used in data fetch.

The address specified by this register must be in 32-bit units.

In this register, set the address of the starting point of the memory area where the captured data is to be stored in a bundle write, as shown in Figure 53.51.

- Frame image capture: Set the address of the starting point of the memory area where the chrominance component of the captured image is to be stored.
- One-field image capture: Set the address of the starting point of the memory area where the chrominance component of the captured image is to be stored.
- Both-field image capture: Set the address of the starting point of the memory area where the chrominance component of the captured top-field image is to be stored.

The chrominance component has an output data format as shown in Figure 53.52, and is saved in the memory in this format.

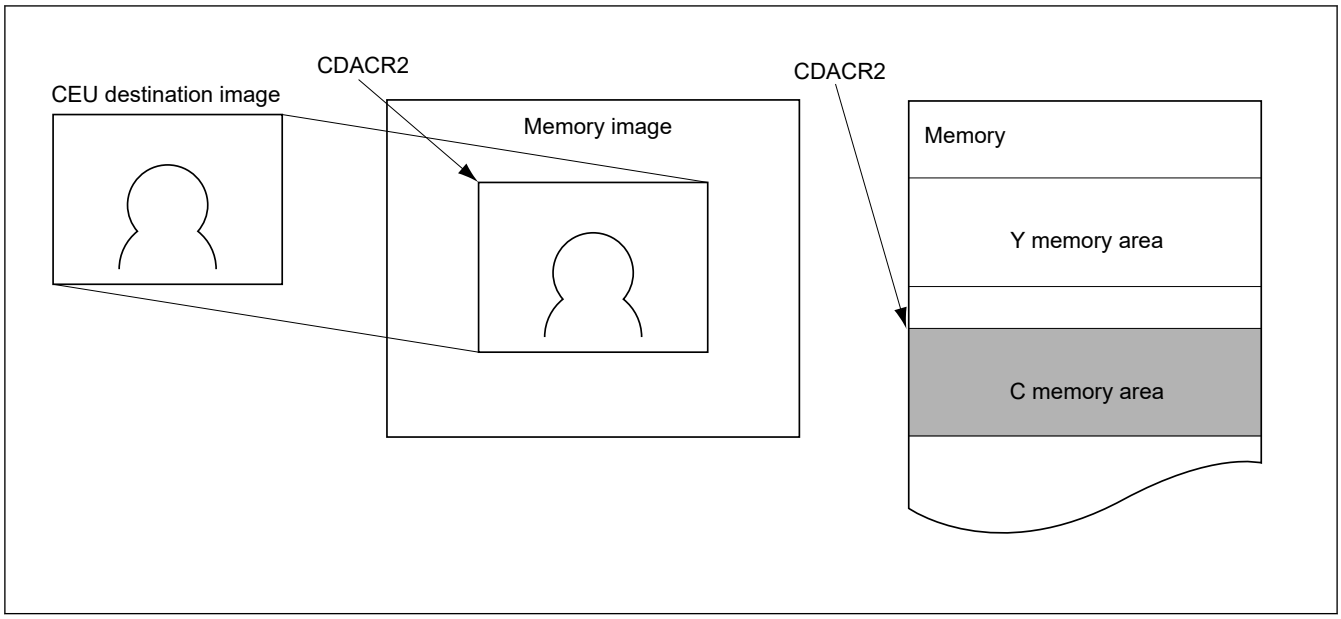


Figure 53.51 Relationship between captured image and chrominance component memory area

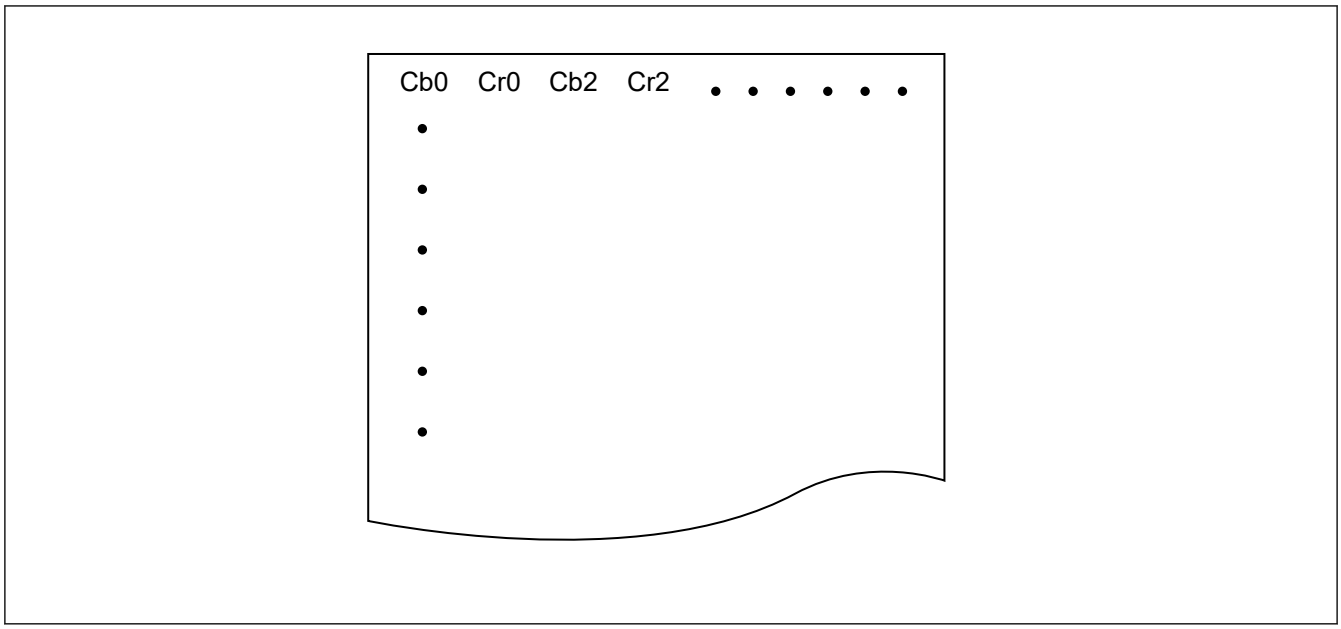


Figure 53.52 Image of storing chrominance components in memory

53.2.27 CDBYR2, CDBYR2_x : Capture Data Bottom-Field Address Y Register 2 (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0098 (CDBYR2)
0x1098 (CDBYR2_B)
0x2098 (CDBYR2_M)

Bit position:	31																												0
Bit field:	CBYR2[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CBYR2[31:0]	Set the address for storing the luminance component data of the captured bottom-field data (8-pixel units). The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDBYR2 specifies the address for the luminance component of the bottom field used in a bundle write. CDBYR2 is used only in a bundle write.

CDBYR2 specifies the address where the luminance component of the captured bottom-field data is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. Set the start address of the memory area where the luminance component of the bottom-field image captured in both-field image capture is to be stored by CDBYR2. CDBYR2 is not used in frame image capture, one-field image capture, or data fetch.

The address specified by this register must be in 32-bit units.

In this register, set the address of the starting point of the memory area where the luminance component of the captured bottomfield image is to be stored in a bundle write, as shown in Figure 53.53.

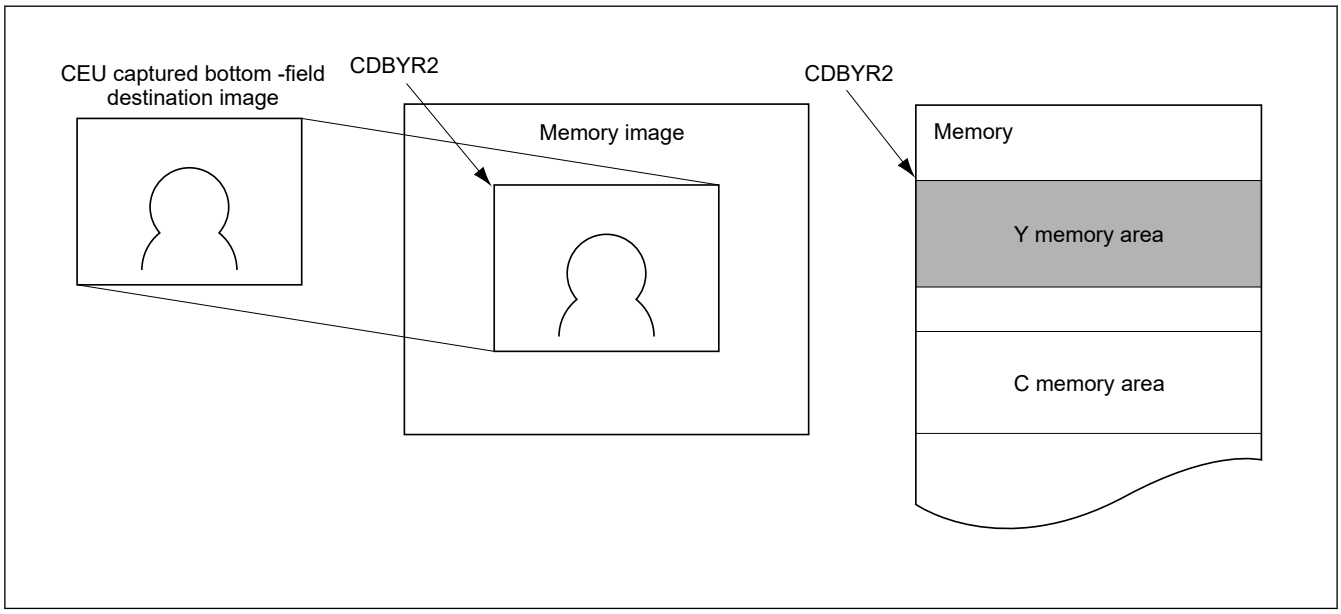


Figure 53.53 Relationship between captured bottom-field image and luminance component memory area

53.2.28 CDBCR2, CDBCR2_x : Capture Data Bottom-Field Address C Register 2 (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x009C (CDBCR2)
0x109C (CDBCR2_B)
0x209C (CDBCR2_M)

Bit position:	31																												0
Bit field:	CBCR2[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CBCR2[31:0]	Set the address for storing the chrominance component data of the captured bottom-field data (8-pixel units). The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDBCR2 specifies the address for the chrominance component of the bottom field used in a bundle write. CDBCR2 is used only in a bundle write.

CDBCR2 specifies the address where the chrominance component of the captured bottom-field data is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. Set the start address of the memory area where the chrominance component of the bottom-field image captured in both-field image capture is to be stored by CDBCR2. CDBCR2 is not used in frame image capture, one-field image capture, or data fetch.

The address specified by this register must be in 32-bit units.

In this register, set the address of the starting point of the memory area where the chrominance component of the captured bottomfield image is to be stored in a bundle write, as shown in Figure 53.54. The chrominance component has an output data format as shown in Figure 53.55, and is saved in the memory in this format.

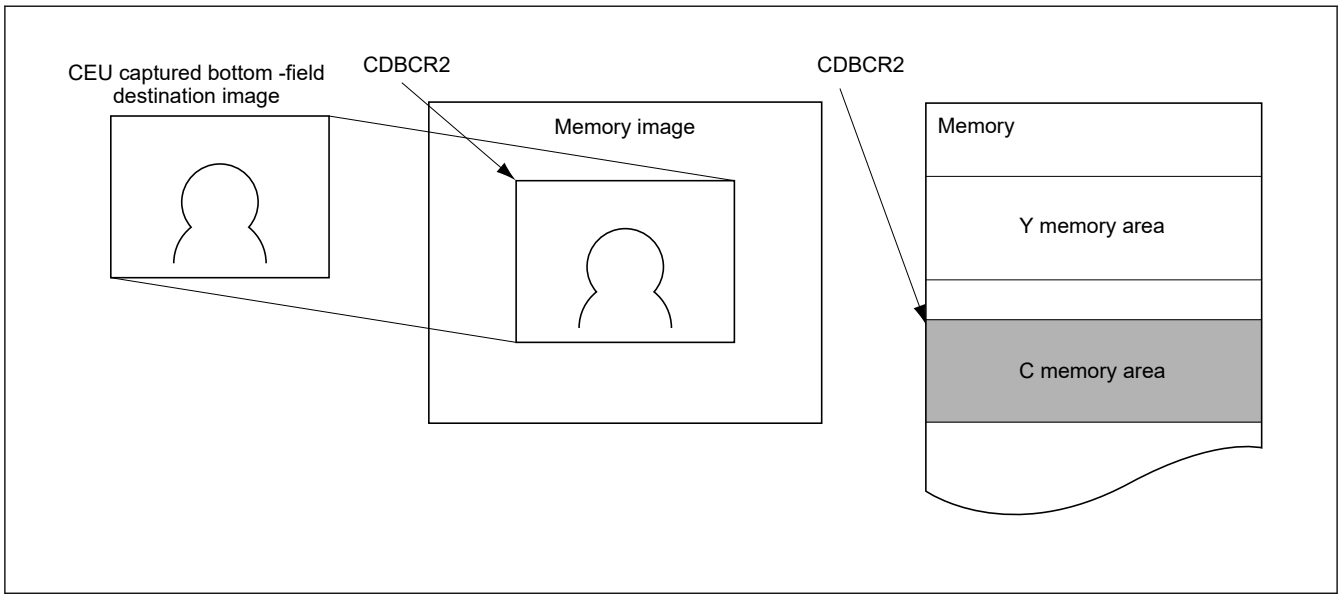


Figure 53.54 Relationship between captured bottom-field image and chrominance component memory area

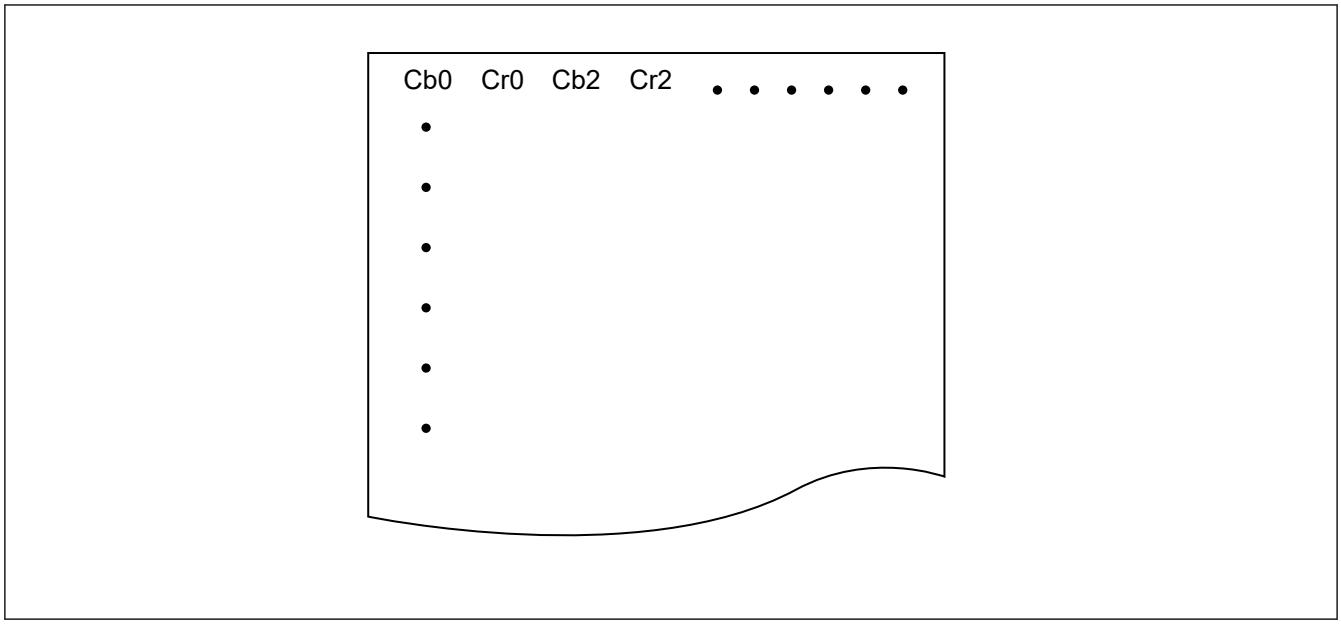


Figure 53.55 Image of storing chrominance components in memory

53.2.29 CBWER : CEU Bufferable Write Enable Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x00A0

Bit position:	31																													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BWE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BWE	Bufferable Write Enable 0: Disables Bufferable Write 1: Enables Bufferable Write	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

BWE bit (Bufferable Write Enable)

BWE bit indicates bufferable write is either enabled or disabled.

Note that if this bit is 1, even if the write access as CEU is completed, the actual slave write may not have ended.

Note: When this bit is 1, it requests an early response by writing write data to the temporary buffer if possible to the slave. Therefore, even if write access as CEU is completed, the actual slave write may not be finished. For details on slave groups that support bufferable writes, see section 14.7.2. Operations When a Bus Error Occurs and respective module sections.

53.3 Usage Notes

53.3.1 Settings for the Module-Stop Function

CEU operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The CEU is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

53.3.2 Conditions for Connection to an External Module

(1) Clock Frequency

The external input clock (VIO_CLK) should have a frequency at most the same as the CEU operating clock (PCLKA) frequency, with jitter on both sides included.

$$PCLKA \geq VIO\_CLK$$

(2) Blanking Period

The period from the last valid pixel in each line to the next horizontal sync signal HD must be at least 20 cycles.

(3) Fixed Period of Field Identification Signal

The field identification signal FLD should be fixed for at least 1-HD period since a VD input

53.3.3 Restrictions on Input/Output Functions

Table 53.11 lists the restrictions regarding the CEU input/output functions.

Table 53.11 Restrictions on CEU Input/Output Functions

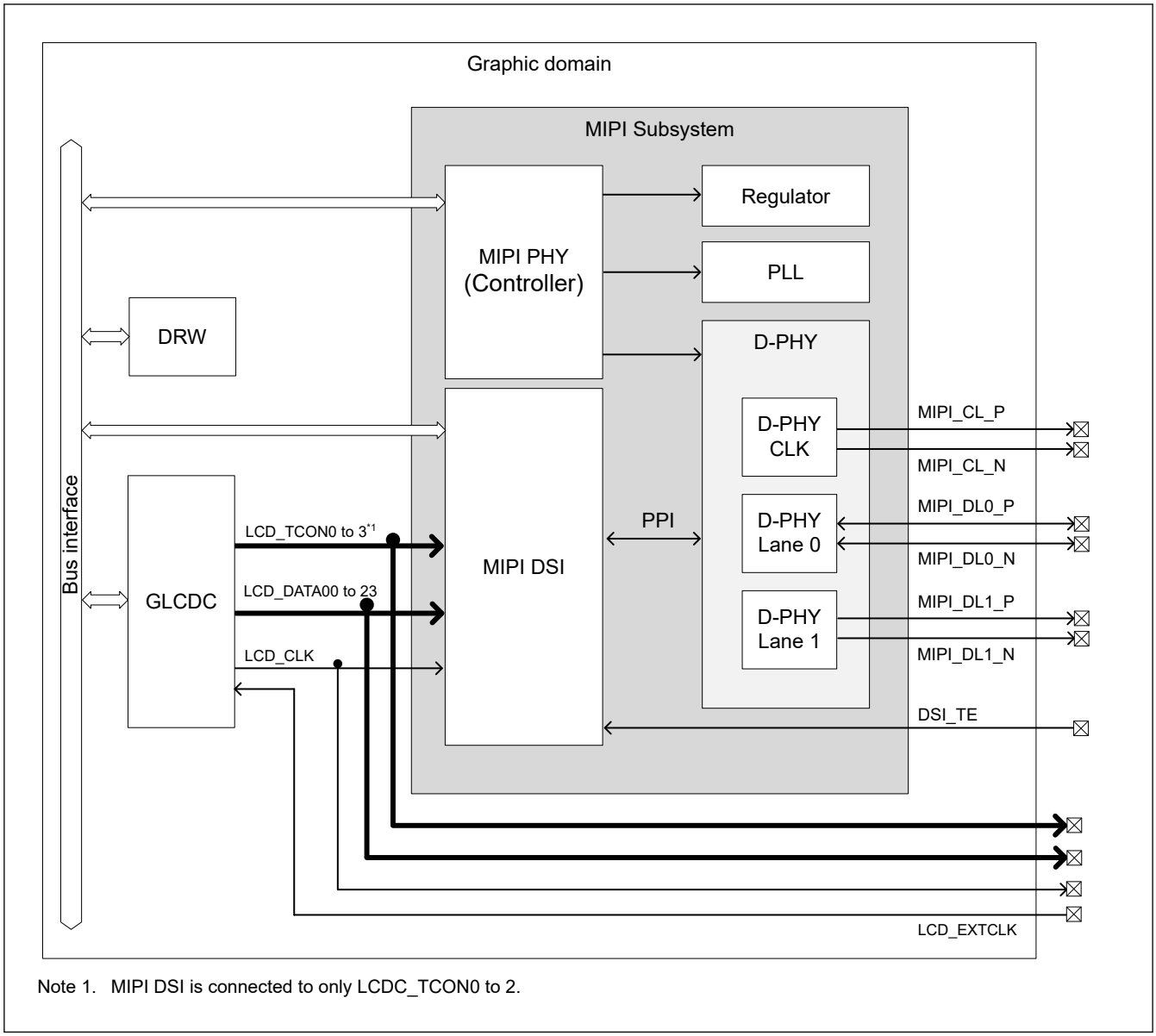
Item	Restrictions
External module interface	The operating clock of the external module (VIO_CLK) should always have a frequency at most the same as that of the CEU operating (PCLKA), with jitter on both sides included.
	Selecting the interface, or modifying the frequency of the external module operating clock or HD/VD polarity must be done when capture operations are halted for sure.
	The capture horizontal size in image capture must be specified as follows: 8-bit interface: 16-cycle units 16-bit interface: 8-cycle units
	The capture horizontal size in data fetch must be specified as follows: 8-bit interface: 8-cycle units 16-bit interface: 4-cycle units
	The capture vertical size must be specified in 4-line units.
	The maximum number of cycles in the horizontal sync signal period should be 16,375 cycles of external input clock for 8-bit digital image input pins, or 16,379 cycles of external input clock for 16-bit digital image input pins.
	The maximum number of lines (HD count) in the vertical sync signal should be 16,382 lines.
	The minimum number of captured pixels should be sub-QCIF (128 × 96).
	The maximum number of captured pixels should be 5 megapixels (2,560 × 1,920).
	The capture size in data enable fetch must be controlled by VD and HD in 4-byte units. The range of the capture size in data enable fetch that can be input is as follows. Maximum: 6 MB (2,048 × 1,536 × 2) Minimum: 16 bytes
Memory output	The output address must be specified in 32-bit units.
	The horizontal size of the destination image (memory) must be specified in 8-pixel units.
	The number of horizontal output pixels (= horizontal clipping size) must be specified in 8-pixel units.
	The number of vertical output lines (HD) (= vertical clipping size) must be specified in 4-line (HD) units.
	In data enable fetch bundle write, the output address must be specified in 32-byte units.
Internal processing	The filter clipping size must be specified as a value equal to or lower than the actual output size of the filter.

54. Graphics Subsystem

54.1 Graphics Overview

The graphic domain has the following features.

- Graphics LCD Controller (GLCDC): 8/16/18/24-bit RGB LCD interface supported.
- 2D Drawing Engine (DRW): Raster and vector graphics supported.
- MIPI D-PHY controller (MIPI PHY): Used for MIPI DSI
- MIPI Display Serial Interface (MIPI DSI): Display Serial interface 2 supported.



Note 1. MIPI DSI is connected to only LCDC_TCON0 to 2.

Figure 54.1 Graphic domain block diagram

Table 54.1 MIPI subsystem I/O pins (1 of 2)

Pin name	I/O	Function
MIPI_CL_P	Output	DSI Clock Lane positive pin
MIPI_CL_N	Output	DSI Clock Lane negative pin
MIPI_DL0_P	I/O	DSI Data Lane 0 positive pin

Table 54.1 MIPI subsystem I/O pins (2 of 2)

Pin name	I/O	Function
MIPI_DL0_N	I/O	DSI Data Lane 0 negative pin
MIPI_DL1_P	Output	DSI Data Lane 1 positive pin
MIPI_DL1_N	Output	DSI Data Lane 1 negative pin
DSI_TE	Input	DSI Tearing Effect pin
AVCC_MIPI	Power	D-PHY Analog Power Connect this pin to VSS_MIPI by a 0.1-µF capacitor. The capacitor should be placed close to the pin.
VCC18_MIPI	Power	D-PHY I/O Power Connect this pin to VSS_MIPI by a 0.1-µF capacitor. The capacitor should be placed close to the pin.
VSS_MIPI	Power	D-PHY GND

54.1.1 Start and stop MIPI DSI

It is recommended that the settings for starting and stopping MIPI DSI be performed in the order shown in Table 54.2 below.

Table 54.2 List of Operation

Function	Item	No	Description
MIPI DSI	Start Setting	1	See section 56.3.1. Overall Control
		2	See section 57.3.1. D-PHY Start-up Procedure
		3	See section 58.3.3. Start of HS Clock
	Stop Setting	1	See section 58.3.4. Stop of HS Clock
		2	See section 57.3.2. D-PHY Stop Procedure
		3	See Stopping and restarting in normal operation or stopping and restarting in abnormal operation in section 56.3.1. Overall Control

55. 2D Drawing Engine (DRW)

55.1 Overview

This MCU incorporates a 2D drawing engine (DRW). Table 55.1 lists the DRW specifications.

Table 55.1 DRW Specifications

Item		Descriptions
Vector drawing engine		Extended rendering primitives supported in hardware ● Lines ● Polygons ● Circles and ellipses ● Quadratic Béziers ● Texture mapping
BitBLT function		Types of BitBLT operations ● Fill ● Copy ● Stretch BitBLT ● Rotate and scale ● Alpha blending ● Bilinear filtering ● Color conversion ● Subpixel exact placement
Color format	Frame buffer	Four types ● A (8) ● RGB (565) ● ARGB (4444) ● ARGB (8888)
	Texture	11 types ● CLUT (1)/I (1) ● CLUT (2)/I (2) ● CLUT (4)/I (4) ● CLUT (8)/I (8) ● A (8) ● ACLUT (44) ● RGB (565) ● ARGB (4444) ● ARGB (1555) ● RGB (888) ● ARGB (8888)
Texture		● Run-length encoding (RLE) unit ● 256-entry color look-up table (CLUT) ● Color keying unit
Render		Two modes for rendering process ● Register mode ● Display list mode
Data arrangement		Little endian
Performance counter		● Two independent 32-bit counters ● Trigger is selectable from 14 events
Interrupt sources		Three interrupt sources ● 2D Drawing Engine bus error interrupt (DRWBUSIRQ) ● Current render process finished interrupt (DRWENUMIRQ) ● Display list interrupt (DRWDLISTIRQ)
Module-stop function		Module-stop state can be set to reduce power consumption
TrustZone Filter		Security and Privilege attribution can be set

The 2D Drawing Engine provides two inputs (texture read and frame buffer read) and one output (frame buffer write). The internal color format is always ARGB (8888). The color formats from the inputs are converted to the internal format on read and converted back on write.

The 2D Drawing Engine also supports a display list mode, which makes it possible to decouple the CPU and graphics controller efficiently and perform rendering in parallel with other CPU activities.

Figure 55.1 shows examples of objects that can be drawn in hardware with the 2D Drawing Engine, Figure 55.2 shows a simplified rendering pipeline setup, and Figure 55.3 shows a block diagram of the module.

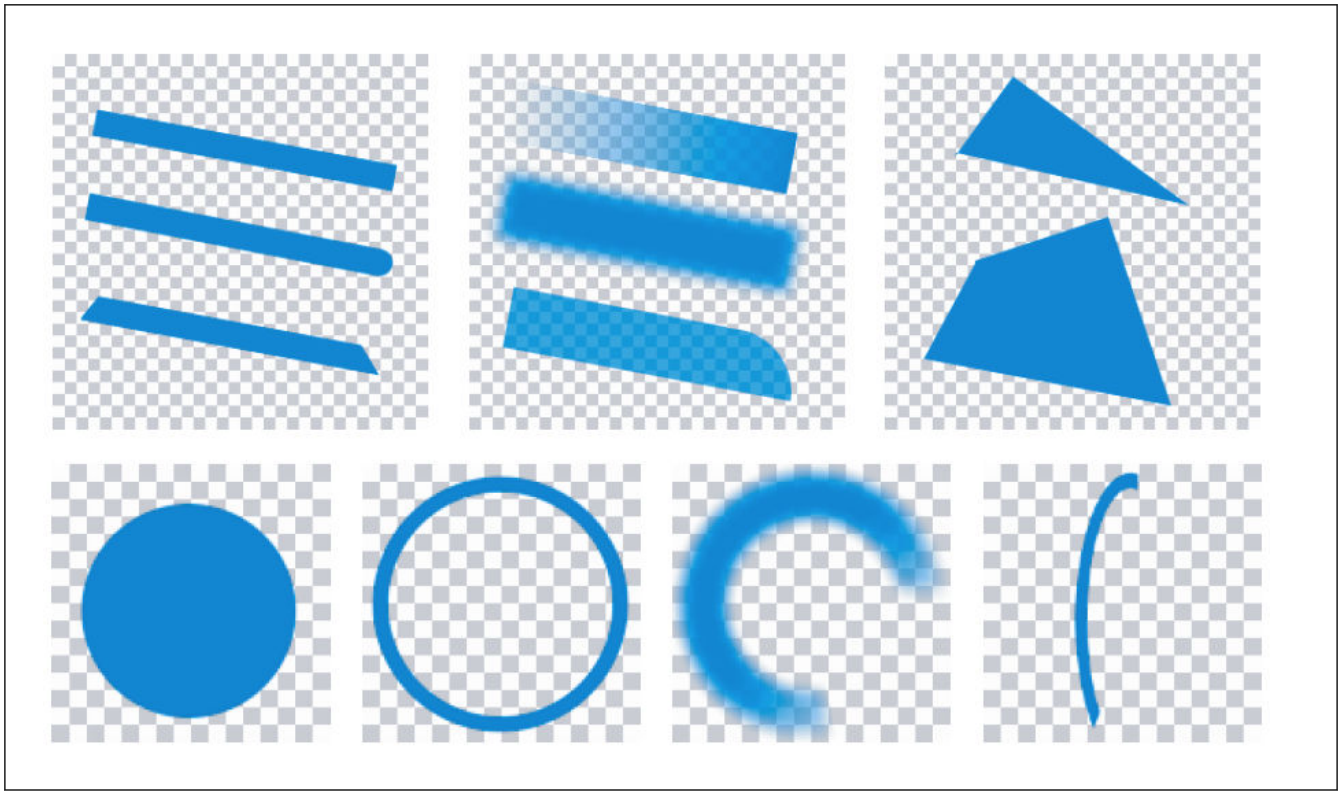


Figure 55.1 Examples of drawing objects

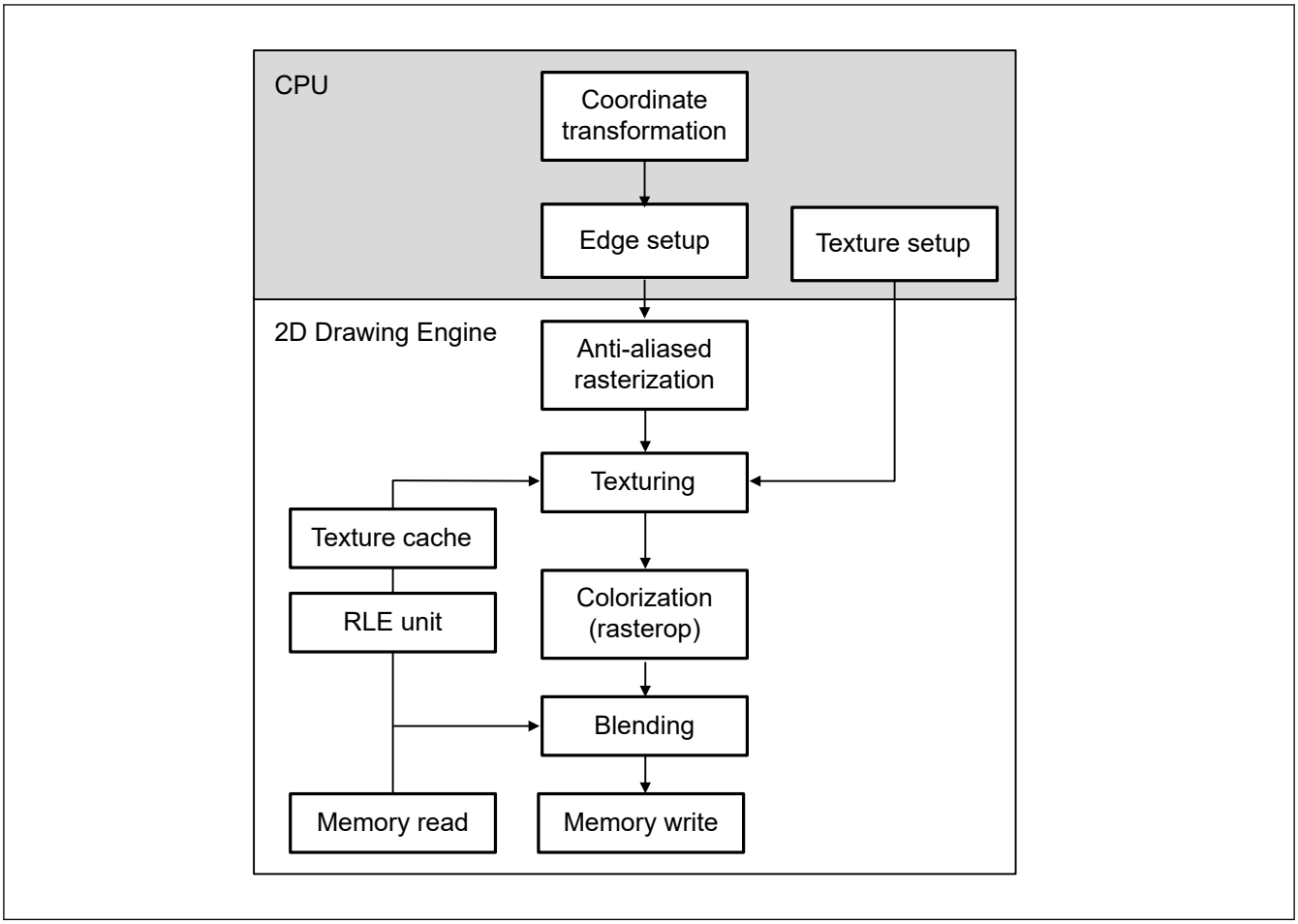


Figure 55.2 Simplified rendering pipeline setup

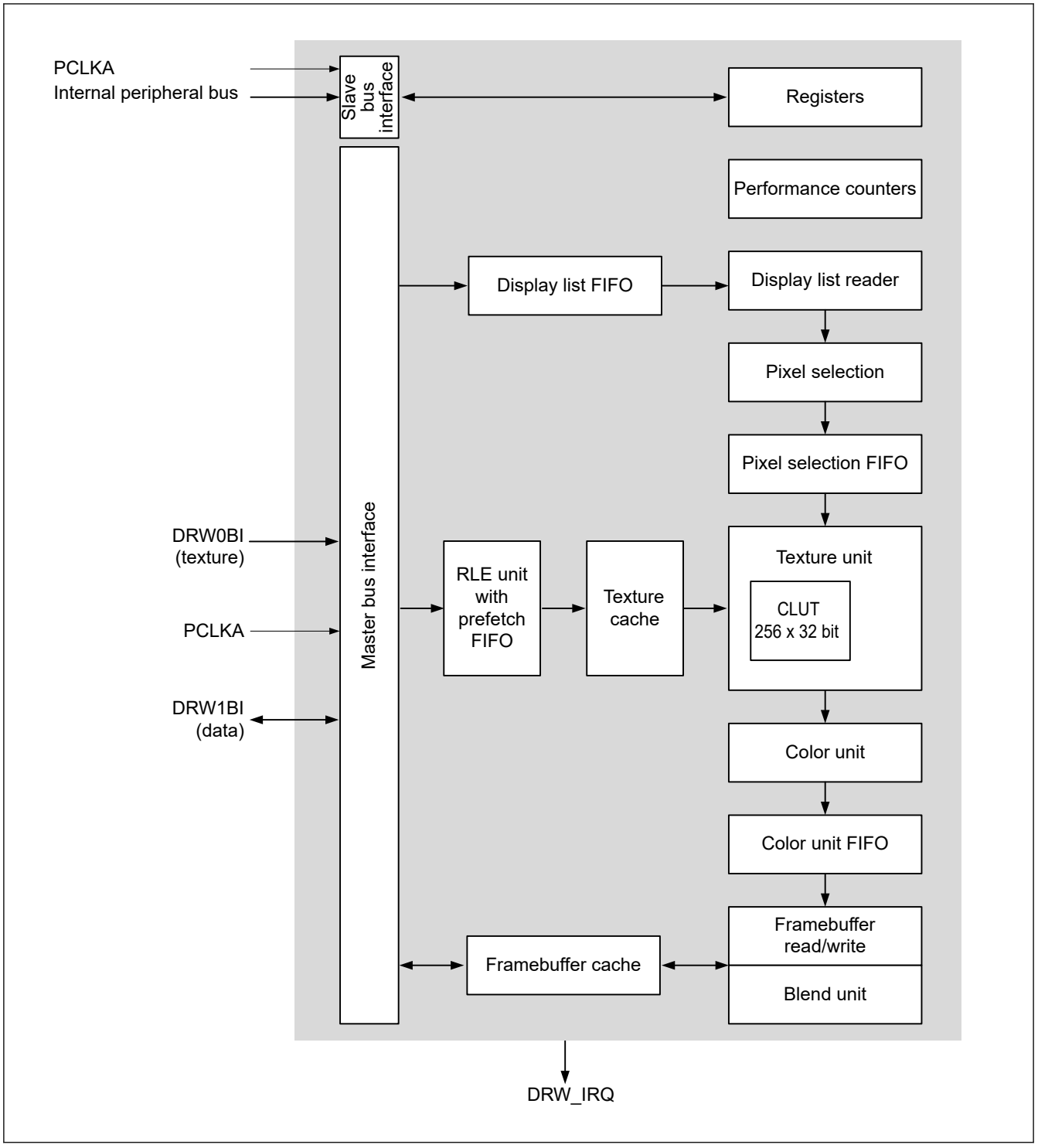


Figure 55.3 2D Drawing Engine block diagram

The 2D Drawing Engine accesses the DRW bus as a bus master through separate caches for:

- Reading and writing pixel data from and to the framebuffer
- Reading textures
- Reading display lists.

The control registers are accessed through the internal peripheral bus interface.

55.2 Register Descriptions

55.2.1 CONTROL : Geometry Control Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	SPANSTORE	SPANABORT	UNIONCD	UNIONAB	UNIONN56	UNIONN34	UNIONN12	BAND2ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BAND1ENABLE	LIM6THRESHOLD	LIM5THRESHOLD	LIM4THRESHOLD	LIM3THRESHOLD	LIM2THRESHOLD	LIM1THRESHOLD	QUAD3ENABLE	QUAD2ENABLE	QUAD1ENABLE	LIM6ENABLE	LIM5ENABLE	LIM4ENABLE	LIM3ENABLE	LIM2ENABLE	LIM1ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LIM1ENABLE	Enable Limiter 1 0: Disable 1: Enable	W
1	LIM2ENABLE	Enable Limiter 2 0: Disable 1: Enable	W
2	LIM3ENABLE	Enable Limiter 3 0: Disable 1: Enable	W
3	LIM4ENABLE	Enable Limiter 4 0: Disable 1: Enable	W
4	LIM5ENABLE	Enable Limiter 5 0: Disable 1: Enable	W
5	LIM6ENABLE	Enable Limiter 6 0: Disable 1: Enable	W
6	QUAD1ENABLE	Enable Quadratic Coupling of Limiters 1 and 2 0: Disable 1: Enable	W
7	QUAD2ENABLE	Enable Quadratic Coupling of Limiters 3 and 4 0: Disable 1: Enable	W
8	QUAD3ENABLE	Enable Quadratic Coupling of Limiters 5 and 6 0: Disable 1: Enable	W
9	LIM1THRESHOLD	Enable Limiter 1 Threshold Mode 0: Disable 1: Enable	W
10	LIM2THRESHOLD	Enable Limiter 2 Threshold Mode 0: Disable 1: Enable	W
11	LIM3THRESHOLD	Enable Limiter 3 Threshold Mode 0: Disable 1: Enable	W
12	LIM4THRESHOLD	Enable Limiter 4 Threshold Mode 0: Disable 1: Enable	W

Bit	Symbol	Function	R/W
13	LIM5THRESHOLD	Enable Limiter 5 Threshold Mode 0: Disable 1: Enable	W
14	LIM6THRESHOLD	Enable Limiter 6 Threshold Mode 0: Disable 1: Enable	W
15	BAND1ENABLE	Enable Band Post Process for Limiter 1 See section 55.2.13. LmBAND : Limiter m Band Width Parameter Register (n = 1, 2). 0: Disable 1: Enable.	W
16	BAND2ENABLE	Enable Band Post Process for Limiter 2 See section 55.2.13. LmBAND : Limiter m Band Width Parameter Register (n = 1, 2). 0: Disable 1: Enable.	W
17	UNION12	Combine Limiters 1 and 2 as Union 0: Select minimum/intersect between limiters 1 and 2 1: Select maximum/union between limiters 1 and 2. The output is called A.	W
18	UNION34	Combine Limiters 3 and 4 as Union 0: Select minimum/intersect between limiters 3 and 4 1: Select maximum/union between limiters 3 and 4. The output is called B.	W
19	UNION56	Combine Limiters 5 and 6 as Union 0: Select minimum/intersect between limiters 5 and 6 1: Select maximum/union between limiters 5 and 6. The output is called D.	W
20	UNIONAB	Combine Outputs A and B as Union 0: Select minimum/intersect between limiters A and B. 1: Select maximum/union between limiters A and B. The output is called C.	W
21	UNIONCD	Combine Outputs C and D as Union 0: Select minimum/intersect between limiters C and D 1: Select maximum/union between limiters C and D. The output is final.	W
22	SPANABORT	Spanabort Shape is horizontally convex, only a single span per scan line. See (2) Spanabort. 0: Disable 1: Enable.	W
23	SPANSTORE	Spanstore 0: Disable 1: Enable. Next line span start is always equal to or left of the current line span start. See (1) Spanstore.	W
31:24	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

55.2.2 CONTROL2 : Surface Control Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RLEPIXELWIDTH[1:0]		BDIA	BSIA	CLUTFORMAT	COLKEYENABLE	CLUTENABLE	RLEENABLE	WRITEALPHA[1:0]		WRITEFORMAT[1:0]		READFORMAT_L[1:0]		TEXTUREFILTERY	TEXTUREFILTERX
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TEXTURECLAMPY	TEXTURECLAMPX	BC2	BDI	BSI	BDF	BSF	WRITEFORMAT2	BDFA	BSFA	READFORMAT_H[3:2]		USEACB	PATTERNSOURCEL5	TEXTUREENABLE	PATTERNENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PATTERNENABLE	Pattern Color Enable for Pixel Source Pixel source is a pattern color (blend of COLOR1 and COLOR2 depending on PATTERN and pattern index). When patterns are used to fill a primitive index into the pattern, bit mask is generated for each pixel with the U limiter. Depending on the pattern bits, the color is selected from COLOR1 and COLOR2 registers. Fractional indices can be interpolated between those two values using TEXTUREFILTERX = 1. The pattern can be wrapped using TEXTURECLAMPX = 0, and the mask must be set in the TEXMASK register using the mask for U limiter. 0: Disable pattern 1: Enable pattern.	W
1	TEXTUREENABLE	Texture Enable for Pixel Source Pixel source is read from texture and used as an alpha to blend between COLOR1 and COLOR2. 0: Disable texture 1: Enable texture.	W
2	PATTERNSOURCEL5	Limiter 5 Enable for Pattern Index Limiter 5 is used as a pattern index instead of the default U limiter. Limiter 5 can be combined with limiter 6 to form a quadratic limiter that can be used to make quadratic pattern functions to draw radial patterns.	W
3	USEACB	Alpha Blend Mode 0: Use WRITEALPHA[1:0] mode 1: Use full alpha channel blending mode.	W
5:4	READFORMAT_H[3:2]	Texture Format Descriptor Bits [3] and [2] of the texture buffer format. See the detailed description of the READFORMAT[1:0] bit in this section.	W
6	BSFA	Blend Source Factor for Alpha Channel Valid in alpha channel blending mode (USEACB = 1). 0: Use 1.0 as blend source factor for alpha channel 1: Use alpha as blend source factor for alpha channel.	W
7	BDFA	Blend Destination Factor for Alpha Channel Valid in alpha channel blending mode (USEACB = 1). 0: Use 1.0 as blend destination factor for alpha channel 1: Use alpha as blend destination factor for alpha channel.	W
8	WRITEFORMAT2	Writeback Framebuffer Format Bit [2] of framebuffer pixel format. See the description of WRITEFORMAT[1:0] in this section.	W
9	BSF	Blend Source Factor Source factor is alpha (factor is 1 per default). 0: Use 1.0 as blend source factor 1: Use alpha as blend source factor.	W
10	BDF	Blend Destination Factor Destination factor is alpha (factor is 1 per default). 0: Use 1.0 as blend destination factor 1: Use alpha as blend destination factor.	W
11	BSI	Blend Source Factor Inverted Source factor is inverted (meaning 1-a or 1-1 depending on BSF). 0: Use blend factor as specified through BSF 1: Invert blend source factor (1-x).	W
12	BDI	Blend Destination Factor Inverted Destination factor is inverted (meaning 1-a or 1-1 depending on BDF). 0: Use blend factor as specified through BDF 1: Invert blend destination factor (1-x).	W
13	BC2	Blend color 2 Select of blend color 2 instead of framebuffer pixel. 0: Use pixel from framebuffer as destination (DST) 1: Use color 2 as destination (DST).	W

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>14</td><td>TEXTURECLAMPX</td><td>Calculating U Limiter Outside Used Texture
This bit describes what happens when the U limiter (x direction in texture space) calculates a u value outside of the used texture.
0: Texture wrap mode: Integer part of the calculated value from the U limiter is AND gated with TEXUMASK, resulting in a repetition of texture in the x/u direction
1: Texture clamp mode: Texture color at the border of the texture is taken, resulting in a repetition of texture border color in the x/u direction.</td><td>W</td></tr>
<tr><td>15</td><td>TEXTURECLAMPY</td><td>Calculating V Limiter Outside Used Texture
This bit describes what happens when the V limiter (y direction in texture space) calculates a v value outside of the used texture.
0: Texture wrap mode: Integer part of the calculated value from the V limiter is AND gated with TEXVMASK, resulting in a repetition of texture in the y/v direction.
1: Texture clamp mode: Texture color at the border of the texture is taken, resulting in a repetition of texture border color in the y/v direction.</td><td>W</td></tr>
<tr><td>16</td><td>TEXTUREFILTERX</td><td>Linear Filtering on Texture U Axis
0: No filtering on texture U axis
1: Linear filtering on texture U axis.</td><td>W</td></tr>
<tr><td>17</td><td>TEXTUREFILTERY</td><td>Linear Filtering on Texture V Axis
0: No filtering on texture V axis
1: Linear filtering on texture V axis.</td><td>W</td></tr>
<tr><td>19:18</td><td>READFORMAT_L [1:0]</td><td>Texture Format Descriptor
Pixel format of the texture buffer.
<table>
<tr><td>b5</td><td>b4</td><td>b19</td><td>b18</td><td></td></tr>
<tr><td colspan="2">READFORMAT_H [3:2]]</td><td colspan="2">READFORMAT_L [1:0]</td><td></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>8 bpp A (8)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>16 bpp RGB (565)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>32 bpp ARGB (8888)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>16 bpp ARGB (4444)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>16 bpp ARGB (1555)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>8 bpp ACLUT (44), 4 bit alpha and 4 bit indexed color</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>8 bpp CLUT (8)/I (8), 8 bit indexed color/ luminance</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>4 bpp CLUT (4)/I (4), 4 bit indexed color/ luminance</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>2 bpp CLUT (2)/I (2), 2 bit indexed color/ luminance</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1 bpp CLUT (1)/I (1), 1 bit indexed color/ luminance.</td></tr>
<tr><td colspan="4">Others:</td><td>Setting prohibited</td></tr>
</table></td><td>W</td></tr>
<tr><td>21:20</td><td>WRITEFORMAT[1:0]</td><td>Writeback Framebuffer Format
Pixel format of the framebuffer.
<table>
<tr><td>b8</td><td>b20</td><td>b21</td><td></td></tr>
<tr><td>WRITEFORMAT2</td><td colspan="2">WRITEFORMAT[1:0]</td><td></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>8 bpp A (8)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>16 bpp RGB (565)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>32 bpp ARGB (8888)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>16 bpp ARGB (4444)</td></tr>
<tr><td colspan="3">Others:</td><td>Setting prohibited</td></tr>
</table></td><td>W</td></tr>
</table>

Bit	Symbol	Function	R/W
23:22	WRITEALPHA[1:0]	Writeback Alpha Source for Framebuffer 0 0: (USEACB = 0) Use alpha from color 2 (USEACB = 0) (USEACB = 1) BC2A = 1: Use alpha in color 2 as destination (DST_A) 0 1: (USEACB = 0) Use source alpha (pixel coverage) (USEACB = 1) BC2A = 0: Use alpha from framebuffer as destination (DST_A) 1 0: (USEACB = 0) Use 0.0 as alpha (USEACB = 1) BC2A = 0: Use alpha from framebuffer as destination (DST_A) 1 0: (USEACB = 0) Use alpha from framebuffer (USEACB = 1) BC2A = 0: Use alpha from framebuffer as destination (DST_A)	W
24	RLEENABLE	RLE Enable 0: Disable RLE 1: Enable RLE.	W
25	CLUTENABLE	CLUT Enable If CLUTENABLE = 0 (CLUT disabled), the index is directly put on the RGB channels. 0: Disable CLUT 1: Enable CLUT	W
26	COLKEYENABLE	Color Keying Enable 0: Disable color keying 1: Enable color keying.	W
27	CLUTFORMAT	CLUT Format 0: Format CLUT as ARGB (8888) 1: Format CLUT as RGB (565).	W
28	BSIA	Blend Source Factor Inverted in Alpha Channel In alpha channel blending mode (USEACB = 1): 0: Use blend factor as specified through BSFA 1: Invert blend source factor (1-x).	W
29	BDIA	Blend Destination Factor Inverted in Alpha Channel In alpha channel blending mode (USEACB = 1): 0: Use blend factor as specified through BDFA 1: Invert destination blend factor (1-x).	W
31:30	RLEPIXELWIDTH [1:0]	Texel Width for RLE Unit 0 0: 1 byte per texel 0 1: 2 bytes per texel 1 0: 3 bytes per texel 1 1: 4 bytes per texel	W

Note: S-TYPE-3, P-TYPE-3

55.2.3 IRQCTL : Interrupt Control Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0xC0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	BUSIRQCLR	BUSIRQEN	DLISTIRQCLR	ENUMIRQCLR	DLISTIRQEN	ENUMIRQEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENUMIRQEN	ENUMIRQ Interrupt Mask Enable 0: Disable (mask) ENUMIRQ enumeration interrupt 1: Enable (unmask) ENUMIRQ enumeration interrupt.	W

Bit	Symbol	Function	R/W
1	DLISTIRQEN	DLISTIRQ Interrupt Mask Enable 0: Disable (mask) DLISTIRQ display list interrupt 1: Enable (unmask) DLISTIRQ display list interrupt.	W
2	ENUMIRQCLR	Clear ENUMIRQ 0: Do not clear ENUMIRQ enumeration interrupt 1: Clear ENUMIRQ enumeration interrupt.	W
3	DLISTIRQCLR	Clear DLISTIRQ 0: Do not clear DLISTIRQ display list interrupt 1: Clear DLISTIRQ display list interrupt.	W
4	BUSIRQEN	BUSIRQ Interrupt Mask Enable 0: Disable (mask) BUSIRQ bus error interrupt 1: Enable (unmask) BUSIRQ bus error interrupt.	W
5	BUSIRQCLR	Clear BUSIRQ 0: Do not clear BUSIRQ bus error interrupt 1: Clear BUSIRQ bus error interrupt.	W
31:6	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

55.2.4 CACHECTL : Cache Control Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0xC4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	CFLUSHTX	CENABLETX	CFLUSHFX	CENABLEFX
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CENABLEFX	Framebuffer Cache Enable 0: Disable the framebuffer cache 1: Enable the framebuffer cache.	W
1	CFLUSHFX	Flush Framebuffer Cache 0: Do not flush the framebuffer cache 1: Flush the framebuffer cache.	W
2	CENABLETX	Texture Cache Enable 0: Disable the texture cache 1: Enable the texture cache.	W
3	CFLUSHTX	Flush Texture Cache 0: Do not flush the texture cache 1: Flush the texture cache.	W
31:4	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

55.2.5 STATUS : Status Control Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	BUSERRMDL	BUSERRMTXMRL	BUSERRMFB	—	BUSIRQ	DLISTIRQ	ENUMIRQ	DLISTACTIVE	CACHEDIRTY	BUSYWRITE	BUSYENUM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BUSYENUM	Enumeration Unit Status 0: Enumeration unit idle 1: Enumeration unit is busy, new primitive cannot be started.	R
1	BUSYWRITE	Framebuffer Writeback Status 0: Framebuffer writeback finished 1: Framebuffer writeback busy, framebuffer type cannot be changed.	R
2	CACHEDIRTY	Framebuffer Cache Status 0: Framebuffer cache is not dirty 1: Framebuffer cache is dirty, and frame should not be flipped.	R
3	DLISTACTIVE	Display List Reader Status 0: Display list reader is idle 1: Display list reader is busy, and no direct write access to registers allowed.	R
4	ENUMIRQ	Enumeration Interrupt Triggered 0: Enumeration not finished or interrupt disabled 1: Enumeration finished interrupt triggered.	R
5	DLISTIRQ	Display List Interrupt Triggered 0: Display list not finished or interrupt disabled 1: Display list finished interrupt triggered.	R
6	BUSIRQ	Bus Error Interrupt Triggered 0: No bus error occurred or interrupt disabled 1: Bus error interrupt triggered.	R
7	—	This bit is read as 0.	R
8	BUSERRMFB	Framebuffer Bus Error Interrupt Triggered 0: No framebuffer bus error occurred or interrupt disabled 1: Framebuffer bus error interrupt triggered.	R
9	BUSERRMTXMRL	Texture Bus Error Interrupt Triggered*1 0: No texture bus error occurred or interrupt disabled 1: Texture bus error interrupt triggered.	R
10	BUSERRMDL	Display List Bus Error Interrupt Triggered 0: No display list bus error occurred or interrupt disabled 1: Display list bus error interrupt triggered.	R
31:11	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3
Note 1. Because the RLE unit is also reading data through the texture bus, an error during RLE data access is also reflected in this bit.

55.2.6 HWREVISION : Hardware Version and Feature Set ID Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	ACBLEND	—	COLORKEY	TEXCLUT256	RLEUNIT	—	TEXCLUT	PERFCOUNT	TXCACHE	FBCACHE	DLR	—
Value after reset:	0	0	0	0	1	1	1	1	1	0	1	1	1	1	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	REV[11:0]											
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1

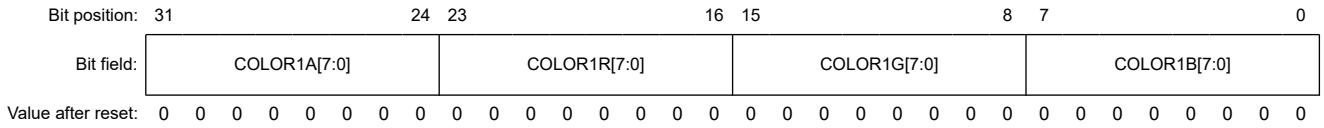
Bit	Symbol	Function	R/W
11:0	REV[11:0]	Revision Number of DRW is stored.	R
16:12	—	These bits are read as 0.	R
17	DLR	Display List Reader Available 0: No display list reader 1: Display list reader is available	R
18	FBCACHE	Framebuffer Cache Available 0: No framebuffer cache 1: Framebuffer cache is available	R
19	TXCACHE	Texture Cache Available 0: No texture cache 1: Texture cache is available	R
20	PERFCOUNT	Two Performance Counter Available 0: No performance counter 1: Tow performance counters available	R
21	TEXCLUT	Texture CLUT Available 0: No texture CLUT 1: Texture CLUT is available	R
22	—	This bit is read as 0.	R
23	RLEUNIT	RLE Unit Available 0: No RLE unit 1: RLE unit is available	R
24	TEXCLUT256	Texture CLUT size 0: Texture CLUT size is 16 entries 1: Texture CLUT size is 256 entries	R
25	COLORKEY	Color Key Available 0: No color key 1: Color key is available	R
26	—	This bit is read as 1.	R
27	ACBLEND	Alpha Channel Blending Available 0: Full alpha channel blending is not available 1: Full alpha channel blending is available	R
31:28	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

55.2.7 COLOR1 : Base Color Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0x64



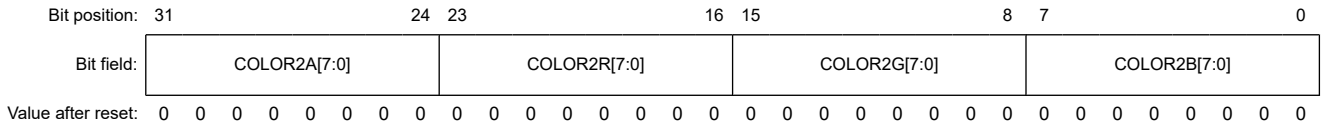
Bit	Symbol	Function	R/W
7:0	COLOR1B[7:0]	Blue Channel of Color 1 Specifies blue channel of color 1.	W
15:8	COLOR1G[7:0]	Green Channel of Color 1	W
23:16	COLOR1R[7:0]	Red Channel of Color 1	W
31:24	COLOR1A[7:0]	Alpha Channel of Color 1 Specifies alpha channel of color 1. 0x00: Transparent ⋮ 0xFF: Opaque.	W

Note: S-TYPE-3, P-TYPE-3

55.2.8 COLOR2 : Secondary Color Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0x68



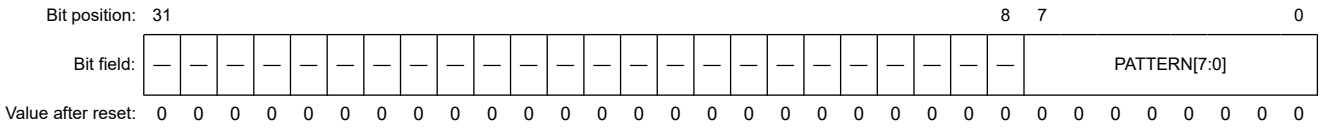
Bit	Symbol	Function	R/W
7:0	COLOR2B[7:0]	Blue Channel of Color 2 Specifies blue channel of color 2.	W
15:8	COLOR2G[7:0]	Green Channel of Color 2 Specifies green channel of color 2.	W
23:16	COLOR2R[7:0]	Red Channel of Color 2 Specifies red channel of color 2.	W
31:24	COLOR2A[7:0]	Alpha Channel of Color 2 Specifies alpha channel of color 2. 0x00: Transparent ⋮ 0xFF: Opaque.	W

Note: S-TYPE-3, P-TYPE-3

55.2.9 PATTERN : Pattern Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0x74



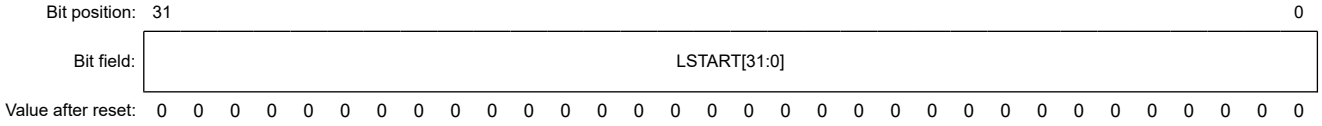
Bit	Symbol	Function	R/W
7:0	PATTERN[7:0]	Bitmap of the Pattern Specifies bitmap of the pattern.	W
31:8	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

55.2.10 LnSTART : Limiter n Start Value Register (n = 1 to 6)

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0x10 + 0x4 × (n - 1)



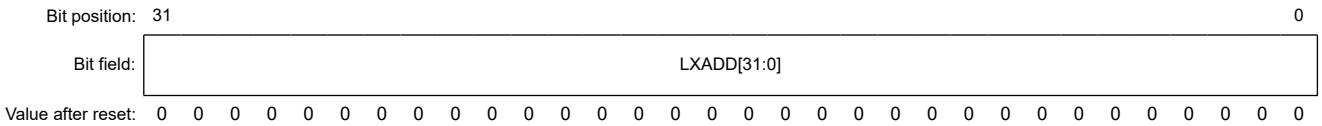
Bit	Symbol	Function	R/W
31:0	LSTART[31:0]	Start Value of the nth Limiter Specifies start value of the nth limiter.	W

Note: S-TYPE-3, P-TYPE-3

55.2.11 LnXADD : Limiter n X-Axis Increment Register (n = 1 to 6)

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0x28 + 0x4 × (n - 1)



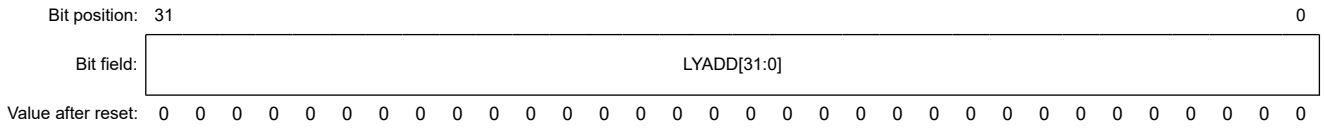
Bit	Symbol	Function	R/W
31:0	LXADD[31:0]	X-Axis Increment Specifies x-axis increment.	W

Note: S-TYPE-3, P-TYPE-3

55.2.12 LnYADD : Limiter n Y-Axis Increment Register (n = 1 to 6)

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0x40 + 0x4 × (n - 1)



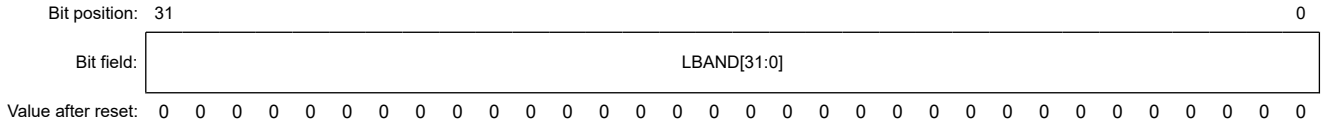
Bit	Symbol	Function	R/W
31:0	LYADD[31:0]	Y-Axis Increment Specifies y-axis increment.	W

Note: S-TYPE-3, P-TYPE-3

55.2.13 LmBAND : Limiter m Band Width Parameter Register (n = 1, 2)

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0x58 + 0x4



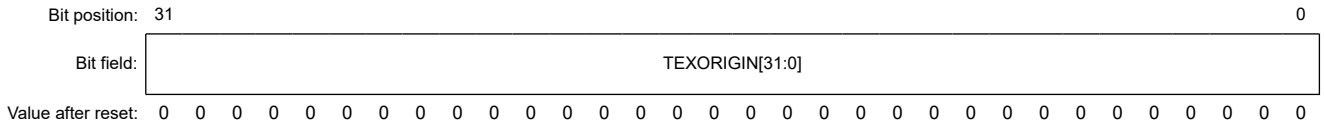
Bit	Symbol	Function	R/W
31:0	LBAND[31:0]	Limiter m Band Width Parameter Specifies limiter m band width parameter.	W

Note: S-TYPE-3, P-TYPE-3

55.2.14 TEXORIGIN : Texture Base Address Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0xBC



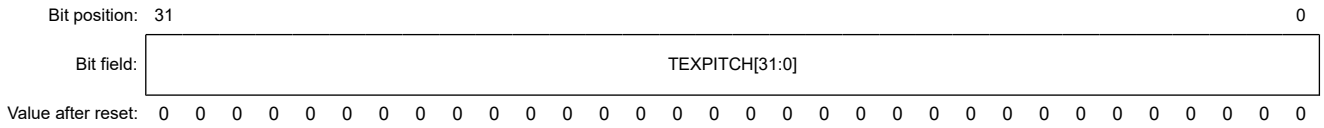
Bit	Symbol	Function	R/W
31:0	TEXORIGIN[31:0]	Texture Base Address Specifies texture base address.	W

Note: S-TYPE-3, P-TYPE-3

55.2.15 TEXPITCH : Texels Per Texture Line Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0xB4



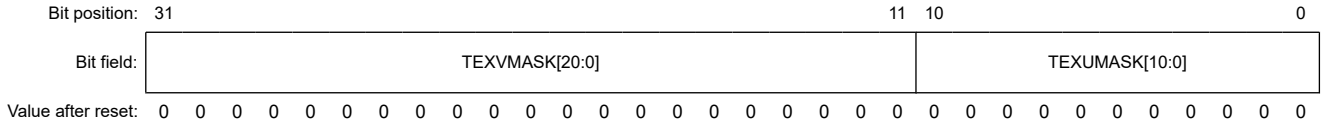
Bit	Symbol	Function	R/W
31:0	TEXPITCH[31:0]	Texels Per Texture Line Specifies texels per texture line. Valid range is from 0 to 2048.	W

Note: S-TYPE-3, P-TYPE-3

55.2.16 TEXMASK : Texture Size or Texture Address Mask Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0xB8



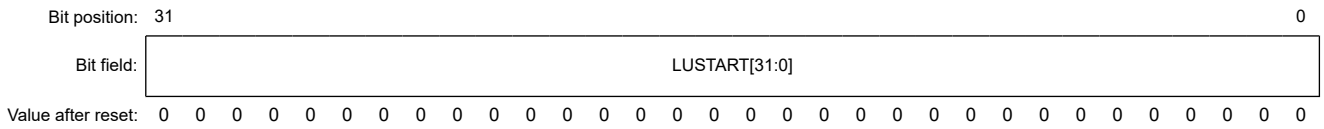
Bit	Symbol	Function	R/W
10:0	TEXUMASK[10:0]	U Mask in Texture Mode Specifies the U mask. Set to texture_width -1. ● In texture wrapping mode (CONTROL2.TEXTURECLAMPX = 0): Texture_width must be a power of 2 ● In texture clamping mode (CONTROL2.TEXTURECLAMPX = 1): All widths up to 2048 are allowed.	W
31:11	TEXVMASK[20:0]	V Mask in Texture Mode Specifies the V mask. Set to DRWTEXPITCH (texture_height - 1). ● In texture wrapping mode (CONTROL2.TEXTURECLAMPY = 0): Texture_height must be a power of 2 ● In texture clamping mode (CONTROL2.TEXTURECLAMPY = 1): All heights up to 1024 are allowed.	W

Note: S-TYPE-3, P-TYPE-3

55.2.17 LUSTART : U Limiter Start Value Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0x90



Bit	Symbol	Function	R/W
31:0	LUSTART[31:0]	U Limiter Start Value Specifies U limiter start value.	W

Note: S-TYPE-3, P-TYPE-3

55.2.18 LUXADD : U Limiter X-Axis Increment Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0x94

Bit position:	31																												0
Bit field:	LUXADD[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	LUXADD[31:0]	U Limiter X-Axis Increment Specifies U limiter x-axis increment.	W

Note: S-TYPE-3, P-TYPE-3

55.2.19 LUYADD : U Limiter Y-Axis Increment Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0x98

Bit position:	31																												0
Bit field:	LUYADD[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	LUYADD[31:0]	U Limiter Y-Axis Increment Specifies U limiter y-axis increment.	W

Note: S-TYPE-3, P-TYPE-3

55.2.20 LVSTARTI : V Limiter Start Value Integer Part Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0x9C

Bit position:	31																												0
Bit field:	LVSTARTI[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

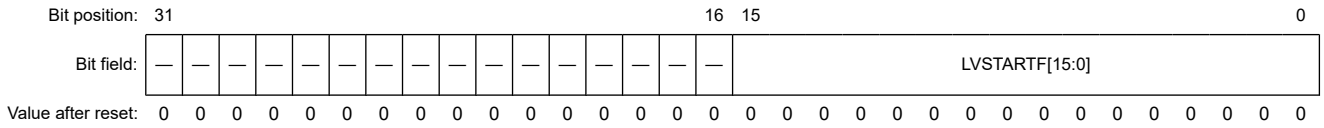
Bit	Symbol	Function	R/W
0	LVSTARTI[31:0]	V Limiter Start Value Integer Part Specifies integer part of V limiter start value.	W

Note: S-TYPE-3, P-TYPE-3

55.2.21 LVSTARTF : V Limiter Start Value Fractional Part Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0xA0



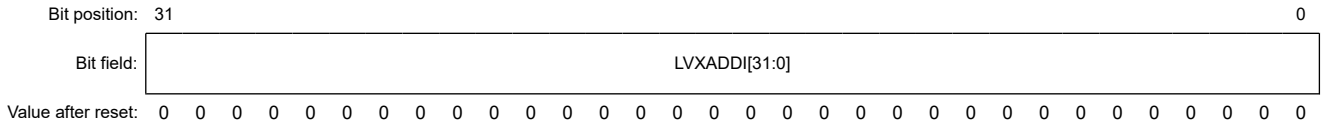
Bit	Symbol	Function	R/W
15:0	LVSTARTF[15:0]	V Limiter Start Value Fractional Part Specifies fractional part of V limiter start value.	W
31:16	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

55.2.22 LVXADDI : V Limiter X-Axis Increment Integer Part Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0xA4



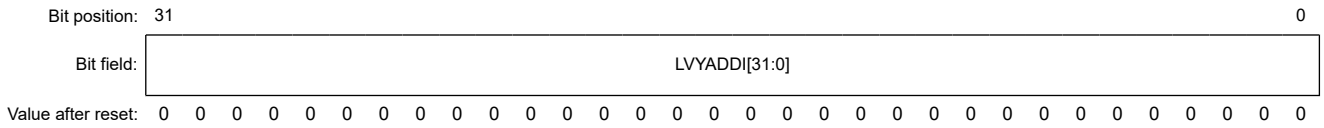
Bit	Symbol	Function	R/W
31:0	LVXADDI[31:0]	V Limiter X-Axis Increment Integer Part Specifies integer part of V limiter x-axis increment.	W

Note: S-TYPE-3, P-TYPE-3

55.2.23 LVYADDI : V Limiter Y-Axis Increment Integer Part Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0xA8



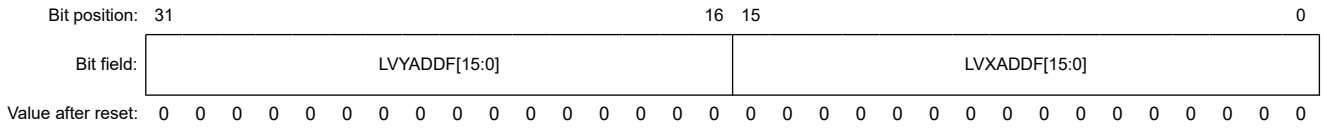
Bit	Symbol	Function	R/W
31:0	LVYADDI[31:0]	V Limiter Y-Axis Increment Integer Part Specifies integer part of V limiter y-axis increment.	W

Note: S-TYPE-3, P-TYPE-3

55.2.24 LVYXADDF : V Limiter Increment Fractional Parts Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0xAC



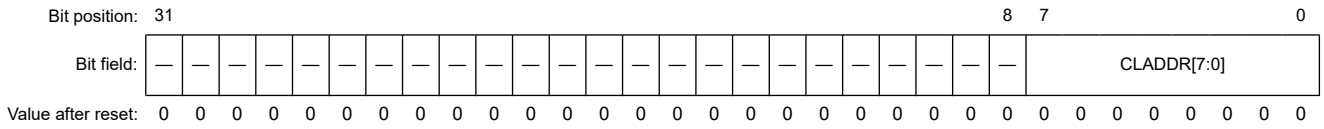
Bit	Symbol	Function	R/W
15:0	LVXADDF[15:0]	V Limiter X-Axis Increment Fractional Part Specifies fractional part of V limiter x-axis increment.	W
31:16	LVYADDF[15:0]	V Limiter Y-Axis Increment Fractional Part Specifies fractional part of V limiter y-axis increment.	W

Note: S-TYPE-3, P-TYPE-3

55.2.25 TEXCLADDR : CLUT Start Address Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0xDC



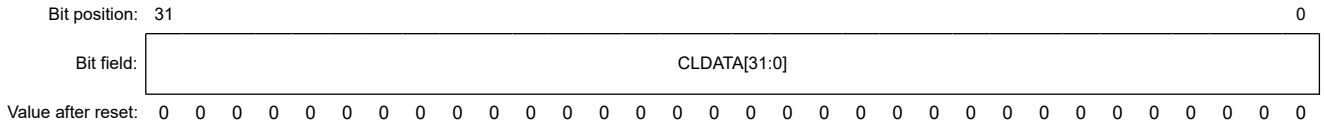
Bit	Symbol	Function	R/W
7:0	CLADDR[7:0]	Texture CLUT Start Address Specifies texture CLUT start address.	W
31:8	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

55.2.26 TEXCLDATA : CLUT Data Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0xE0



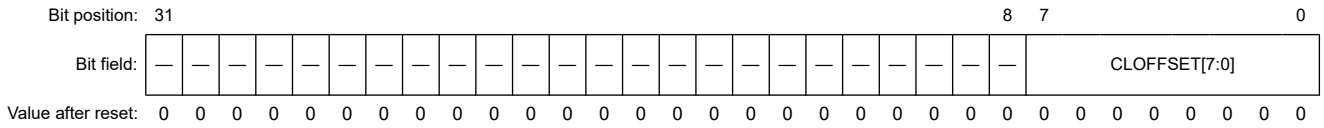
Bit	Symbol	Function	R/W
31:0	CLDATA[31:0]	Texture CLUT Data Specifies texture CLUT data.	W

Note: S-TYPE-3, P-TYPE-3

55.2.27 TEXCLOFFSET : CLUT Offset Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0xE4



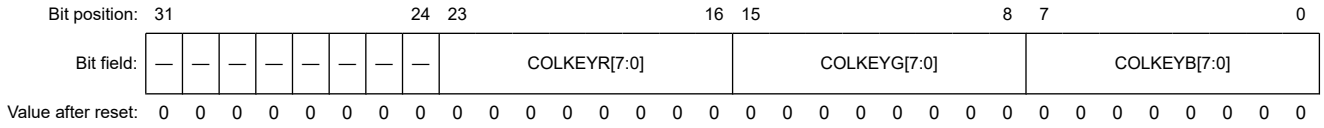
Bit	Symbol	Function	R/W
7:0	CLOFFSET[7:0]	Texture CLUT Offset Specifies texture CLUT offset. CLOFFSET[7:0] is OR gated with the original index.	W
31:8	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

55.2.28 COLKEY : Color Key Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0xE8



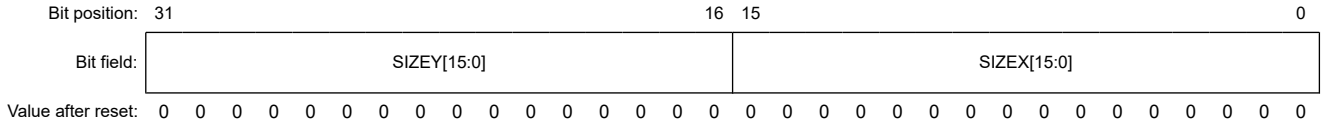
Bit	Symbol	Function	R/W
7:0	COLKEYB[7:0]	Blue Channel of Color Key Specifies blue channel of color key.	W
15:8	COLKEYG[7:0]	Green Channel of Color Key Specifies green channel of color key.	W
23:16	COLKEYR[7:0]	Red Channel of Color Key Specifies red channel of color key.	W
31:24	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

55.2.29 SIZE : Bounding Box Dimension Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0x78



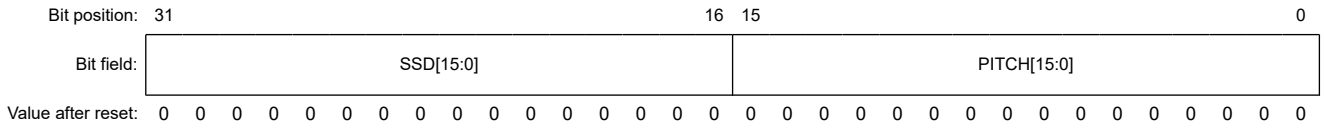
Bit	Symbol	Function	R/W
15:0	SIZEX[15:0]	Bounding Box Width Specifies the width of bounding box in pixels. Valid range is 0 to 1024.	W
31:16	SIZEY[15:0]	Bounding Box Height Specifies the height of bounding box in pixels. Valid range is 0 to 1024.	W

Note: S-TYPE-3, P-TYPE-3

55.2.30 PITCH : Framebuffer Pitch And Spanstore Delay Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0x7C



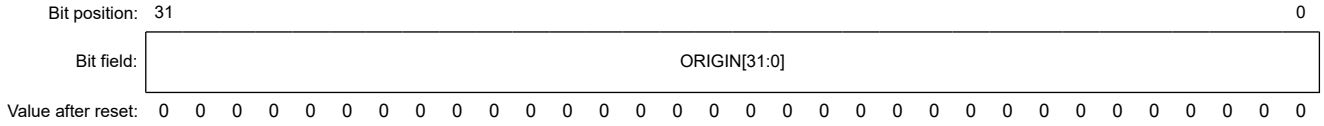
Bit	Symbol	Function	R/W
15:0	PITCH[15:0]	Pitch of the Framebuffer A negative width can be used to render bottom-up instead of top-down.	W
31:16	SSD[15:0]	Spanstore Delay Specifies the number of scan lines to delay spanstore operations.	W

Note: S-TYPE-3, P-TYPE-3

55.2.31 ORIGIN : Framebuffer Base Address Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0x80



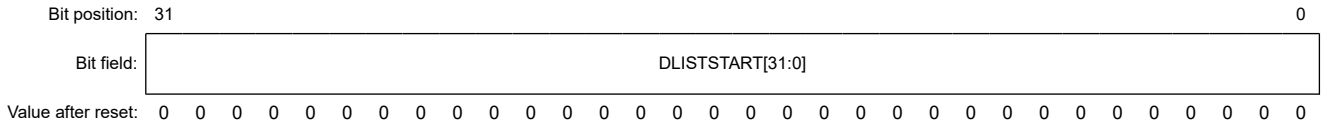
Bit	Symbol	Function	R/W
31:0	ORIGIN[31:0]	Address of the First Pixel in Framebuffer Writing to ORIGIN[31:0] triggers the start of rendering.	W

Note: S-TYPE-3, P-TYPE-3

55.2.32 DLISTSTART : Display List Start Address Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0xC8



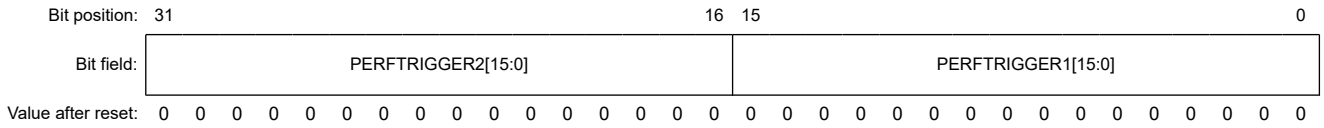
Bit	Symbol	Function	R/W
31:0	DLISTSTART[31:0]	Display List Start Address Setting a new display list base address triggers execution of the new display list. Execution stops only when a new list is set or the current list terminates.	W

Note: S-TYPE-3, P-TYPE-3

55.2.33 PERFTRIGGER : Performance Counters Control Register

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0xD4



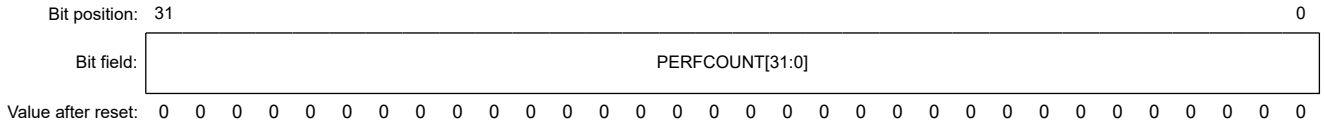
Bit	Symbol	Function	R/W
15:0	PERFTRIGGER1 [15:0]	Trigger of Performance Counter 1 Select the internal event that increments the PERFCOUNT1 register. 0x0000: Disable performance counter 0x0001: Select 2D Drawing Engine active cycles 0x0002: Select framebuffer read access 0x0003: Select framebuffer write access 0x0004: Select texture read access 0x0005: Select invisible pixels (enumerated but selected with alpha 0%) 0x0006: Select invisible pixels while internal FIFO is empty (lost cycles) 0x0007: Select display list reader active cycles 0x0008: Select framebuffer read hits 0x0009: Select framebuffer read misses 0x000A: Select framebuffer write hits 0x000B: Select framebuffer write misses 0x000C: Select texture read hits 0x000D: Select texture read misses 0x001F: Select every clock cycle (for use as timer). Others: Setting prohibited.	W
31:16	PERFTRIGGER2 [15:0]	Trigger of Performance Counter 2 Select the internal event that increments the PERFCOUNT2 register. See the above settings for performance counter 2.	W

Note: S-TYPE-3, P-TYPE-3

55.2.34 PERFCOUNTk : Performance Counter k (k = 1, 2)

Base address: DRW = 0x4034_4000
DRW_NS = 0x5034_4000

Offset address: 0xCC (PERFCOUNT1)
0xD0 (PERFCOUNT2)



Bit	Symbol	Function	R/W
31:0	PERFCOUNT[31:0]	Performance Counter k Value Specifies counter k value. The counter is reset by writing PERFCOUNTk = 0x0000_0000.	R/W

Note: S-TYPE-3, P-TYPE-3

55.3 Drawing Features

55.3.1 Drawing Features Summary

55.3.1.1 Color formats

Supported color formats are:

(1) Framebuffer formats

- 8-bit: A (8)
- 16-bit: RGB (565), ARGB (4444)
- 32-bit: ARGB (8888).

(2) Texture formats

- 1-bit: CLUT (1)/I (1)
- 2-bit: CLUT (2)/I (2)
- 4-bit: CLUT (4)/I (4)
- 8-bit: A (8), CLUT (8)/I (8), ACLUT (44)
- 16-bit: ARGB (4444), ARGB (1555), RGB (565)
- 24-bit: RGB (888) (run length encoded (RLE) unit)
- 32-bit: ARGB (8888).

CLUT formats use a 256-entry color Lookup table.

55.3.1.2 BitBLT features

The 2D Drawing Engine supports the BitBLT features using its vector drawing function to draw a rectangle and texture it based on the selected BitBLT function. This approach results in the following BitBLT features:

- Fill
- Copy
- Stretch BitBLT
- Rotate and scale
- Alpha blending
- Bilinear filtering
- Color conversion
- Subpixel exact placement.

55.3.1.3 Vector drawing features

The vector 2D Drawing Engine uses a half-plane rendering approach, which simplifies implementation of edge antialiasing and blurring features without much overhead. When combining some of its functional units, the module can draw not only linear primitives such as lines or polygons, but also quadratic equation-based primitives such as circles and ellipsoids. The following primitives are supported:

- Lines
- Polygons
- Circles and ellipses
- Quadratic curves (software driver support)

- 2D texture mapping
- Bilinear filtering of the textures.

55.3.2 Vector Drawing

For a detailed explanation of the algorithms, see section 55.6. Rendering Pipeline. Supported vector drawing includes:

(1) Lines

- Arbitrary width
- Round endpoints
- Truncated endpoints
- Alpha gradients
- Soft edges (blurring)
- Render attribute: color, pattern, or texture.

(2) Polygons

- Triangles and quadrangles (complex polygons are tessellated by software)
- Alpha gradients
- Soft edges (blurring)
- Per edge controls for anti-aliasing
- Render attribute: color, pattern, or texture.

(3) Circles and ellipses

- All conic sections
- Filled or with arbitrary width
- Arcs of 0° to 360°
- Soft edges
- Alpha gradients
- Render attribute: color, pattern, or texture.

(4) Quadratic Bézier

- Approximated by circle arcs
- Arbitrary width
- Round or truncated endpoints
- Outlines, blurring
- Alpha gradients
- Render attribute: color.

(5) Texture mapping

- 2D array of pixels that can be mapped implicitly or explicitly on all primitives provided by the 2D Drawing Engine
- Translation, rotation, and scaling/shearing
- Bilinear filtering of the textures
- 3D-like texturing accomplished with line-by-line mapping, if constant in one axis.

55.3.3 BitBLT

A dedicated BitBLT unit is not required in the 2D Drawing Engine. The rendering pipeline described for vector drawing is used as the BitBLT unit and already provides a 1 pixel/cycle throughput. For details, see section 55.6. Rendering Pipeline.

55.3.3.1 Fill

A rectangle in the framebuffer can be filled with any value. Possible color formats are 8-, 16-, or 32-bpp format. The driver optimizes the fill to gain the full benefit of 32-bit parallel rasterization. If the selected color format is less than 32 bpp, the driver corrects the alignment and fills 32 bits per clock, resulting in 2 to 4 times faster fill performance for 8- and 16-bpp formats.

55.3.3.2 Copy

A rectangle in the framebuffer can be filled with any rectangular data from the texture input. When the texture input points to the framebuffer, copying from framebuffer to framebuffer is possible. To avoid copy problems because of overlapping source and destination areas, the copy start point can be selected from top left to bottom right. Possible color formats are 8-, 16-, or 32-bpp format.

The driver optimizes the copy to gain the full benefit of 32-bit parallel rasterization. If the selected color format is less than 32 bpp, the driver corrects the alignment and copies 32 bits per clock, resulting in 2 to 4 times faster copy performance for 8- and 16-bpp formats.

55.3.3.3 Stretch BitBLT

This is similar to the normal copy operation. Because the copy is done as a type of texture mapping, the full texture mapping feature set can be used. Any scaling ratios in the x and y directions is selectable, and filtering can be enabled independently for each axis.

55.3.3.4 Rotate and scale

This is similar to the normal copy operation. Because the copy is done as a type of texture mapping, the full texture mapping feature set can be used. Any scaling ratios in the x and y direction and any rotation angle is selectable. The x and y directions can be rotated and scaled independently, and filtering for the scalers can be enabled independently for each axis.

55.3.3.5 Alpha blending

Alpha blending is a fundamental block in the rendering pipeline, so the full alpha blend feature set is available for any BitBLT operation. It is possible to copy an area and blend it over the destination by using any constant global alpha value (register value) or by using an alpha mask. The alpha mask is part of the texture data and can be either a per-pixel value together with a pixel color (ARGB formats) or an alpha-only format using a register color.

In addition to the color channels, the alpha channel can be blended. The formula for the alpha channel can be set independently from the formula for the color channels.

55.3.3.6 Bilinear filtering

The texture unit can be used to scale, rotate, or shear images. The texturing result can be filtered in the x and y directions independently. When selecting both filters, the result is a bilinear filtered texture. Using the unit twice with two independent textures would generate trilinear filtered bitmaps, improving the visual impression for high dynamic scale ratios.

55.3.3.7 Color conversion

Color conversion is required when using different texture formats than the framebuffer format. To save texture memory, several formats are supported with less bpp than the framebuffer normally has. The 2D Drawing Engine always operates internally with 32-bpp ARGB (8888). All input data is converted into 32 bpp, and is finally converted back into the framebuffer format.

55.4 Input and Output Data Formats

55.4.1 Source and Destination Data

There are two possible inputs, the framebuffer and the texture or pattern input. The output is always the framebuffer.

Every drawing operation is internally rendered in 32 bpp ARGB (8888). If the input color does not provide an alpha channel, the blue channel is taken as the alpha channel. This alpha can be substituted with any alpha (for example, by an external constant) during the colorization step in the 2D Drawing Engine.

55.4.2 Framebuffer Color Formats

Table 55.2 shows the supported framebuffer color formats.

Table 55.2 Framebuffer color formats

Framebuffer memory occupation	Format	Remarks
8 bpp	A (8)	This color format uses 1 byte per pixel. The alpha channel is internally replicated on the red, blue, and green channels and can be substituted with any color during the color step in the 2D Drawing Engine.
16 bpp	RGB (565)	This color format uses 2 bytes per pixel with 5 bits for red and blue and 6 bits for green. The blue color is taken as the alpha channel during color conversion. The alpha can be substituted with any alpha during the colorization step in the 2D Drawing Engine.
	ARGB (4444)	This color format uses 2 bytes per pixel with 4 bits for each color and alpha channel.
32 bpp	ARGB (8888)	This color format uses 4 bytes per pixel with 8 bits for each color and alpha channel.

The framebuffer color format is selected in the Surface Control Register with the CONTROL2.READFORMAT[2:0] bits.

55.4.3 Texture Color Formats

Table 55.3 shows the supported texture color formats.

Table 55.3 Texture color formats

Texture memory occupation	Format	Remarks
1 bpp	CLUT (1)/I (1)	In this mode, a 1-bit index is used to address one of 256 predefined colors in the color lookup table. If the CLUT is not used, the index is taken as a luminance value.
2 bpp	CLUT (2)/I (2)	In this mode, a 2-bit index is used to address one of 256 predefined colors in the color lookup table. If the CLUT is not used, the index is taken as a luminance value.
4 bpp	CLUT (4)/I (4)	In this mode, a 4-bit index is used to address one of 256 predefined colors in the color lookup table. If the CLUT is not used, the index is taken as a luminance value.
8 bpp	CLUT (8)/I (8)	In this mode, an 8-bit index is used to address one of 256 predefined colors in the color lookup table. If the CLUT is not used, the index is taken as a luminance value.
	A (8)	This color format uses 1 byte per pixel. The alpha channel is internally replicated on the red, blue, and green channels and can be substituted with any color during the colorization step in the 2D Drawing Engine.
	ACLUT (44)	This color format uses 1 byte per pixel. 4 bits are used as an alpha value and 4 bits are used as an index to a color palette. This approach saves space if 16 colors are sufficient to describe the image, because the next bigger alpha format would be 2-byte ARGB (4444).
16 bpp	RGB(565)	This color format uses 2 bytes per pixel with 5 bits for red and blue and 6 bits for green. The blue color is taken as the alpha channel during color conversion. The alpha can be substituted with any alpha during the colorization step in the 2D Drawing Engine.
	ARGB (4444)	This color format uses 2 bytes per pixel with 4 bits for each color and alpha channel.
	ARGB (1555)	This color format uses 2 bytes per pixel. Every color channel has 5 bits and the topmost single bit is taken as an alpha value. This can be used to hold an image with a transparency mask.
24 bpp	RGB (888)	This color format uses 3 bytes per pixel with 8 bits for each color channel. This format is only available as run length encoded data (RLE compression).
32 bpp	ARGB (8888)	This color format uses 4 bytes per pixel with 8 bits for each color and alpha channel.

The texture color format is selected in the Surface Control Register with the CONTROL2.WRITEFORMAT[3:0] bits.

55.5 Texture Data Processing

Figure 55.4 shows the processing of texture data.

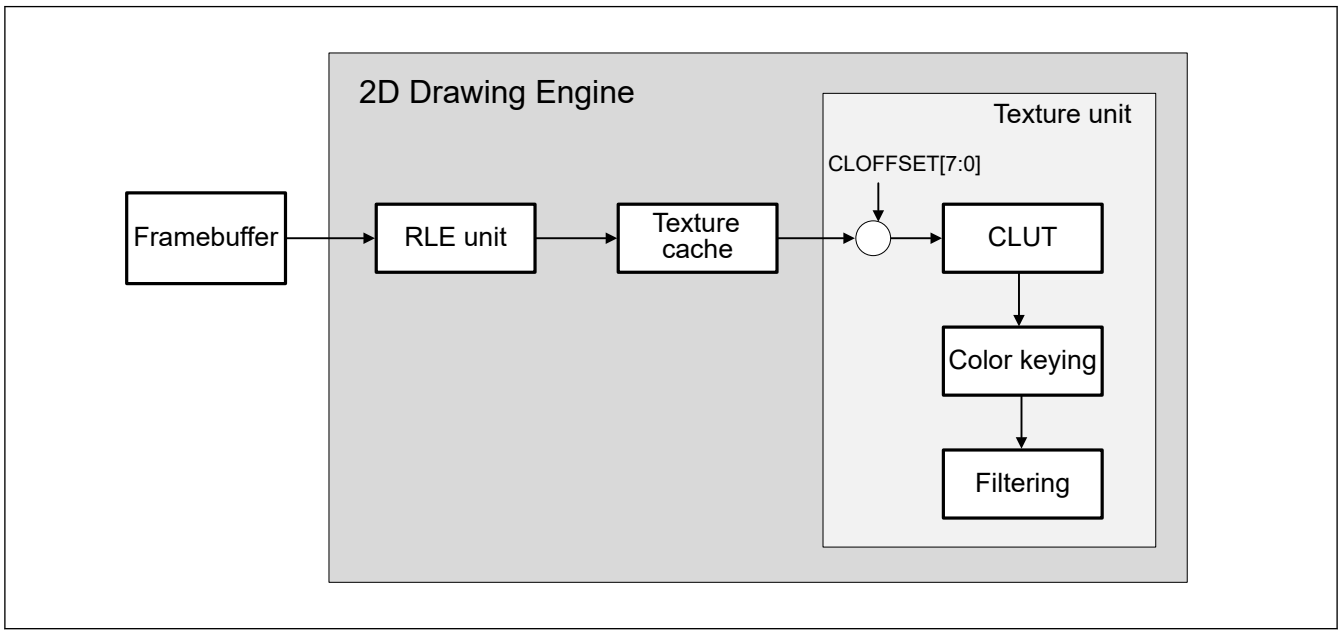


Figure 55.4 Texture data processing

55.5.1 Texture Color Format

Table 55.4 shows the supported texture data formats.

Table 55.4 Texture color formats

Texel bit width	Texel format
32 bits	ARGB (8888)
24 bits	RGB (888)
16 bits	ARGB (4444), ARGB (1555), RGB (565)
8 bits	CLUT (8)I (8), A (8), ACLUT (44)
4 bits	CLUT (4)/I (4)
2 bits	CLUT (2)/I (2)
1 bit	CLUT (1)/I (1)

55.5.2 Run Length Encoded (RLE) Unit

The RLE unit decompresses Targa-like compressed textures and hands the decompressed texel data over to the texture unit. The key features are:

- Support for Targa format
- Avoid the additional Targa limitation to scan lines
- Support for clipping of compressed images, in which the 2D Drawing Engine is allowed to copy only a portion of a larger original texture
- Control of the RLE unit in drawing list operation
- Bypassing of the RLE unit logic if uncompressed textures are fetched from the framebuffer.

(1) Texture cache

The RLE unit feeds the texture cache. The texture cache can be disabled by setting CACHECTL.CENABLETX = 0.

Caution: A texture cache flush operation (CACHECTL.CFLUSHTX = 1) is necessary at the beginning and end of every new RLE texture.

The texture cache and the RLE unit can be bypassed by setting CONTROL2.RLEENABLE = 0.

55.5.2.1 RLE Texel formats

Table 55.5 lists the data formats supported by the RLE unit.

Table 55.5 Texel formats supported by the RLE unit

Memory texel format	RLE parameters	RLE coded unit format (CONTROL2.RLEPIXELWIDTH [1:0])	Delivered format
32-bit ARGB (8888)	Included in Targa and RLE formats	32 bits*1 (11b)	32 bits
24-bit RGB (888)		24 bits (10b)	32 bits
16-bit ARGB (4444), ARGB (1555), RGB (565)		16 bits (01b)	16 bits
8-bit CLUT (8)I (8), A (8), ACLUT (44)		8 bits (00b)	8 bits
4-bit CLUT (4)/I (4)	Optional for RLE*2	8 = 2 x 4 bits	4 + 4 bits
2-bit CLUT (2)/I (2)	No RLE		
1-bit CLUT (1)/I (1)			

Note 1. 24-bit RGB (888) encoded texels are delivered as ARGB (8888) with Alpha set to 1.
Note 2. Encoding of textures with 4 bits per texel is not defined by the Targa specification but can be done by:

- Combining two 4-bit texels to one byte
- Padding with four 0-bit at the end of the file, if the number of texels is odd
- Encoding as with 8-bit texels.

(1) Texel addressing for RLE textures

The address of a texel is the byte address of the first byte of the texel. The origin of the texture is given by the register TEXORIGIN.

Note: The RLE code must begin at a word boundary of the memory.

Caution: When the FIFO is filled, there is no provision to inhibit read access beyond the end of the RLE code. To avoid memory access violations, the RLE code must be padded by 32 memory words, where every bit of each word is set to 1.

55.5.2.2 Targa RLE format

Run-length encoded (RLE) images include two types of data elements:

- Run-length packets
- Raw packets.

The first field (1 byte) of each packet is called the repetition count field. The second field is called the pixel value field (1, 2, 3. or 4 bytes). For run-length packets, the pixel value field contains a single pixel value. For raw packets, the field is a variable number of pixel values.

The highest order bit of the repetition count indicates whether the packet is a raw packet or a run-length packet, as follows:

- If bit [7] of the repetition count is set to 1, the packet is a run-length packet
- If bit [7] of the repetition count is set to zero, the packet is a raw packet.

The lower 7 bits of the repetition count specify how many pixel values are represented by the packet. For a run-length packet, this count indicates how many successive pixels have the pixel value specified in the pixel value field. For raw packets, this count specifies how many pixel values are actually contained in the next field. This 7-bit value is actually encoded as 1 less than the number of pixels in the packet (a value of 0 implies 1 pixel while a value of 0x7F implies 128 pixels).

(1) Run-length packet

Run-length packets are composed of two parts. The first is a repetition count and the second is the pixel value to repeat.

Table 55.6 Run-length packet

Field name	Field size
Packet type (must be 1 for run-length)	1 bit
Pixel count (number of pixels encoded in this packet - 1)	7 bits
Pixel data (the shared pixel value to be used)	Pixel depth (field 5.5)

(2) Raw packet

The raw packet always includes two fields. The first field is the repetition count and the second field is the pixel data field.

Table 55.7 Raw packet

Field name	Field size
Packet type (must be 0 for raw packet)	1 bit
Pixel count (number of pixels encoded by this packet - 1)	7 bits
Pixel data	Pixel depth x pixel count - 1

55.5.3 Color Lookup Table (CLUT)

The color lookup table receives an index that addresses one out of the 256 predefined colors.

The predefined color format can be selected as:

- CONTROL2.CLUTFORMAT = 0: ARGB (8888)
- CONTROL2.CLUTFORMAT = 1: RGB (565).

The CLUT is filled by the use of two registers:

- TEXCLDATA
  The ARGB (8888) color definition is written to this register, while the CLUT address is taken from TEXCLADDR.
- TEXCLADDR.
  This is set to the first address of the CLUT to write to and is automatically incremented after each write to TEXCLDATA.

An offset for indexed formats (CLUT (1), CLUT (2), CLUT (4), and CLUT (8)) can be set up in the TEXCLOFFSET register to allow selecting an offset part of the CLUT. The CLUT index is calculated by CLUT (x) or TEXCLOFFSET.CLOFFSET[7:0].

55.5.3.1 CLUT/I pixel data formats

(1) CLUT (1)/I (1) format

The CLUT (1)/I (1) format expresses 1 pixel by using a total of 1 bit.

Memory byte	Pixel
7 (MSB)	P7
6	P6
5	P5
4	P4
3	P3
2	P2
1	P1
0 (LSB)	P0

The left-most pixel is stored at the lowest bit of the memory byte.

(2) CLUT (2)/I (2) format

The CLUT (2)/I (2) format expresses 1 pixel by using a total of 2 bits.

Memory byte	Pixel
7 (MSB)	P3
6	
5	P2
4	
3	P1
2	
1	P0
0 (LSB)	

The left-most pixel is stored at lowest 2 bits of the memory byte.

(3) CLUT (4)/I (4) format

The CLUT (4)/I (4) format expresses 1 pixel by using a total of 4 bits.

Memory byte	Pixel
7 (MSB)	P1
6	
5	
4	
3	P0
2	
1	
0 (LSB)	

The left-most pixel is stored at lowest 4 bits of the memory byte.

55.5.4 Color Keying

The 2D Drawing Engine provides a color keying unit in front of the texture unit. It operates as follows:

1. If enabled, the incoming color is compared with a transparent color, defined by COLKEY.
2. If the value matches, the alpha and color values are set to 0 to mark the color as transparent and handle it as if alpha was pre-multiplied.
3. If the value does not match, then alpha is set to 1.
4. Additional operations such as alpha_in × alpha_const are still possible.

With this approach, an object such as a round icon can be cut out from a rectangular texture and still can be faded by a constant alpha over the background.

55.6 Rendering Pipeline

55.6.1 Coordinate Transformation

Coordinate transformation such as rotation, translation, projection, and scaling must be done on the application side. This is not part of the 2D Drawing Engine hardware or driver. Because all coordinates fed into the 2D Drawing Engine are in fixed point format, these calculations can be made in fixed point format and do not require a floating point unit.

55.6.2 Rasterization

During rasterization, the vector data of the object must be converted to pixel data. To convert the data, the program sets up the edge interpolation hardware, called a limiter, for each edge of the object that calculates a decision value. The limiter determines which side of the edge the pixel is positioned on. The 2D Drawing Engine includes six internal hardware limiters. In principle, the limiter registers contain the distance between the pixel being processed and the edge.

In the linear setup, a limiter describes a half plane. The intersection of all half planes is the object. If three half planes intersect, a triangle is created as shown in Figure 55.5.

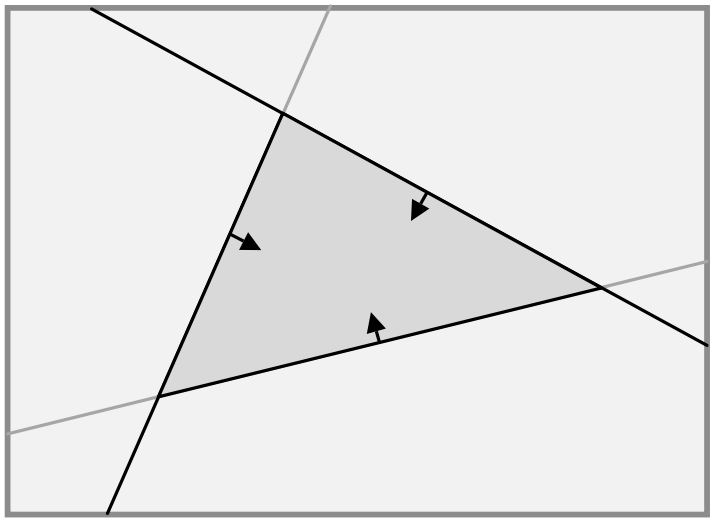


Figure 55.5 Intersection of half planes

The limiter output is clamped to an interval of [0:1]. In limiters 1 and 2 it is possible to apply a band filter before the clamping operation. In this case, the limiter is not describing a half plane but a small band. With this approach, a single limiter can describe a thick line of infinite length.

The output of the different limiters can be combined by the combiner units with a maximum or minimum operation. Maximum operation describes the union of both half planes, and minimum operation describes the intersection of both half planes. The final output is then used as an alpha value. Edge anti-aliasing can be done with no additional effort with this hardware.

To calculate the decision value for each possible pixel with a limiter, the bounding box of the object must be calculated. Then, the decision value for the top left corner of the bounding box must be calculated for each edge. Finally, the increments for a step in the x direction and a step in the y direction must be calculated. This is done by the CPU in the driver.

With this information, the 2D Drawing Engine scans the whole bounding box and calculates the decision value for every pixel incrementally. For a block diagram of the entire rasterization unit, see section 55.1. Overview.

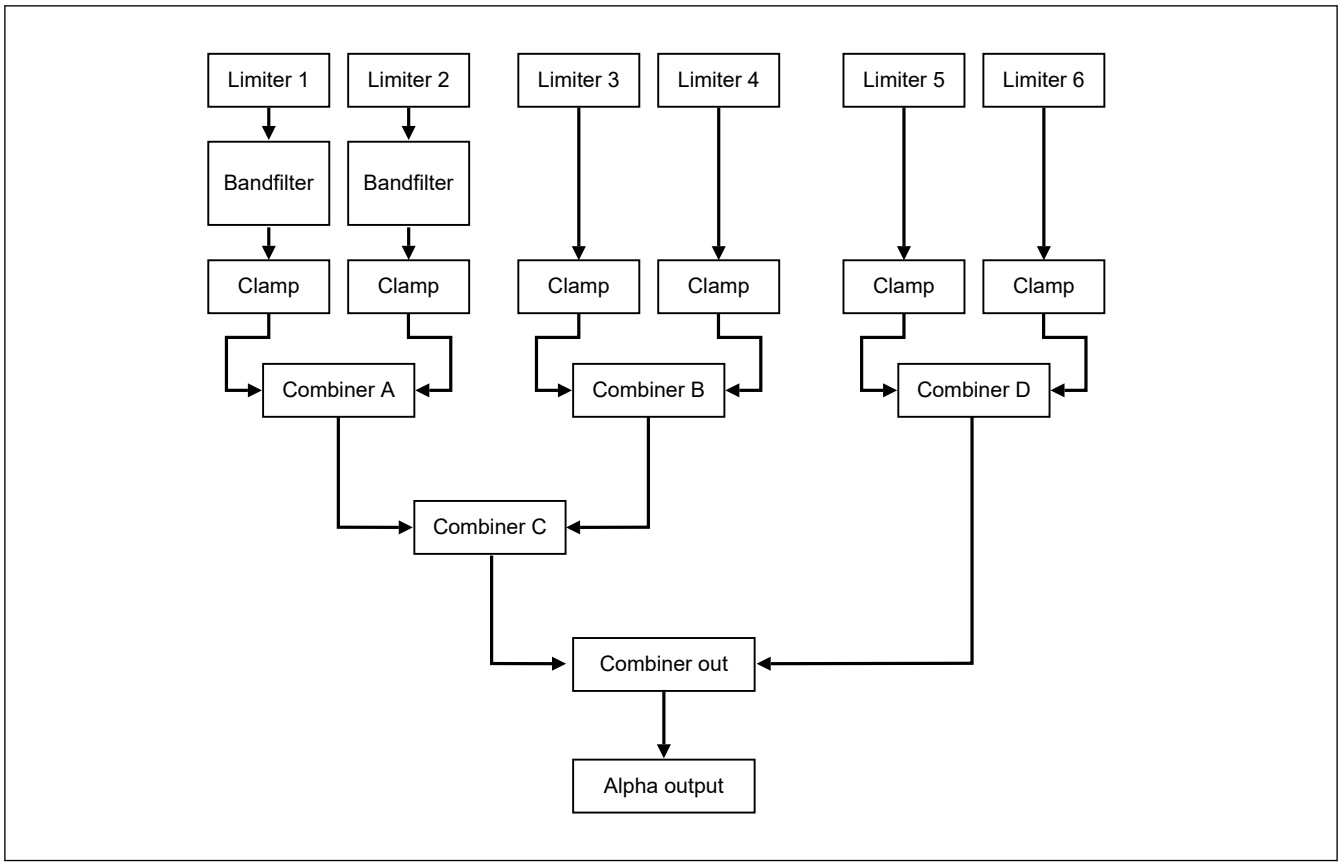


Figure 55.6 Block diagram of rasterization unit

The limiters calculate distances and the combiner units combine them to an alpha value. The combiner units define the conditions for whether or not a pixel is in the bounding box. The alpha value must be greater than 0.

Note: It is possible to have all limiters switched off.

55.6.2.1 Edge setup linear case

(1) Mathematical background

To setup a linear edge, consider the line equation in the classical form.

This can be written as:

$y = \widetilde{a}x + \widetilde{b}$

This can be rewritten as:

$0 = f(x,y) = \widetilde{a}x - y + \widetilde{b} = \mathrm{ax} + \mathrm{by} + c$

with $a = \widetilde{a}, b = -1, c = \widetilde{b}$

This is a more general form. Consider a vector form of this equation:

$$\vec{p} = \begin{pmatrix} x \\ y \end{pmatrix}, \vec{n} = \begin{pmatrix} a \\ b \end{pmatrix} \Rightarrow f(x,y) = \mathrm{ax} + \mathrm{by} + c = \vec{p} \cdot \vec{n} + c$$

If a point $\vec{p}_0$ is on the line:

$0 = f(x,y) = \vec{p}_0 \cdot \vec{n} + c \Rightarrow c = -\vec{p}_0 \cdot \vec{n}$

Rewriting the constant, the equation becomes:

$c = -\vec{p}_0 \cdot \vec{n} \Rightarrow f(x,y) = \left(\vec{p} - \vec{p}_0\right) \cdot \vec{n}$

The vector $\vec{n}$ is called the normal vector and is perpendicular to the line. The setup can be seen in Figure 55.7.

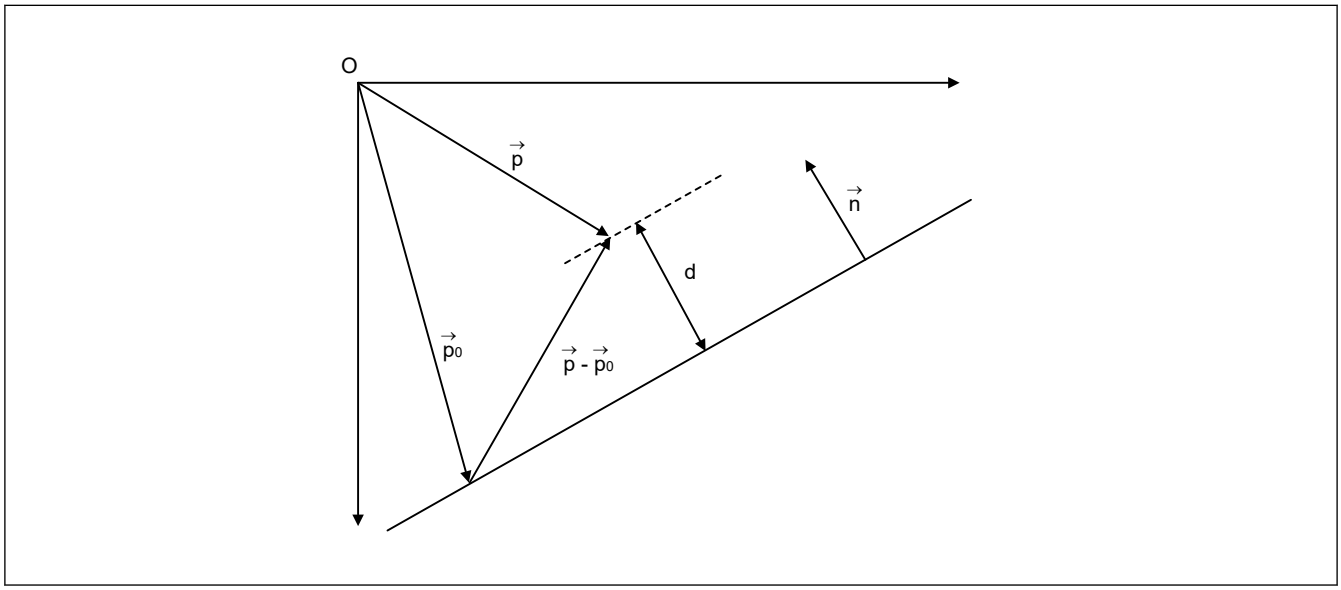


Figure 55.7 Distance of a point to a line

The projection of $\vec{p} - \vec{p}_0$ to $\vec{n}$ is the distance d of the point P to the line. In this case, f (x, y) describes the distance to the line of the pixel with coordinates (x, y).

To calculate the distance of every pixel of the bounding box incrementally, first the distance to the line at origin must be calculated:

$f(0,0) = -\vec{p}_0 \cdot \vec{n} = c$

Next the increments for a step in the x direction and a step in the y direction must be calculated:

$f\left(\vec{p} + \vec{e}_x\right) = \left(\vec{p} + \vec{e}_x - \vec{p}_0\right) \cdot \vec{n} = f\left(\vec{p}\right) + \vec{e}_x \cdot \vec{n} = f\left(\vec{p}\right) + a$

$f\left(\vec{p} + \vec{e}_y\right) = \left(\vec{p} + \vec{e}_y - \vec{p}_0\right) \cdot \vec{n} = f\left(\vec{p}\right) + \vec{e}_y \cdot \vec{n} = f\left(\vec{p}\right) + b$

with $\vec{e}_x = \begin{pmatrix}1\\0\end{pmatrix}$, $\vec{e}_y = \begin{pmatrix}0\\1\end{pmatrix}$

Consequently, the new distance can be calculated from the old distance with the increments a and b. A step in the x direction changes the distance by a, and a step in the y direction changes the distance by b. The distance of the origin to the line is c.

With this information, the entire bounding box can be scanned. If the bounding box top left corner is not at the origin, an offset must be added.

(2) Limiter operation

The 2D Drawing Engine contains six limiters. Each limiter contains three registers:

- LnSTART
- LnXADD
- LnYADD.

See Figure 55.8 for details. It is possible to drive the limiters in a threshold mode, in which all values above 0.5 are set to 1, and all values below or equal to 0.5 are set to 0. This feature is used when anti-aliasing is not wanted.

Note: In Figure 55.8, the following abbreviations are used:

start = LnSTART
xadd = LnXADD
yadd = LnYADD

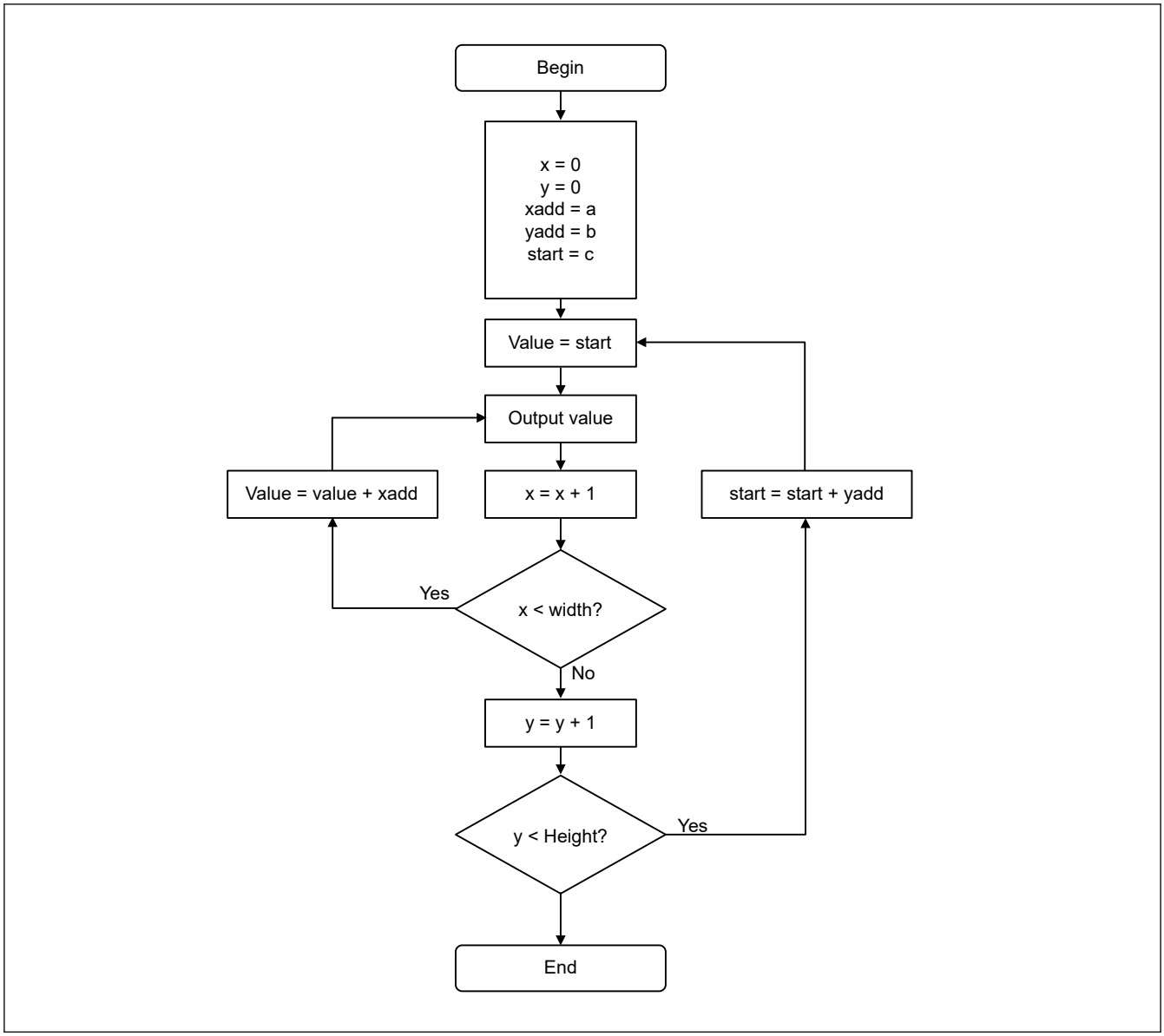


Figure 55.8 Operation flow of the linear limiter

(3) Example

If a straight line is given by the points P0 and P1, then the values are calculated as follows:

$$\Delta\vec{p} = \vec{p}_1 - \vec{p}_0 = \begin{pmatrix} x_1 - x_0 \\ y_1 - y_0 \end{pmatrix} = \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

The normal vector (perpendicular but not unit size) is then:

$$\vec{n} = \begin{pmatrix} -\Delta y \\ \Delta x \end{pmatrix}$$

The not normalized distance between edge and origin is then:

$$\vec{p}_0 \cdot \vec{n} = -x_0\Delta y + y_0\Delta x$$

The limiter parameters would be:

$\text{start} = -x_0\Delta y + y_0\Delta x$

$\text{xadd} = -\Delta y$

$\text{yadd} = \Delta x$

In the normalized case, the normal vector is:

$$\vec{n} = \begin{pmatrix} -\Delta y \\ \Delta x \end{pmatrix} \cdot \frac{1}{\sqrt{\Delta x^2 + \Delta y^2}}$$

The distance between edge and origin is:

$$\vec{p}_0 \cdot \vec{n} = \left(-x_0 \cdot \Delta y + y_0 \cdot \Delta x\right) \cdot \frac{1}{\sqrt{\Delta x^2 + \Delta y^2}}$$

The limiter parameters would be:

$$\text{start} = \left(-x_0\Delta y + y_0\Delta x\right) \cdot \frac{1}{\sqrt{\Delta x^2 + \Delta y^2}}$$

$$\text{xadd} = -\Delta y \cdot \frac{1}{\sqrt{\Delta x^2 + \Delta y^2}}$$

$$\text{yadd} = \Delta x \cdot \frac{1}{\sqrt{\Delta x^2 + \Delta y^2}}$$

Normalization is only required if anti-aliasing is used. The driver contains an optimized inverse square root function to speed up the normalization process.

55.6.2.2 Edge setup quadratic case

(1) Mathematical background

It is also possible to set up the limiters to incrementally calculate the following equation:

$f(x,y) = ax^2 + by^2 + cx + dy + f$

At the origin, the value is:

$f(0,0) = f$

The step in the x direction is:

$f(x+1,y)$

$= a(x+1)^2 + by^2 + c(x+1) + dy + f$

$= ax^2 + 2ax + a + by^2 + cx + c + dy + f$

$= f(x,y) + 2ax + c + a$

$dx(x) = f(x+1,y) - f(x,y) = 2ax + c + a$

The step in the y direction is:

$f(x,y+1)$

$= ax^2 + b(y+1)^2 + cx + d(y+1) + f$

$= ax^2 + by^2 + 2by + b + cx + d(y+1) + f$

$= f(x,y) + 2by + d + b$

$dy(y) = f(x,y+1) - f(x,y) = 2by + d + b$

In the quadratic case, the increments dx and dy depend on x and y and are not constant. They can be calculated incrementally:

$d^2x = dx(x+1) - dx(x) = 2a(x+1) + c + a - (2ax + c + a) = 2a$

$d^2y = dy(y+1) - dy(y) = 2b(y+1) + d + b - (2by + d + b) = 2b$

At the origin, the increments are:

$\mathrm{dx}(0) = c + a$ and $\mathrm{dy}(0) = d + b$

By incrementing the value by dx and dy for every step in the x and y direction and incrementing dx, dy by d^2x, and d^2y for every step in the x and y direction, the quadratic equation can be easily calculated for the whole bounding box.

(2) Limiter operation

In the quadratic case, two linear limiters are combined to operate as one quadratic limiter, called limiter 1 and limiter 2. The registers are:

- L1START, L1XADD, L1YADD
- L2START, L2XADD, L2YADD.

See Figure 55.9 for details. The gray box is an addition that performs a different operation, as in the linear setup. It is possible to drive the limiters in a threshold mode, in which all values above 0.5 are set to 1, and all values below or equal to 0.5 are set to 0. This feature is used when anti-aliasing is not needed.

Note: In Figure 55.9, the following abbreviations are used:

- start1 = L1START, start2 = L2START
- xadd1 = L1XADD, xadd2 = L2XADD
- yadd1 = L1YADD, yadd2 = L2YADD.

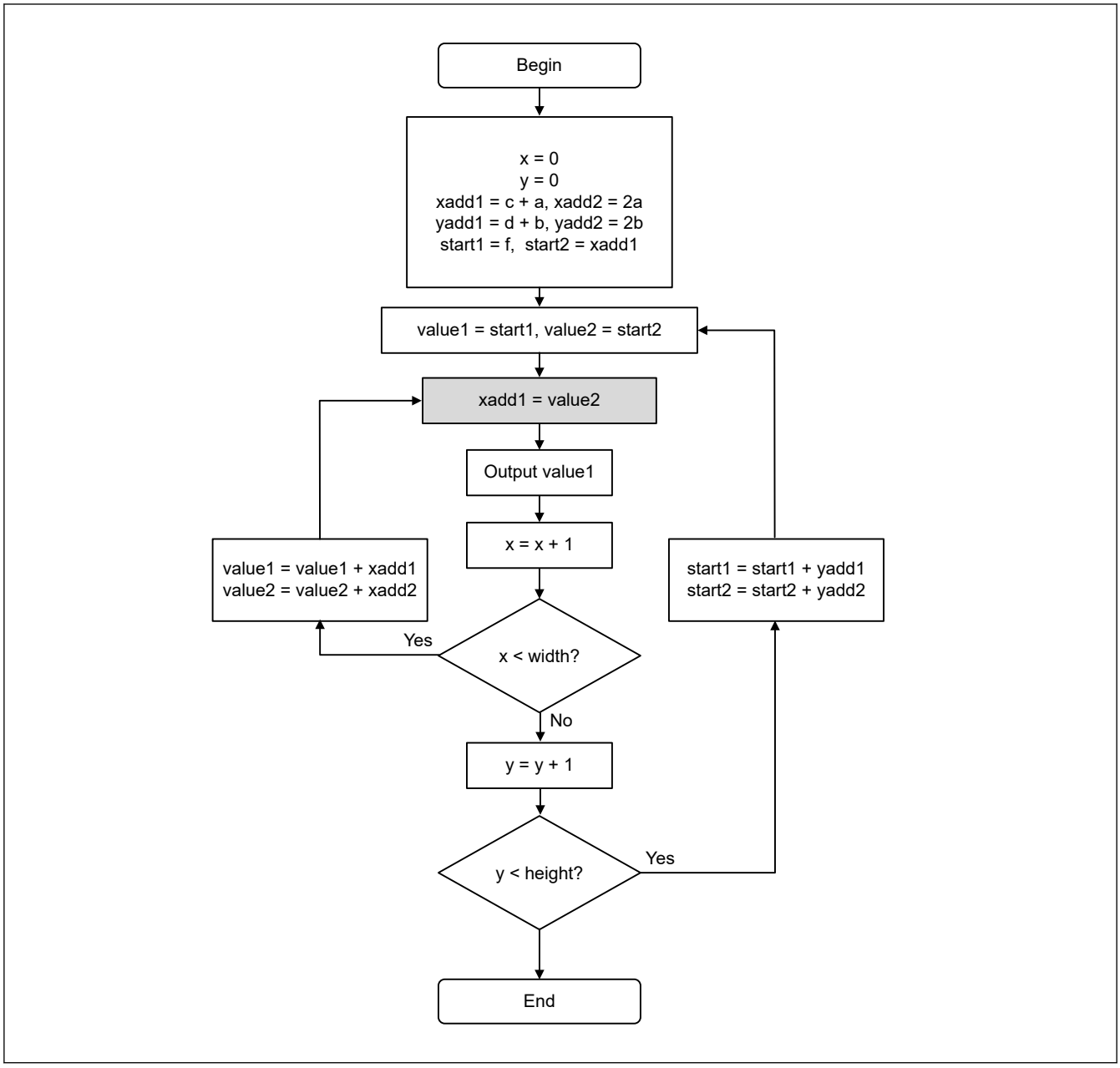


Figure 55.9 Operation flow of the quadratic limiter

(3) Example

Consider the equation for a circle with the center at $\vec{c} = \begin{pmatrix} s \\ t \end{pmatrix}$ and radius r:

$0 = f(x,y) = (x-s)^2 + (y-t)^2 - r^2$

This equation can be rewritten as:

$f(x,y) = x^2 - 2xs + s^2 + y^2 - 2yt + t^2 - r^2$

This can be sorted to fit to the original equation:

$f(x,y) = x^2 + y^2 - 2sx - 2ty + \left(s^2 + t^2 - r^2\right)$

With the following assignments, the circle equation can be calculated incrementally:

$a = 1$

$b = 1$

$c = -2s$

$d = -2t$

$f = s^2 + t^2 - r^2$

For the limiters with the results calculated in (1) Mathematical background, this would result in:

$\text{start1} = f = s^2 + t^2 - r^2$

$\text{xadd1} = c + a = -2s + 1$

$\text{yadd1} = d + b = -2t + 1$

$\text{start2} = \text{xadd1}$

$\text{xadd2} = 2a = 2$

$\text{yadd2} = 2b = 2$

55.6.2.3 Band filter

The output of limiter 1 and 2 can be modified to use a band filter. The band filter has a single filter parameter w.

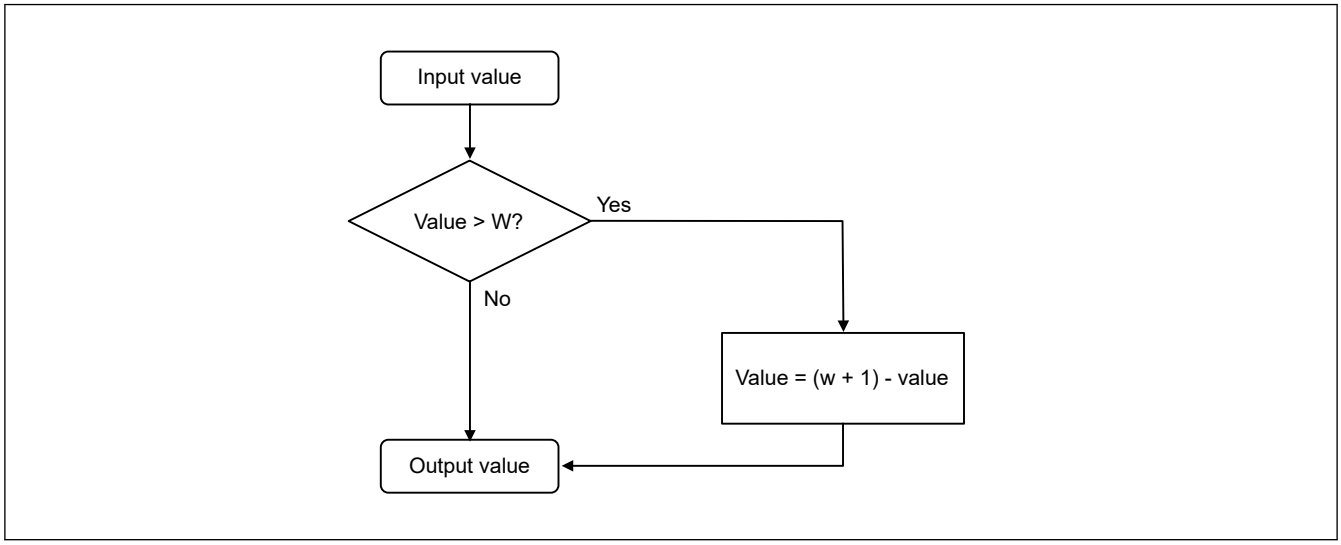


Figure 55.10 Band filter

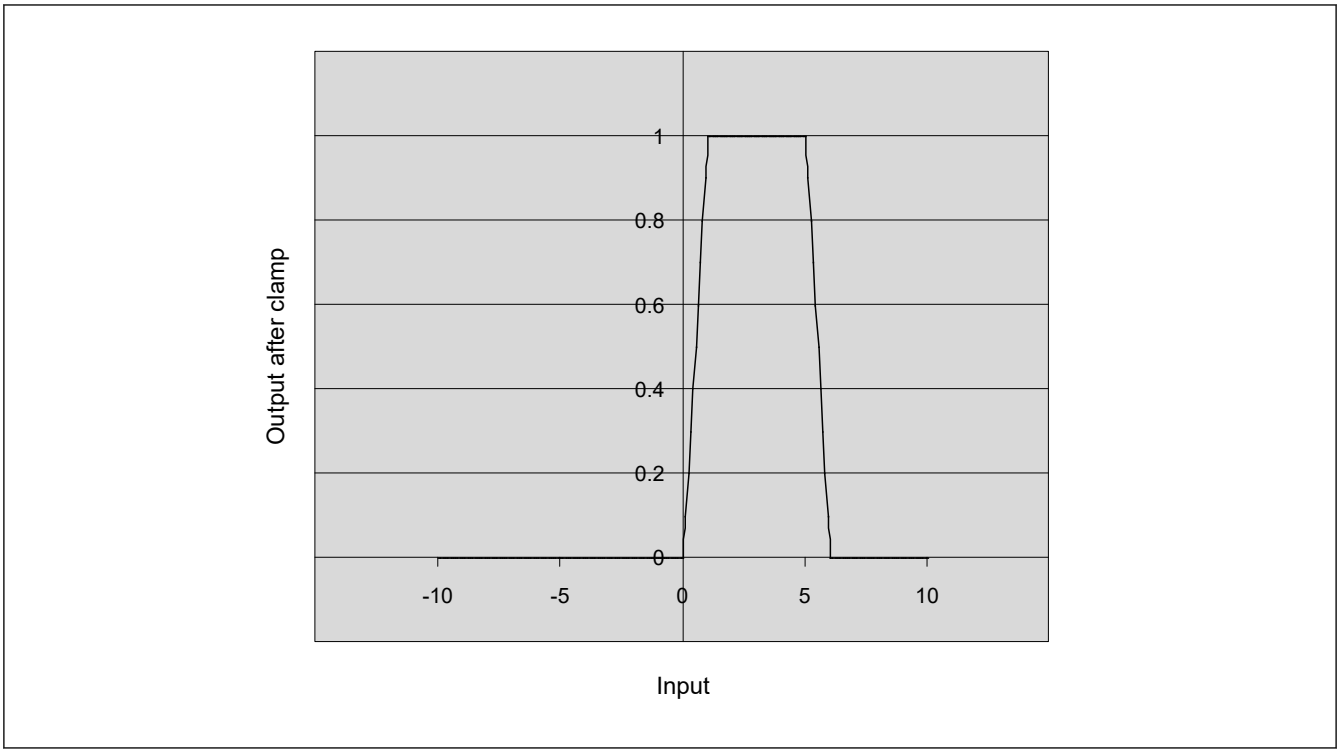


Figure 55.11 Band filter output after clamp with w = 7

55.6.2.4 Clamping unit

The clamping unit cuts the limiter output to the interval [0:1].

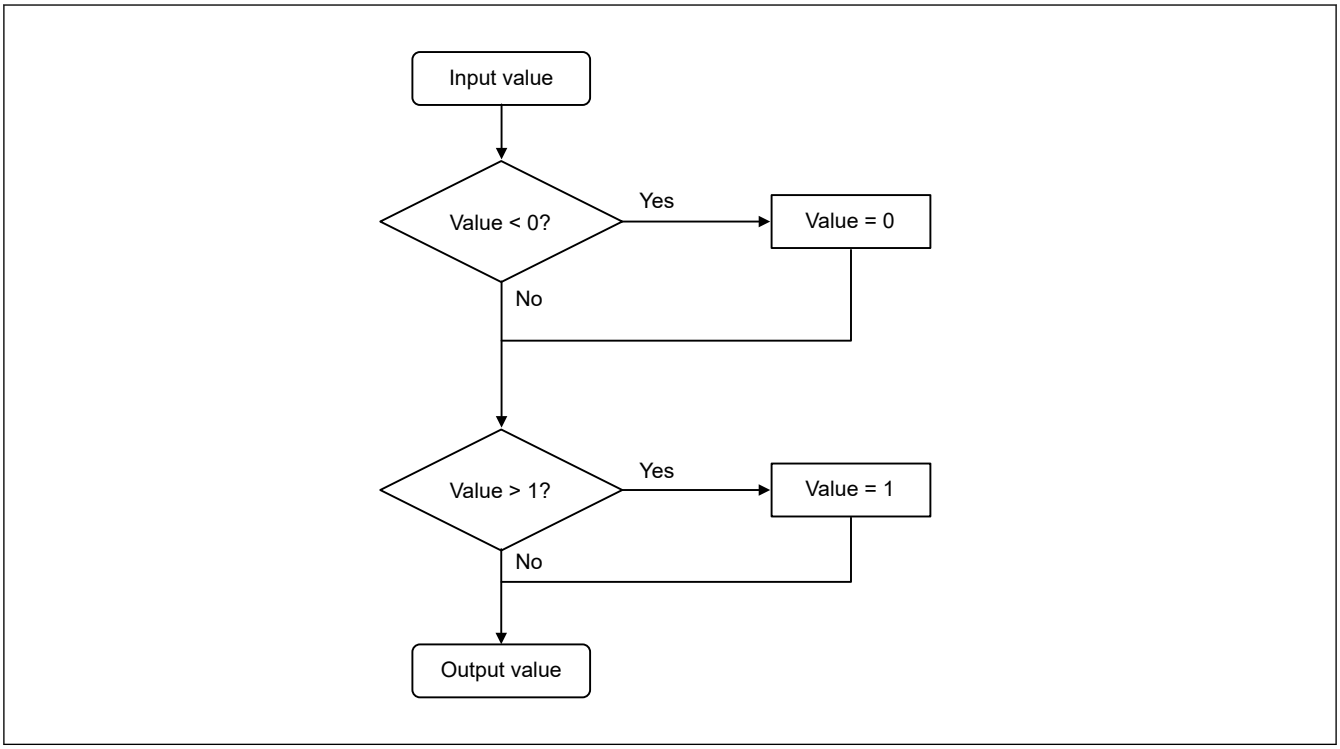


Figure 55.12 Clamping unit

The clamping unit can be put into threshold mode, in which all values greater than 0.5 are set to 1, and all values below or equal to 0.5 are set to 0. This feature is used when anti-aliasing is not needed, such as for shared edges.

55.6.2.5 Combiner unit

The combiner unit can be operated in minimum mode and in maximum mode. In minimum mode the smaller value is output, and in maximum mode the larger value is output. The minimum mode represents the intersection, and the maximum mode represents the union of the two regions.

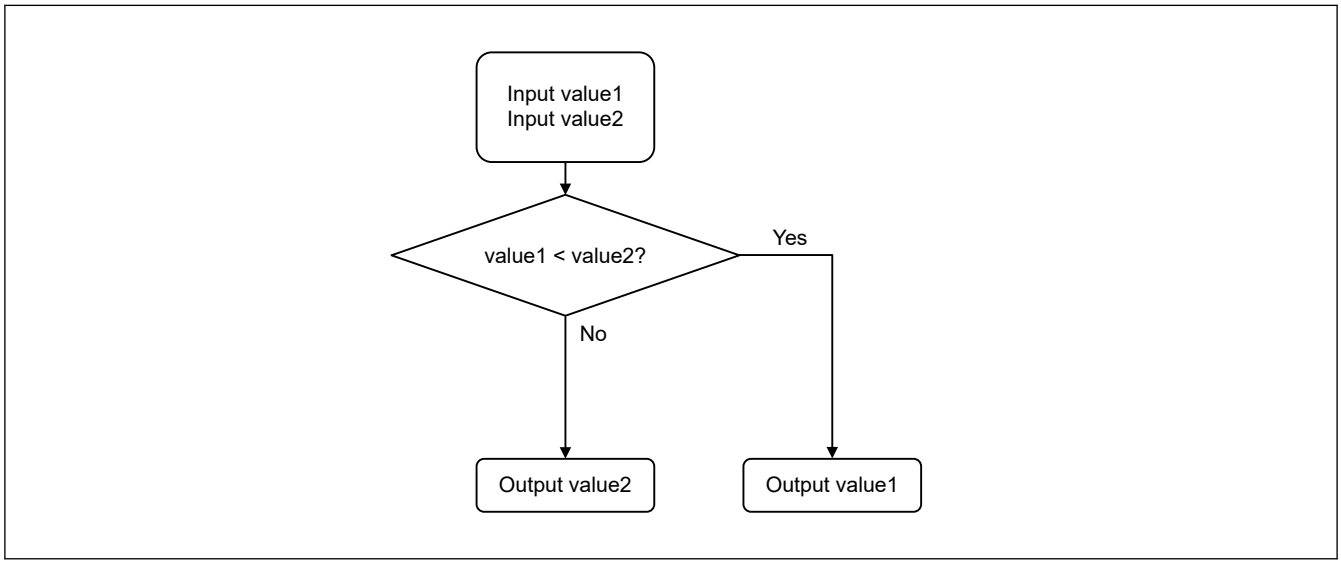


Figure 55.13 Combiner operated in minimum mode with intersection

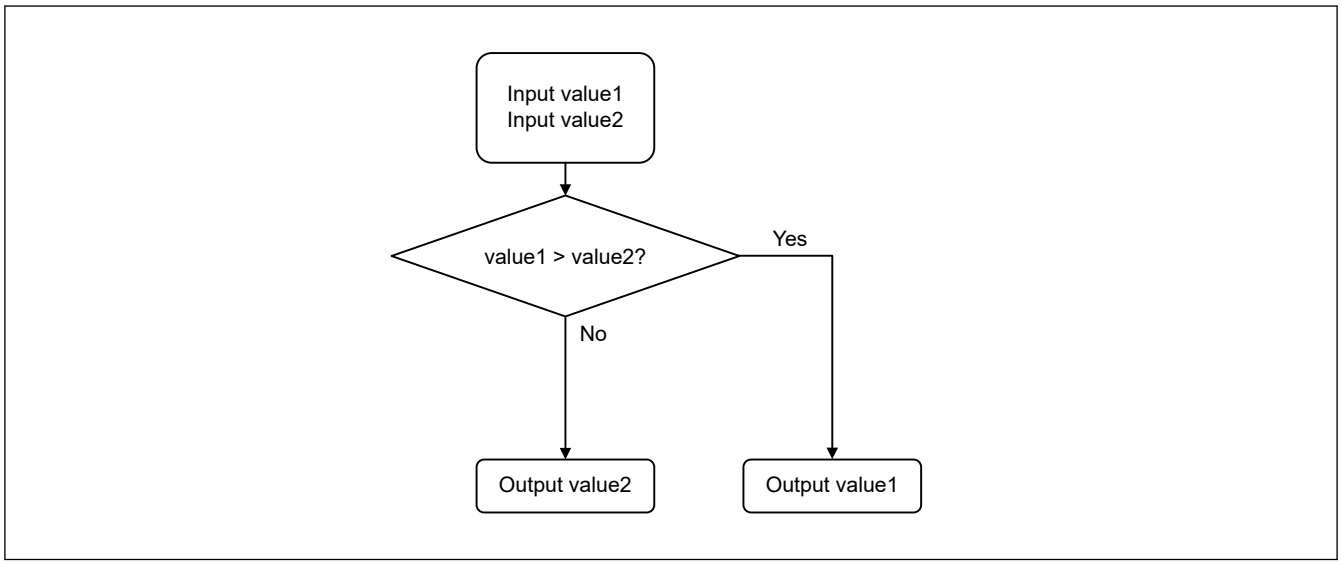


Figure 55.14 Combiner operated in maximum mode with union

55.6.2.6 Rasterization optimization

During rasterization, it is necessary to step through the whole bounding box one pixel at a time. This requirement can lead to an unnecessary number of steps for pixels that are not drawn. The 2D Drawing Engine provides optimization methods designed to reduce the number of steps required during rasterization. One optimization relies on the fact that any convex primitive can have only one span per line (a span is a contiguous horizontal line). This form of optimization detects a span start and saves the information for the next line. Another optimization is to detect a span end and stop rasterization for the current line.

(1) Spanstore

Consider the case in Figure 55.15.

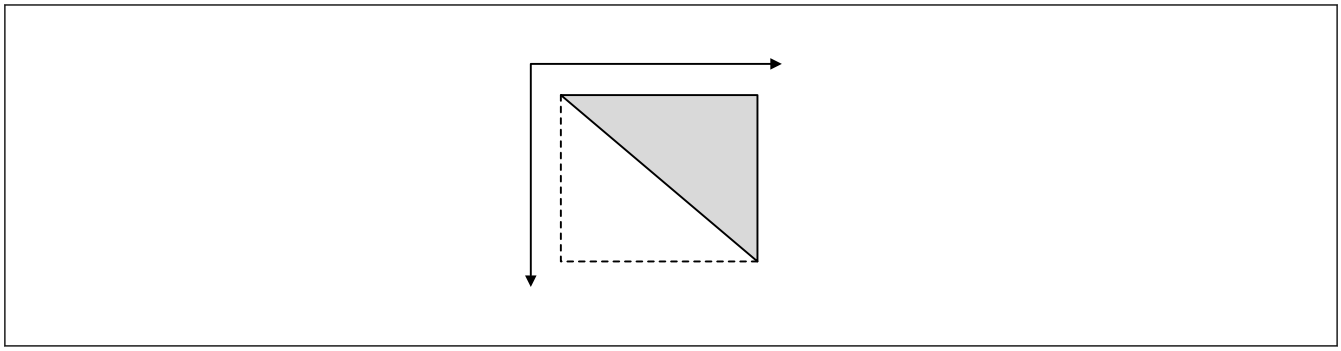


Figure 55.15 Triangle where the first edge is monotone growing

If the gray triangle must be rendered, half of the pixels processed by the rendering engine in the dotted bounding box would not be drawn. This situation can be optimized with the spanstore operation. When spanstore is activated and a span start is detected, the x position of the span start is detected.

In the next line the rendering starts with the stored x position. This only works if the edge is monotonically increasing ($y1 > y2 \Rightarrow x1 \geq x2$).

Consider the case in Figure 55.16.

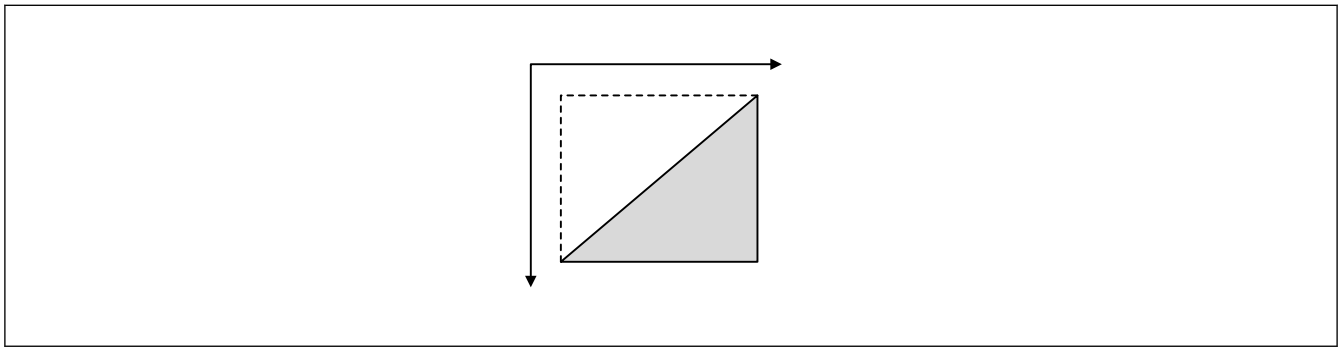


Figure 55.16 Triangle where the first edge is monotone falling

In this case, the normal spanstore operation cannot be performed. For this, the y direction of the rendering is reversed, and spanstore can operate again.

Consider the last case in Figure 55.17.

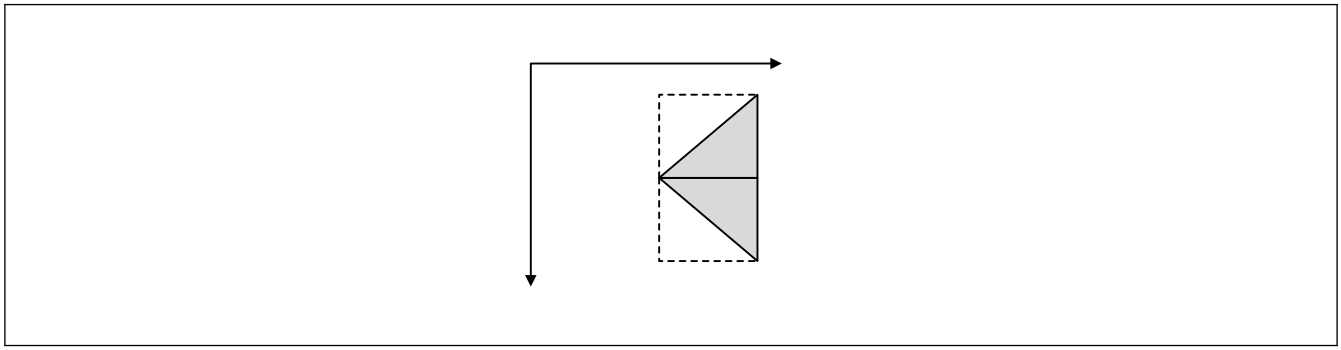


Figure 55.17 Triangle where the first edge is first monotone falling and then monotone growing

In this case, the triangle must be split and rendered as two parts for the spanstore optimization to work.

It is also possible to delay spanstore activation for a number of lines. This approach is used for rasterizing circles, as shown in Figure 55.18.

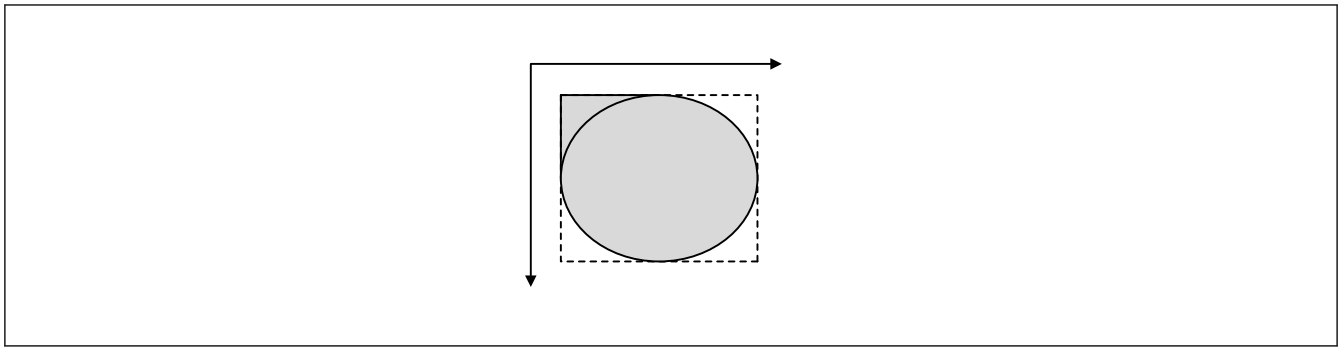


Figure 55.18 Full circle where the first edge is monotone falling for the first half and monotone growing for the second half

In this case, spanstore cannot be activated in the top left corner but can be activated in the bottom left corner. The empty corners in the top right and the bottom right cannot be rasterized because of the spanabort optimization.

(2) Spanabort

The second optimization assumes that the object that must be drawn is convex, which means there is only one span per scan line to be drawn. A non-convex object includes an object such as a triangle that is not filled and only consists of a thick border.

For a convex object, the rasterization can be stopped when the end of the first span is detected. No other constraints apply to this optimization for convex objects.

(3) Optimization efficiency

The efficiency of the optimizations can be seen for a typical case in Figure 55.19. In this case, the triangle is always rendered as single piece and is not separated into multiple triangles for higher optimizations. For this, the spanstore delay is used.

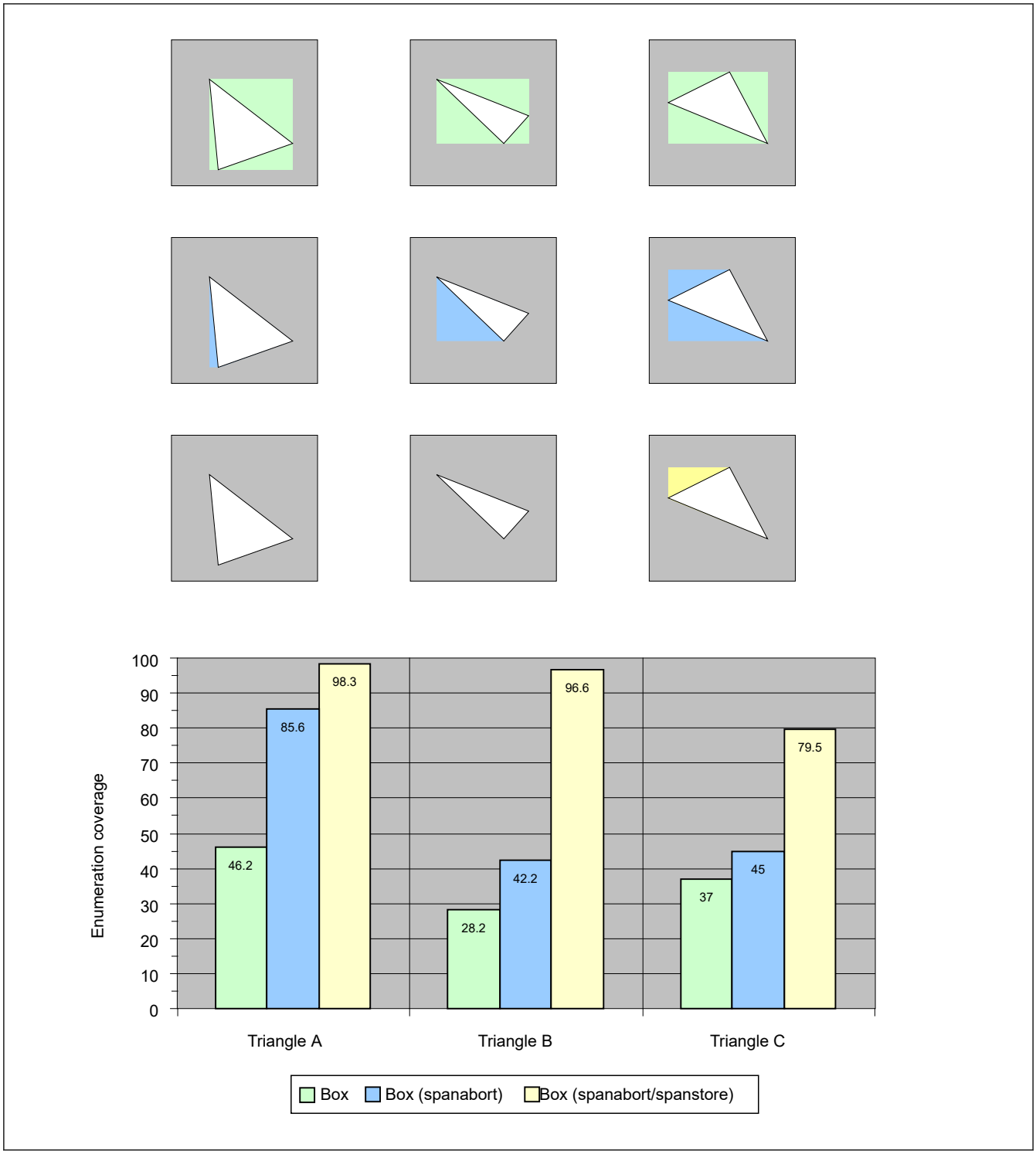


Figure 55.19 Efficiency of spanstore and spanabort optimizations with enumeration coverage equal to pixels of primitive/pixels of bounding box

55.6.3 Texturing

The texture unit can cover any primitive with a picture. The picture can be stretched, sheared, rotated, and translated in one step. To avoid aliasing, the result can be filtered bilinear in the u and v directions.

55.6.3.1 Mathematical background

The arbitrary mapping problem is completely determined by a mapping from 3 points in the object space (x, y) to 3 points in the texture space (u, v).

Consider the following mapping:

$$\vec{p}_0 = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \Rightarrow \left(\widetilde{\vec{p}_0}\right) = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\vec{p}_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \Rightarrow \left(\widetilde{\vec{p}_1}\right) = \begin{pmatrix} u_1 \\ v_1 \end{pmatrix} = \begin{pmatrix} w \\ 0 \end{pmatrix}$$

$$\vec{p}_2 = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \Rightarrow \left(\widetilde{\vec{p}_2}\right) = \begin{pmatrix} u_2 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ h \end{pmatrix}$$

where w is the width of the texture and h is the height of the texture.

Examine Figure 55.20 in the object space. To simplify calculations the difference vectors are taken as calculations:

$$\vec{d}_1 = \vec{p}_1 - \vec{p}_0$$

$$\vec{d}_2 = \vec{p}_2 - \vec{p}_0$$

This is equivalent to transforming from coordinate system O to coordinate system O'.

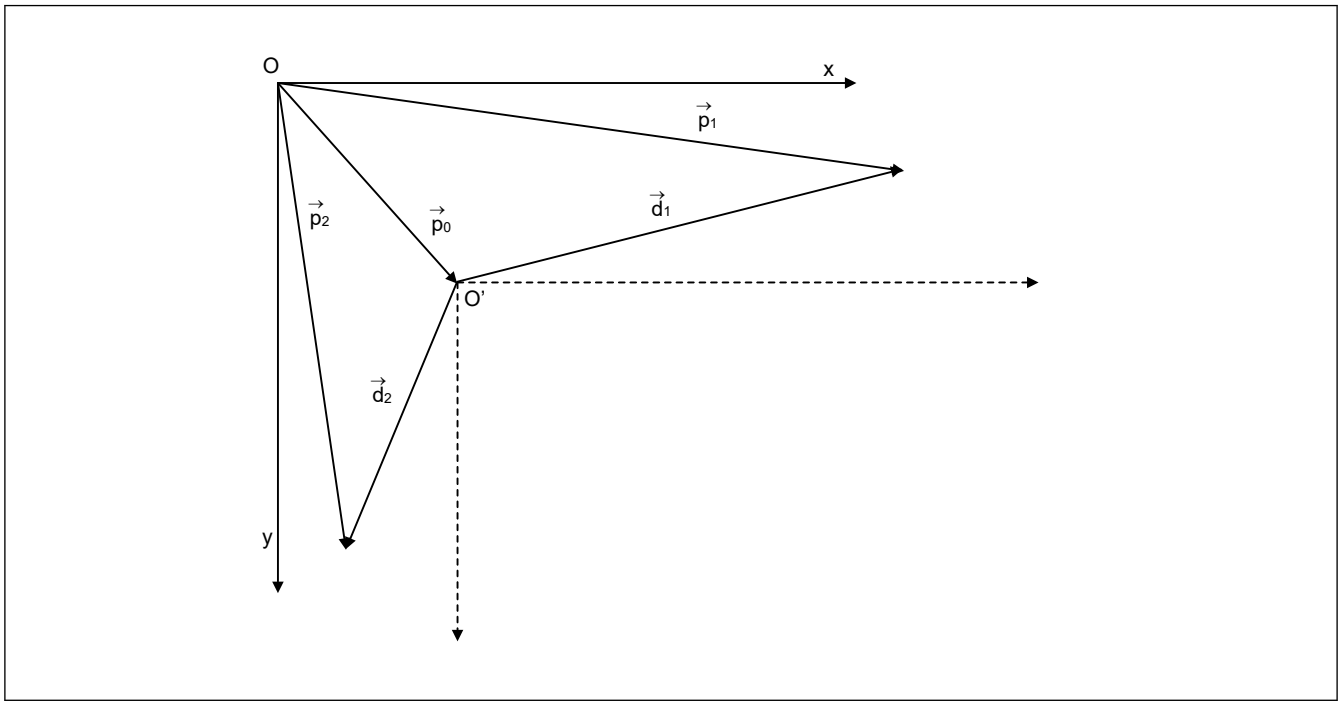


Figure 55.20 Texture mapping, object space, and transformation from coordinate system O to O' to simplify calculations

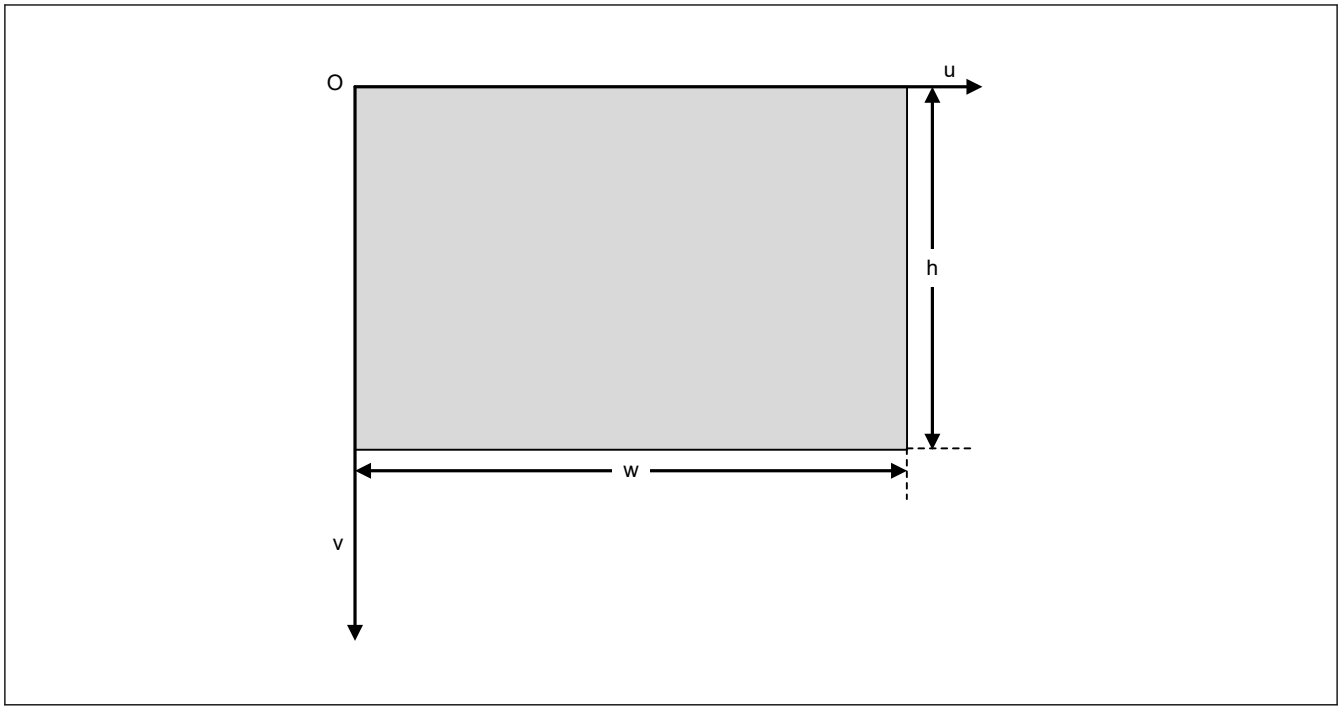


Figure 55.21 Texture mapping, texture space, and texture with width w and height h

In O', the mapping is:

$$\overrightarrow{\mathrm{p}'}_0 = \begin{pmatrix}0\\0\end{pmatrix} \Rightarrow \left(\widetilde{\overrightarrow{p}}_0\right) = \begin{pmatrix}u_0\\v_0\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix}$$

$$\overrightarrow{d}_1 = \begin{pmatrix}\mathrm{dx}_1\\\mathrm{dy}_1\end{pmatrix} \Rightarrow \left(\widetilde{\overrightarrow{p}}_1\right) = \begin{pmatrix}u_1\\v_1\end{pmatrix} = \begin{pmatrix}w\\0\end{pmatrix}$$

$$\overrightarrow{d}_2 = \begin{pmatrix}\mathrm{dx}_2\\\mathrm{dy}_2\end{pmatrix} \Rightarrow \left(\widetilde{\overrightarrow{p}}_2\right) = \begin{pmatrix}u_2\\v_2\end{pmatrix} = \begin{pmatrix}0\\h\end{pmatrix}$$

This is a linear mapping that can be described by a 2 x 2 matrix.

$$\left(\widetilde{\overrightarrow{p}}\right) = M \cdot \overrightarrow{\mathrm{p}'} = \begin{bmatrix}m_{11} & m_{12}\\m_{21} & m_{22}\end{bmatrix} \cdot \overrightarrow{\mathrm{p}'}$$

$$\Rightarrow \begin{pmatrix}\mathrm{w}\\0\end{pmatrix} = M \cdot \overrightarrow{d}_1 \wedge \begin{pmatrix}0\\h\end{pmatrix} = M \cdot \overrightarrow{d}_2$$

If the equations are expanded and sorted, the result is two equation systems each with two unknowns. These can be described more easily with a new matrix.

Let A = $\begin{bmatrix}\mathrm{dx}_1 & \mathrm{dy}_1\\\mathrm{dx}_2 & \mathrm{dy}_2\end{bmatrix}$ then the equation can be rewritten as:

$$\begin{pmatrix}\mathrm{w}\\0\end{pmatrix} = A \cdot \begin{pmatrix}m_{11}\\m_{12}\end{pmatrix} \wedge \begin{pmatrix}0\\h\end{pmatrix} = A \cdot \begin{pmatrix}m_{21}\\m_{22}\end{pmatrix}$$

This can be easily solved with determinants.

Let $\mathrm{c} = \frac{1}{\mathrm{detA}} = \frac{1}{\mathrm{dx}_1 \cdot \mathrm{dy}_2 - \mathrm{dx}_2 \cdot \mathrm{dy}_1}$

The resulting constants are:

$$m_{11} = c \cdot w \cdot \mathrm{dy}_2 = \frac{\mathrm{du}}{\mathrm{dx}}$$

$$m_{12} = -\mathrm{c} \cdot w \cdot \mathrm{dx}_2 = \frac{\mathrm{du}}{\mathrm{dy}}$$

$m_{21} = -\mathrm{c} \cdot h \cdot \mathrm{dy}_1 = \frac{\mathrm{dv}}{\mathrm{dx}}$

$m_{22} = c \cdot h \cdot \mathrm{dx}_1 = \frac{\mathrm{dv}}{\mathrm{dy}}$

To calculate the start values for u and v at the top of the bounding box, the transformation from O' to O must be reversed. Let U_s and V_s be the start values, then:

$$\begin{pmatrix} U_s \\ V_s \end{pmatrix} = M \cdot \left(-\vec{p}_0 \right) = c \cdot \begin{pmatrix} -\mathrm{w} \cdot (x_0 \cdot \mathrm{dy}_2 - y_0 \cdot \mathrm{dx}_2) \\ h \cdot (x_0 \cdot \mathrm{dy}_1 - y_0 \cdot \mathrm{dx}_1) \end{pmatrix}$$

(1) Examples

U and v are in the texture space. Enter the following into any case:

- Copy case
  $\mathrm{dx}_1 = 1, \mathrm{dx}_2 = 0, \mathrm{dy}_1 = 0, \mathrm{dy}_2 = 1$
- Scaling case x scaling copy case
  $\mathrm{dx}_1 = f, \mathrm{dx}_2 = 0, \mathrm{dy}_1 = 0, \mathrm{dy}_2 = 1$
  with f being the scaling factor in the x direction similar for the y direction.
- Rotation case
  $\mathrm{dx}_1 = \mathrm{cosa}, \mathrm{dx}_2 = -\mathrm{sina}, \mathrm{dy}_1 = \mathrm{sina}, \mathrm{dy}_2 = \mathrm{cosa}$
  with the angle between d1 and the x axis in the clockwise direction.

55.6.3.2 Limiter operation

The texture limiters operate exactly the same as the spatial limiters shown in Figure 55.8

The register layout for the u limiter is the same.

- LUSTART = U_s
- LUXADD = du/dx
- LUYADD = du/dy.

The register layout for the v limiter is slightly different. TEXPITCH is multiplied to save one hardware multiplier.

- LVSTARTI = floor (V_s) • TEXPITCH
  Contains the integer part of the start value.
- LVSTARTF = (V_s - floor (V_s)) • TEXPITCH
  Contains the fractional part of the start value.
- LVXADDI = floor (dv/dx) • TEXPITCH
  Contains the integer part of dv/dz.
- LVYADD = floor (dv/dy) • TEXPITCH
  Contains the integer part of dv/dy.
- LVYXADDF = (dv/dy - floor (dv/dy)) • TEXPITCH ((dv/dx - floor (dv/dx)) • TEXPITCH)
  Contains the fractional part of dv/dy and dv/dx combined in one register.
- TEXMASK
  Contains a mask for u and v separately to wrap around values of u and v. This is useful for staying inside the limits of the texture or repeat a texture. Wrap around textures have to have a size multiple of 2.
- TEXPITCH
  Contains the width of the texture in pixels in framebuffer memory. This information is required to calculate the new address if stepping to a new line.
- TEXORIGIN
  Contains the base address of the texture.

55.6.4 Colorization

After a pixel is found to be part of the geometry, its color is calculated. The 2D Drawing Engine supports a very general color calculation scheme, allowing support for several color modes. This color scheme uses an interpolation between two color registers, COLOR1 and COLOR2. See Figure 55.22 for details. COLOR1 and COLOR2 are marked as A, B in the figure for clarity.

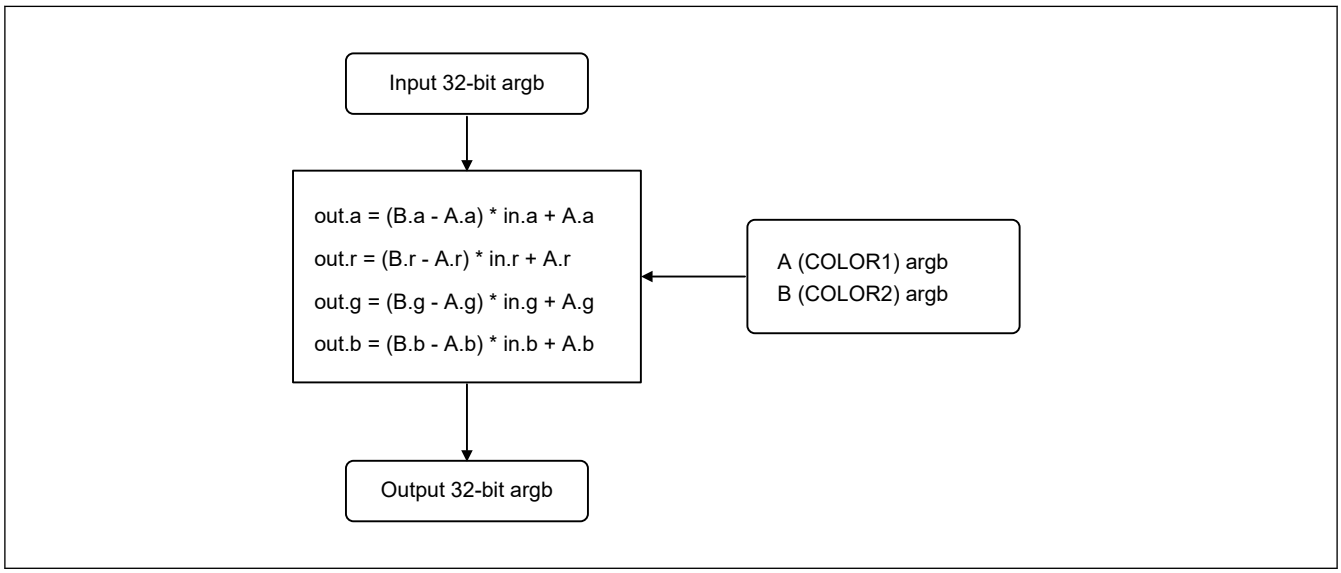


Figure 55.22 Colorization step and interpolation between the two color registers, A (COLOR1) and B (COLOR2)

This general approach can be used to support several different color modes that can be individually applied to any color or alpha channel of the input.

Table 55.8 Colorization operations

Operation	Settings for A and B*1
Copy	A = 0, B = 0xff
Replace with a constant value v	A = v, B = v
Multiply by a constant value v	A = 0, B = v
Colorize an alpha texture with the RGB value v	A.a = 0, A.r = B.r, A.g = B.g, A.b = B.b, B.a = 0xff, B.r = v.r, B.g = v.g, B.b = v.b
Invert a channel	A = 0xff, B = 0
Invert multiply with v	A = v, B = 0
Interpolate between color v and color u	A = v, B = u

Note 1. A = COLOR1, B = COLOR2

55.6.5 Blending

55.6.5.1 Color channel blending

The last step before the pixel is written to the framebuffer is to blend the pixel with the data that is already written to the framebuffer. If blending is activated, the framebuffer, referred to as DST, must be read. SRC is the output from the colorization unit.

The following color blend modes are supported:

- SRC_ZERO
- SRC_ONE
- SRC_ALPHA

- SRC_ONE_MINUS_ALPHA
- DST_ZERO
- DST_ONE
- DST_ALPHA
- DST_ONE_MINUS_ALPHA.

The selection of the color channel blend modes is performed with the following flags:

- BSF: blend source factor is alpha
- BSI: blend source factor invert
- BDF: blend destination factor is alpha
- BDI: blend destination factor invert.

The formula for the blending is:

$dst = src \cdot f_S + dst \cdot f_D$

where:

$BSF = 0, BSI = 0 \Rightarrow f_S = 1$

$BSF = 1, BSI = 0 \Rightarrow f_S = \alpha$

$BSF = 0, BSI = 1 \Rightarrow f_S = 0$

$BSF = 1, BSI = 1 \Rightarrow f_S = 1 - \alpha$

$BDF = 0, BDI = 0 \Rightarrow f_D = 1$

$BDF = 1, BDI = 0 \Rightarrow f_D = \alpha$

$BDF = 0, BDI = 1 \Rightarrow f_D = 0$

$BDF = 1, BDI = 1 \Rightarrow f_D = 1 - \alpha$

Table 55.9 lists all possible color channel blend modes.

Table 55.9 Color channel blend modes

Mode	BSF	BSI	BDF	BDI	Blend equation
SRC_ONE DST_ONE	0	0	0	0	SRC + DST
SRC_ONE	0	0	0	1	SRC
SRC_ONE DST_ALPHA	0	0	1	0	SRC + DST × ALPHA
SRC_ONE DST_ONE_MINUS_ALPHA	0	0	1	1	SRC + DST × (1 - ALPHA)
SRC_ZERO DST_ONE	0	1	0	0	DST
SRC_ZERO DST_ZERO	0	1	0	1	0
SRC_ZERO DST_ALPHA	0	1	1	0	DST × ALPHA
SRC_ZERO DST_ONE_MINUS_ALPHA	0	1	1	1	DST × (1 - ALPHA)
SRC_ALPHA DST_ONE	1	0	0	0	SRC × ALPHA + DST
SRC_ALPHA	1	0	0	1	SRC × ALPHA
SRC_ALPHA DST_ALPHA	1	0	1	0	SRC × ALPHA + DST × ALPHA
SRC_ALPHA DST_ONE_MINUS_ALPHA	1	0	1	1	SRC × ALPHA + DST × (1 - ALPHA)
SRC_ONE_MINUS_ALPHA DST_ONE	1	1	0	0	SRC × (1 - ALPHA) + DST
SRC_ONE_MINUS_ALPHA	1	1	0	1	SRC × (1 - ALPHA)
SRC_ONE_MINUS_ALPHA DST_ALPHA	1	1	1	0	SRC × (1 - ALPHA) + DST × ALPHA
SRC_ONE_MINUS_ALPHA DST_ONE_MINUS_ALPHA	1	1	1	1	SRC × (1 - ALPHA) + DST × (1 - ALPHA)

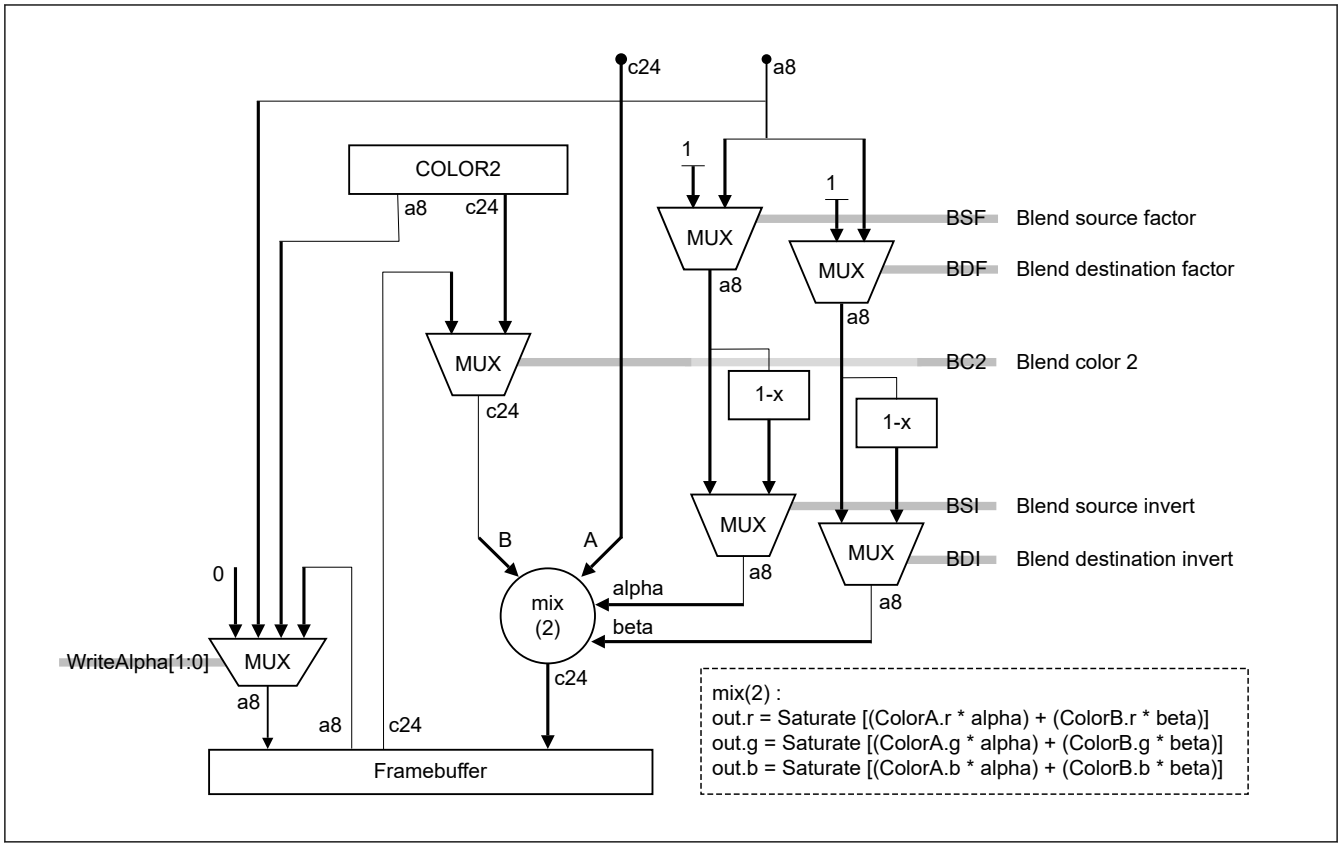


Figure 55.23 Color channel blend unit when CONTROL2.USEACB = 0

55.6.5.2 Alpha channel blending

The alpha channel can be blended in addition to the color channels. Alpha channel blending is enabled by setting CONTROL2.USEACB = 1. Alpha channel blending uses the same formulas and same blend modes as for the color channels. The alpha channel formulas can be set independently from the color channels.

The following alpha channel blend modes are supported:

- SRC_A_ZERO
- SRC_A_ONE
- SRC_A_SRC_A
- SRC_A_ONE_MINUS_SRC_A
- DST_A_ZERO
- DST_A_ONE
- DST_A_SRC_A
- DST_A_ONE_MINUS_SRC_A.

The alpha channel blend modes selected with the following flags:

- BSFA: blend source factor is SRC_A
- BSIA: blend source factor invert
- BDFA: blend destination factor is SRC_A
- BDIA: blend destination factor invert.

The formula for the blending is:

$dst\_alpha = src\_a \cdot f_{S\_a} + dst\_a \cdot f_{D\_a}$

where:

BSFA = 0, BSIA = 0 ⇒ $f_{S\_a} = 1$

BSFA = 1, BSIA = 0 ⇒ $f_{S\_a} = src\_a$

BSFA = 0, BSIA = 1 ⇒ $f_{S\_a} = 0$

BSFA = 1, BSIA = 1 ⇒ $f_{S\_a} = 1 - src\_a$

BDFA = 0, BDIA = 0 ⇒ $f_{D\_a} = 1$

BDFA = 1, BDIA = 0 ⇒ $f_{D\_a} = src\_a$

BDFA = 0, BDIA = 1 ⇒ $f_{D\_a} = 0$

BDFA = 1, BDIA = 1 ⇒ $f_{D\_a} = 1 - src\_a$

Table 55.10 lists all possible alpha channel blend modes.

Table 55.10 Alpha channel blend modes

Mode	BSF	BSI	BDF	BDI	Blend equation
SRC_A_ONE DST_A_ONE	0	0	0	0	SRC_A + DST_A
SRC_A_ONE	0	0	0	1	SRC_A
SRC_A_ONE DST_A_SRC_A	0	0	1	0	SRC_A + DST_A × SRC_A
SRC_A_ONE DST_A_ONE_MINUS_SRC_A	0	0	1	1	SRC_A + DST_A × (1 - SRC_A)
SRC_A_ZERO DST_A_ONE	0	1	0	0	DST_A
SRC_A_ZERO DST_A_ZERO	0	1	0	1	0
SRC_A_ZERO DST_A_SRC_A	0	1	1	0	DST_A × SRC_A
SRC_A_ZERO DST_A_ONE_MINUS_SRC_A	0	1	1	1	DST_A × (1 - SRC_A)
SRC_A_SRC_A DST_A_ONE	1	0	0	0	SRC_A × SRC_A + DST_A
SRC_A_SRC_A	1	0	0	1	SRC_A × SRC_A
SRC_A_SRC_A DST_A_SRC_A	1	0	1	0	SRC_A × SRC_A + DST_A × SRC_A
SRC_A_SRC_A DST_A_ONE_MINUS_SRC_A	1	0	1	1	SRC_A × SRC_A + DST_A × (1 - SRC_A)
SRC_A_ONE_MINUS_SRC_A DST_A_ONE	1	1	0	0	SRC_A × (1 - SRC_A) + DST_A
SRC_A_ONE_MINUS_SRC_A	1	1	0	1	SRC_A × (1 - SRC_A)
SRC_A_ONE_MINUS_SRC_A DST_A_SRC_A	1	1	1	0	SRC_A × (1 - SRC_A) + DST_A × SRC_A
SRC_A_ONE_MINUS_SRC_A DST_A_ONE_MINUS_SRC_A	1	1	1	1	SRC_A × (1 - SRC_A) + DST_A × (1 - SRC_A)

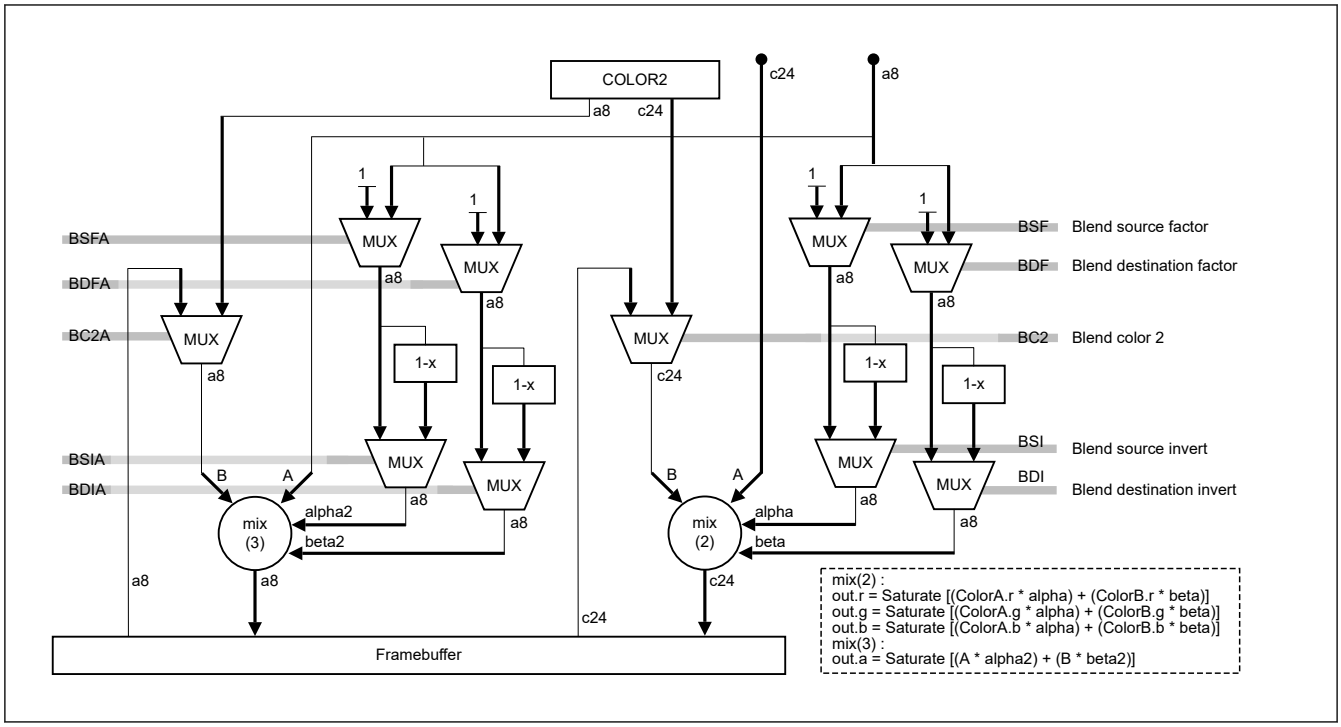


Figure 55.24 Alpha- and color-channel blend unit when CONTROL2.USEACB = 1

55.7 Rendering Modes

The rendering process can be performed in two different modes, register mode and display list mode.

55.7.1 Register Mode

In register mode, when operation is based on register settings, the host CPU configures and initiates each render process separately. To start a new render process, the host CPU must wait until the previous one is completed. In this mode, the host CPU is heavily engaged throughout the entire drawing procedure and is consequently to a large extent unavailable for other tasks.

The host CPU must set up all registers for performing a certain drawing operation before it can start the rendering process. A new register setup can only be started when the previous render process has completed. Before starting a new register setup, make sure that:

- STATUS.DLISTACTIVE = 0: Display list reader is idle
- STATUS.BUSENUM = 0: Pixel selection unit is idle.

Lastly, write the framebuffer start address to the ORIGIN register. This write triggers the 2D Drawing Engine to start rendering.

55.7.2 Display List Mode

In display list mode, the host CPU creates a display list in memory prior to starting the 2D Drawing Engine. Such a display contains a bundle of render operations. When started, the 2D Drawing Engine executes the display list autonomously in parallel with the host CPU, which is not involved with drawing operations most of the time. Use of a display list allows a fully asynchronous operation of the host CPU and the 2D Drawing Engine and offers the best possible system performance.

In this mode, the display list reader reads a memory block containing instructions on how to set the 2D Drawing Engine control registers and executes these control register writes accordingly.

(1) Display list start

To start execution of a display list, which already resides in memory, the start address of the display list is written to the display list start address register DLISTSTART. Because rewriting DLISTSTART also stops any ongoing display list execution, make sure that the previous display list process is completed by either of the following two methods:

- Check that STATUS.DLISTACTIVE = 0, which indicates idle status of the display list reader
- Wait for the display list interrupt DRWDLISTIRQ, which indicates completion of the previous display list process.

Caution: Direct writing to 2D Drawing Engine registers while the display list mode is active (STATUS.DLISTACTIVE = 1) might lead to a 2D Drawing Engine hang-up. To prevent this, always check that the display list reader is idle ((STATUS.DLISTACTIVE = 0) before writing to any 2D Drawing Engine register.

(2) Display list format

Display lists are stored using direct register-to-value mappings, which means that the display list contains a one byte index that addresses a certain register and the value to be written to the register. The register index is derived from the address offset of the register address and can be calculated by dividing the address offset by 4. For the index of each register, see section 55.2. Register Descriptions.

As the 2D Drawing Engine registers are always 32 bits wide, each data unit (called a data word) to be written to a register is of the same size. An address word that contains the indices of the register to be written is stored in a packed notation with up to four indices stored in one 32-bit address word.

A display list command always starts with an address word, followed by up to four data words, one for each register. The indices are read and interpreted from LSB to MSB, so the register of the low byte index is written first.

(3) Example

In the following example:

- DWORD 0x201A_1930 // start of list address word
- DWORD 0x0000_0013 // data word 1 (for register 0x30)
- DWORD 0xFFFF_FFAA // data word 2 (for register 0x19)
- DWORD 0x4033_6480 // data word 3 (for register 0x1A)
- DWORD 0x0001_0000 // data word 4 (for register 0x20)
- DWORD... // next address word.

This stream of dwords updates the DRW registers as follows:

- Write 0x0000_0013 to register 0x30 = 48, which is IRQCTL
- Write 0xFFFF_FFAA to register 0x19 = 25, which is COLOR1
- Write 0x4033_6480 to register 0x1A = 26, which is COLOR2
- Write 0x0001_0000 to register 0x20 = 32, which is ORIGIN.

(4) Address word indices

Besides referencing a register, the indices of an address word can also have other meanings, depending on their value.

Table 55.11 Address indices function (1 of 2)

Index	Function	
0x00 to 0x7F	Register indices Two register indices trigger additional actions:	
	0x20 = 32:	A write to ORIGIN to set a new frame buffer address is delayed until the ongoing frame buffer write-back is complete, when STATUS.BUSYWRITE = 0.
	0x32 = 50:	A write to DLISTSTART sets a new display list start address stops the current display list and starts the new one.
0x80	Gap index, which is used to fill unused bytes of an address word. For example, if fewer than four indices are required, the remaining bytes are filled with 0x80. In this case, the number of the subsequent data words must be reduced accordingly.	

Table 55.11 Address indices function (2 of 2)

Index	Function	
0xFF	If the first index of an address word contains the special index 0xFF, the subsequent (second) index is interpreted as follows:	
	Bit [0] set:	Display list end.
	Bit [1] set:	Issue a full pipeline flush and wait (necessary before flip).
	Bit [2] set:	Wait for writeback complete (necessary before framebuffer format change).
	Bits [3:7]	Set all to 0s.
	Bit [1] and [2] settings are mutually exclusive. All indices after the special index 0xFF are ignored, and the next address word is read, if no display list end (bit [0] = 1) was set.	
	The remaining two indices must be set to 0x00.	

Caution : Gap indices 0x80 must not be placed between other indices, for example as “index1 - 0x80 - index3 - index4”. Always fill all indices after 0x80 with the gap index.

Caution : If any of the special indices 0x80 and 0xFF are used, no register index can follow after them in the same address word.

Table 55.12 2D Drawing Engine registers overview (1 of 2)

Register function	Symbol	Index
Control registers:		
Geometry control 0	CONTROL	0
Surface control	CONTROL2	1
Interrupt control	IRQCTL	48
Cache control	CACHECTL	49
Status control	STATUS	N/A*1
Hardware version and feature set ID	HWREVISION	N/A*1
Color registers:		
Base color	COLOR1	25
Secondary color	COLOR2	26
Pattern	PATTERN	29
Limiter registers:		
Limiter 1 start value	L1START	4
Limiter 2 start value	L2START	5
Limiter 3 start value	L3START	6
	L4START	7
Limiter 5 start value	L5START	8
Limiter 6 start value	L6START	9
Limiter 1 x-axis increment	L1XADD	10
Limiter 2 x-axis increment	L2XADD	11
Limiter 3 x-axis increment	L3XADD	12
Limiter 4 x-axis increment	L4XADD	13
Limiter 5 x-axis increment	L5XADD	14
Limiter 6 x-axis increment	L6XADD	15
Limiter 1 y-axis increment	L1YADD	16
Limiter 2 y-axis increment	L2YADD	17
Limiter 3 y-axis increment	L3YADD	18

Table 55.12 2D Drawing Engine registers overview (2 of 2)

Register function	Symbol	Index
Limiter 4 y-axis increment	L4YADD	19
Limiter 5 y-axis increment	L5YADD	20
Limiter 6 y-axis increment	L6YADD	21
Limiter 1 band width parameter	L1BAND	22
Limiter 2 band width parameter	L2BAND	23
Texture registers:		
Texture base address	TEXORIGIN	47
Texels per texture line	TEXPITCH	45
Texture size or texture address mask	TEXMASK	46
U limiter start value	LUSTART	36
U limiter x-axis increment	LUXADD	37
U limiter y-axis increment	LUYADD	38
V limiter start value integer part	LVSTARTI	39
V limiter start value fractional part	LVSTARTF	40
V limiter x-axis increment integer part	LVXADDI	41
V limiter y-axis increment integer part	LVYADDI	42
V limiter increment fractional parts	LVYXADDF	43
Color lookup table start address	TEXCLADDR	55
Write Data to DRWTEXCLADDR; after each data write, DRWTEXCLADDR is incremented by 1.	TEXCLDATA	56
Offset to the index for the indexed texture formats i8, i4, i2, and i1	TEXCLOFFSET	57
Compare value for R,G, B components of internal texel color representation.	COLKEY	58
Miscellaneous registers:		
Bounding box dimension	SIZE	30
Framebuffer pitch and spanstore delay	PITCH	31
Address of the first pixel in framebuffer	ORIGIN	32
Display list start address	DLISTSTART	50
Performance counters control	PERFTRIGGER	53
Performance counter 1	PERFCOUNT1	51
Performance counter 2	PERFCOUNT2	52

Note 1. These registers are read-only and cannot be accessed in display list mode, and they therefore have no index.

55.7.3 Stopping the Render Process

Stopping an ongoing render process requires a specific procedure, which is described in section 55.10. Stopping the 2D Drawing Engine Render Process.

55.8 Interrupts

The 2D Drawing Engine generates three interrupts:

- DRWBUSIRQ
- DRWENUMIRQ
- DRWDLISTIRQ.

55.8.1 Interrupt sources

(1) DRWBUSIRQ

This is the 2D Drawing Engine bus error interrupt. It occurs when the 2D Drawing Engine attempts to access an undefined address range through the following registers:

- Framebuffer Base Address Register ORIGIN
- Texture Base Address Register TEXORIGIN
- Display List Start Address Register (DLISTSTART).

The bus error interrupt DRWBUSIRQ is then generated. The bus error interrupt only serves for debugging purposes. The interrupt source can be determined by the BUSERRMFB, BUSERRMTXMRL, and BUSERRMDL bits of the STATUS register.

Note: After a DRWBUSIRQ occurrence, you must apply a system reset.

(2) DRWENUMIRQ

This is the current render process finished interrupt.

(3) DRWDLISTIRQ

This is the display list interrupt. It is asserted on completion of a display list process. DRWDLISTIRQ is activated if one of the following conditions is true:

- The entire display list is complete
- The display list processing stops because a new display list start is triggered by a write to the Display List Start Address Register (DLISTSTART).

55.8.2 Interrupt Control

The three 2D Drawing Engine interrupts are combined into a single shared interrupt, DRW_IRQ, to the CPU. Each interrupt can be masked (disabled), or unmasked (enabled) by setting its associated enable bit in the Interrupt Control Register (IRQCTL).

The occurrence of an enabled interrupt (one with its mask bit set to 1 in IRQCTL) is monitored in the Status Control Register (STATUS) with its associated interrupt status bit is set to 1. The shared 2D Drawing Engine interrupt DRW_IRQ is then generated.

To clear the interrupt, the host CPU must write 1 to the interrupt clear bit in IRQCTL. The interrupt clear bit returns to 0 automatically.

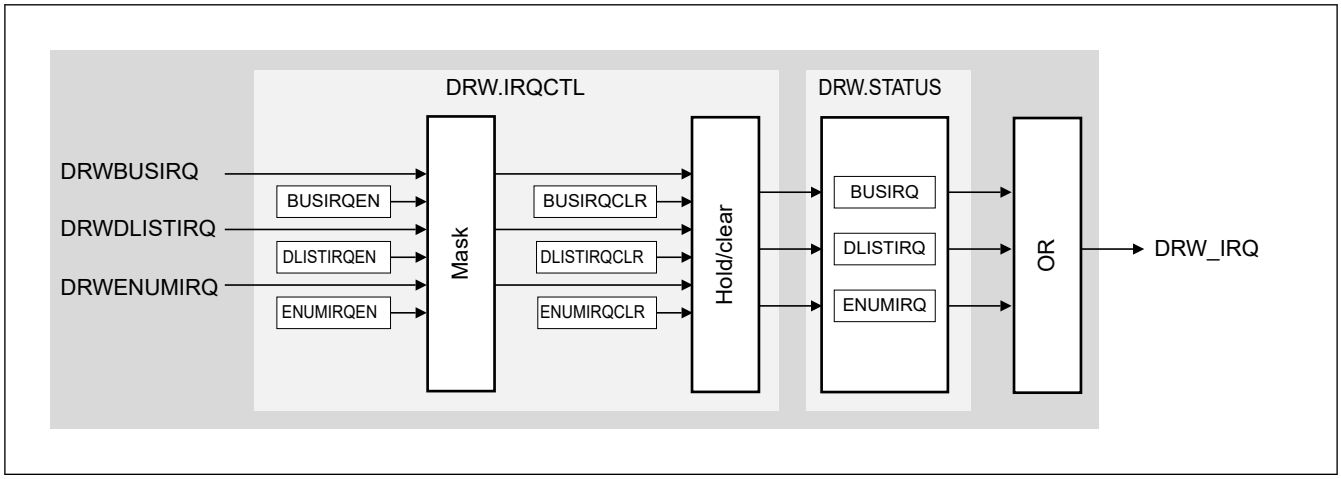


Figure 55.25 Interrupt Controller Unit (ICU)

55.9 Performance Counters

The 2D Drawing Engine features two independent 32-bit performance counter registers (PERFCOUNTk (k = 1, 2)) to count the number of occurrences of a certain event. The events to be counted can be set up independently for each performance counter register with the Performance Counter Control Registers, PERFTRIGGER.PERFTRIGGER2 for PERFCOUNT2 and PERFTRIGGER.PERFTRIGGER1 for PERFCOUNT1.

Table 55.13 lists the performance counter trigger events that can be selected.

Table 55.13 Performance counter trigger events

PERFTRIGGER.PERFTRIGGERk	Event
0	Disable performance counter
1	2D Drawing Engine active cycles
2	Framebuffer read access
3	Framebuffer write access
4	Texture read access
5	Invisible pixels (enumerated but selected with alpha 0%)
6	Invisible pixels while internal FIFO is empty (lost cycles)
7	Display list reader active cycles
8	Framebuffer read hits
9	Framebuffer read misses
10	Framebuffer write hits
11	Framebuffer write misses
12	Texture read hits
13	Texture read misses
31	Every clock cycle (for use as timer)

55.10 Stopping the 2D Drawing Engine Render Process

If a render process has started either in register or display list mode, the 2D Drawing Engine processes the data autonomously until the render process is finished. Depending on the rendering, this process might take several milliseconds.

If the 2D Drawing Engine is to be disabled because, for example, the MCU enters a low power mode, proceed as follows to stop the ongoing rendering:

1. Set the following registers as follows:
   SIZE = 0x00010001
   Set bounding box dimensions to 1 pixel x 1 line.
   CONTROL2 = 0x00000000
   Color format A (8), no texture, CLUT.
   ORIGIN = UnmappedAddress
   The UnmappedAddress is an address that is not available for 2D Drawing Engine access.
   The recommended UnmappedAddress is given here under the key word “UnmappedAddress”.
   Alternatively do the same in display list mode:
   DWORD 0x8020011E // start of “address word” list
   DWORD 0x00010001 // SIZE = 0x00010001
   DWORD 0x00000000 // CONTROL2 = 0x00000000
   DWORD UnmappedAddress // ORIGIN = UnmappedAddress
2. Wait for the Bus Error corresponding to the unmapped address violation, which indicates access to an unmapped address and the stop of the render process.
   UnmappedAddress = 0xFFFFFFF0
3. Disable the 2D Drawing Engine, if wanted.

55.11 MathML sample

#1

$y = \widetilde{a}x + \widetilde{b}$

#2

$0 = f(x,y) = \widetilde{a}x - y + \widetilde{b} = \mathrm{ax} + \mathrm{by} + c$

with $a = \widetilde{a}$, $b = -1$, $c = \widetilde{b}$

#3

$\overrightarrow{p} = \begin{pmatrix} x \\ y \end{pmatrix}, \overrightarrow{n} = \begin{pmatrix} a \\ b \end{pmatrix} \Rightarrow f(x,y) = \mathrm{ax} + \mathrm{by} + c = \overrightarrow{p} \cdot \overrightarrow{n} + c$

#4

The projection of $\overrightarrow{p} - \overrightarrow{p}_0$ to $\overrightarrow{n}$ is the distance d of the point P to the line. In this case, f (x, y) describes the distance to the line of the pixel with coordinates (x, y).

#5

$\Delta\overrightarrow{p} = \overrightarrow{p}_1 - \overrightarrow{p}_0 = \begin{pmatrix} x_1 - x_0 \\ y_1 - y_0 \end{pmatrix} = \begin{pmatrix} \Delta\mathrm{x} \\ \Delta\mathrm{y} \end{pmatrix}$

#6

Consider the equation for a circle with the center at $\overrightarrow{c} = \begin{pmatrix} s \\ t \end{pmatrix}$ and radius r:

#7

$0 = f(x,y) = (x-s)^2 + (y-t)^2 - r^2$

#8

$0 = f(x,y) = \overrightarrow{p}_0 \cdot \overrightarrow{n} + c \Rightarrow c = -\overrightarrow{p}_0 \cdot \overrightarrow{n}$

#9 OLD

$\overrightarrow{p}_0 = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \Rightarrow \widetilde{(\overrightarrow{p}_0)} = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

#9 NEW

$\overrightarrow{p}_0 = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \Rightarrow \left(\widetilde{\overrightarrow{p}_0}\right) = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

#10 OLD

$\overrightarrow{n} = \begin{pmatrix} -\Delta\mathrm{y} \\ \Delta\mathrm{x} \end{pmatrix} / \left(\sqrt{\Delta x^2 + \Delta y^2}\right)$

#10 NEW

$\overrightarrow{n} = \begin{pmatrix} -\Delta\mathrm{y} \\ \Delta\mathrm{x} \end{pmatrix} \cdot \frac{1}{\sqrt{\Delta x^2 + \Delta y^2}}$

#11

$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$

#11-2

$\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}$

#12

$\mathrm{start2} = \mathrm{xadd1}$

#13

$f(x+1, y)$

#14

$$\begin{pmatrix} \mathrm{w} \\ 0 \end{pmatrix} = A \cdot \begin{pmatrix} m_{11} \\ m_{12} \end{pmatrix} \wedge \begin{pmatrix} 0 \\ h \end{pmatrix} = A \cdot \begin{pmatrix} m_{21} \\ m_{22} \end{pmatrix}$$

#15

$$\mathrm{sqrt}\left(x^2 + y^2\right) = \frac{\sqrt{x^2 + y^2}}{k}$$

#16

$$k = \prod_{i=0}^{\infty} \frac{1}{\sqrt{1 + 2^{-2i}}} \fallingdotseq 0.607252935008881256169 4$$

55.12 Usage notes

55.12.1 Module-stop function

DRW operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The DRW module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

55.12.2 Restriction for OSPI

There is a restriction to using OSPI, see section 37.4.5. Restriction in 8D-8D-8D profile 1.0 format.

56. Graphics LCD Controller (GLCDC)

56.1 Overview

The multifunctional Graphics LCD Controller (GLCDC) supports multiple data formats and panels. Key GLCDC features include:

- GLCDC0 and GLCDC1 bus master function for accessing graphics data
- Superimposition of three planes (single color background plane, graphics 1 plane, and graphics 2 plane)
- Supports many types of 32- and 16-bit/pixel graphics data and 8-, 4-, and 1-bit LUT data formats
- Digital interface signal output supporting the video image size up to WXGA(1280 x 800).

Figure 56.1 shows a block diagram of the GLCDC.

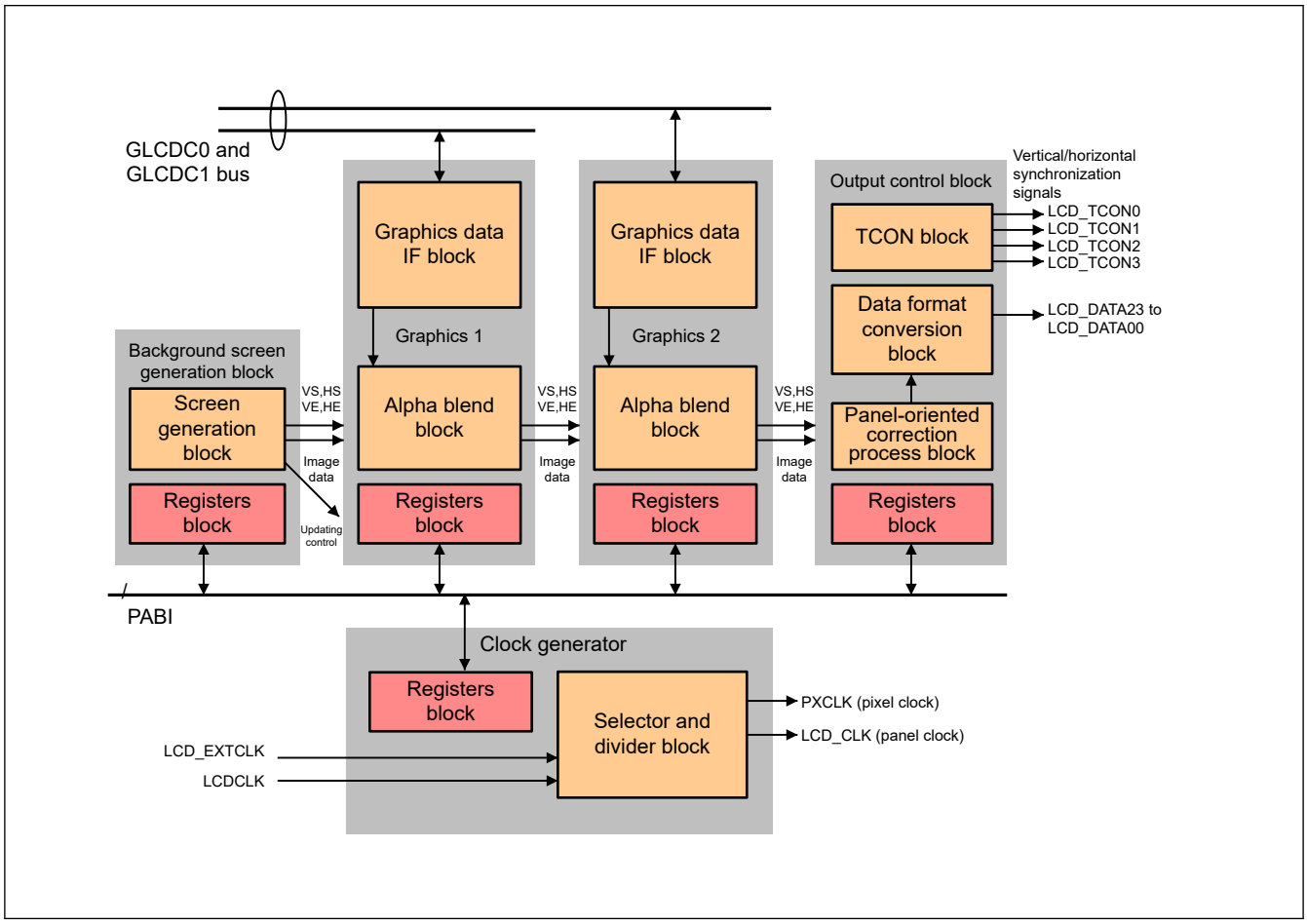


Figure 56.1 GLCDC block diagram

Table 56.1 GLCDC I/O pins

Function	Pin name	I/O	Description
GLCDC	LCD_EXTCLK	Input	Panel clock source input pin
	LCD_CLK	Output	Panel clock output pin
	LCD_DATA23 to LCD_DATA00	Output	LCD signal output pins RGB888 signal R/G/B: 8 bits each (unsigned) RGB666 signal R/G/B: 6 bits each (unsigned) RGB565 signal R/B: 5 bits each (unsigned) G: 6 bits (unsigned) Serial RGB signal R/G/B/-: 8 bits (unsigned)
	LCD_TCON3	Output	LCD_TCON3 output signal pin
	LCD_TCON2	Output	LCD_TCON2 output signal pin
	LCD_TCON1	Output	LCD_TCON1 output signal pin
	LCD_TCON0	Output	LCD_TCON0 output signal pin

Table 56.2 provides a functional overview of GLCDC.

Table 56.2 Functional overview of GLCDC (1 of 2)

Parameter		Function
Graphics	Graphics planes	• Single color background (lowest layer) and two graphics planes • Graphics planes can be alpha-blended with the lower-layer plane
	Pixel format	• RGB-888 progressive format (-: 8 bits, R: 8 bits, G: 8 bits, B: 8 bits; 32 bits in total) • ARGB8888 progressive format (A: 8 bits, R: 8 bits, G: 8 bits, B: 8 bits; 32 bits in total) • RGB565 progressive format (A: None, R: 5 bits, G: 6 bits, B: 5 bits; 16 bits in total) • ARGB1555 progressive format (CLUT: 1 bit, R: 5 bits, G: 5 bits, B: 5 bits; 16 bits in total) • ARGB4444 progressive format (A: 4 bits, R: 4 bits, G: 4 bits, B: 4 bits; 16 bits in total) • CLUT8 progressive format (color palette address: 8 bits) • CLUT4 progressive format (color palette address: 4 bits) • CLUT1 progressive format (color palette address: 1 bit) • CLUT memory: 512 words × 32 bits per graphics plane (ARGB8888)
	Frame buffer control	The following parameters can be set for the frame buffer: • Base address: Start address of the frame buffer, aligned with a 64-byte boundary (burst transfer size) • Macro line offset: Offset address from the start address to the next macro line, aligned with a 64-byte boundary (burst transfer size) • Frame offset: Offset address from the start address to the next frame, aligned with a 64-byte boundary (burst transfer size) • Number of data transfers: Number of data transfers of a macro line • Number of macro lines: Number of macro lines in a single frame
	Alpha blending	• Alpha blending in rectangular area (blending ratio: 256 gradation levels) • Alpha blending in pixel units (blending ratio: 256 gradation levels) • RGB-index chroma key (replaced with the specified color when the object color agrees with the preset value)
Internal video image format		• Total number of vertical lines: Up to 2048 lines • Number of valid vertical lines: 16 to 2044 lines (resolution: 1 line) • Vertical front porch: 2 lines (minimum) • Vertical back porch: 1 line (minimum) • Vertical pulse width: 1 line • Total number of horizontal pixels: Up to 2048 pixels • Number of valid horizontal pixels: 16 to 2040 pixels (resolution: 2 pixels) • Horizontal front porch: 3 pixels (minimum) (preliminary) • Horizontal back porch: 1 pixel (minimum) (preliminary) • Horizontal pulse width: 4 pixels

Table 56.2 Functional overview of GLCDC (2 of 2)

Parameter		Function
Data format conversion	Output video image size	• From 16 lines × 16 pixels to 2044 lines × 2040 pixels
	Data format	• RGB888 (parallel 24 bits) • RGB666 (parallel 18 bits) • RGB565 (parallel 16 bits) • RGB888 (serial 8 bits); clock cycle is four times the pixel clock • Bit endian order change and B/R signal swap
	Dither processing	• Reduces 10-bit signals (output after panel-oriented correction) to 8-, 6-, or 5-bit signals (output data format) • Supports the following modes: – Truncate mode – Round-off mode – 2 × 2 pattern dither mode
Timing control signal	Signal generation	5 timing signals (STVA, STVB, STHA, STHB, and DE) can be generated from HS / VS: • Vertical synchronization signal (variable) • Horizontal synchronization signal (variable) • Data enable signal
	Signal select	• Signals generated by the signal generation circuit can be output from LCD_TCONn pins (n = 0 to 3)
Output control panel-oriented correction processing	Brightness and contrast	• 10-bit internal processing; the sequence of this processing and gamma correction can be swapped. • Brightness: DC value adjustment range: from -512[LSB] to +512[LSB] • Contrast: gain value adjustment range: from 0 to 2 times (from 0/128 to 255/128)
	Gamma correction	• Sixteen areas; input/output: 10 bits • Gain value adjustment range in the area: From 0 to 2 times (from 0/1024 to 2047/1024)
Interrupts		• Number of specified lines interrupt (GLCDC_VPOS) • Graphics 1 buffer underflow interrupt (GLCDC_L1UNDF) • Graphics 2 buffer underflow interrupt (GLCDC_L2UNDF)
Module-stop function		Module-stop state can be set to reduce power consumption.
TrustZone Filter		Security and Privilege attribution can be set.

56.1.1 GLCDC Configuration

Figure 56.2 shows the configuration of the GLCDC.

The GLCDC includes the following blocks:

- Background screen generation block: Generates the background screen (including the blanking interval), selects the background color, and generates the synchronization signals for controlling the screens.
- Graphics data interface blocks (2 blocks): Convert the graphics data/CLUT data read through the GLCDC0 and GLCDC1 bus into ARGB (8888) data for internal processing, and transfer the clocks (PCLKA → PXCLK).
- Alpha blending blocks (2 blocks): Superimpose graphics data on the lower-layer screen and perform alpha blending based on the register settings and the alpha blending values for the current screen graphics data.
- TCON block: Generates the vertical and horizontal synchronization and enable signals suited for the specifications of the connected panel, from the internal vertical and horizontal synchronization signals.
- Data format conversion block: Processes data into the specific internal RGB888 format into the data of the specific format suited for the specifications of the panel with dither correction for output image data length.
- Panel-oriented correction processing block: Corrects brightness and contrast, and performs gamma correction suited for the characteristics of the connected panel, allowing either brightness and contrast correction or gamma correction to be performed first.
- Clock generator block: Generates the pixel and the panel clocks on a specific frequency from either LCD_EXTCLK or LCDCLK.

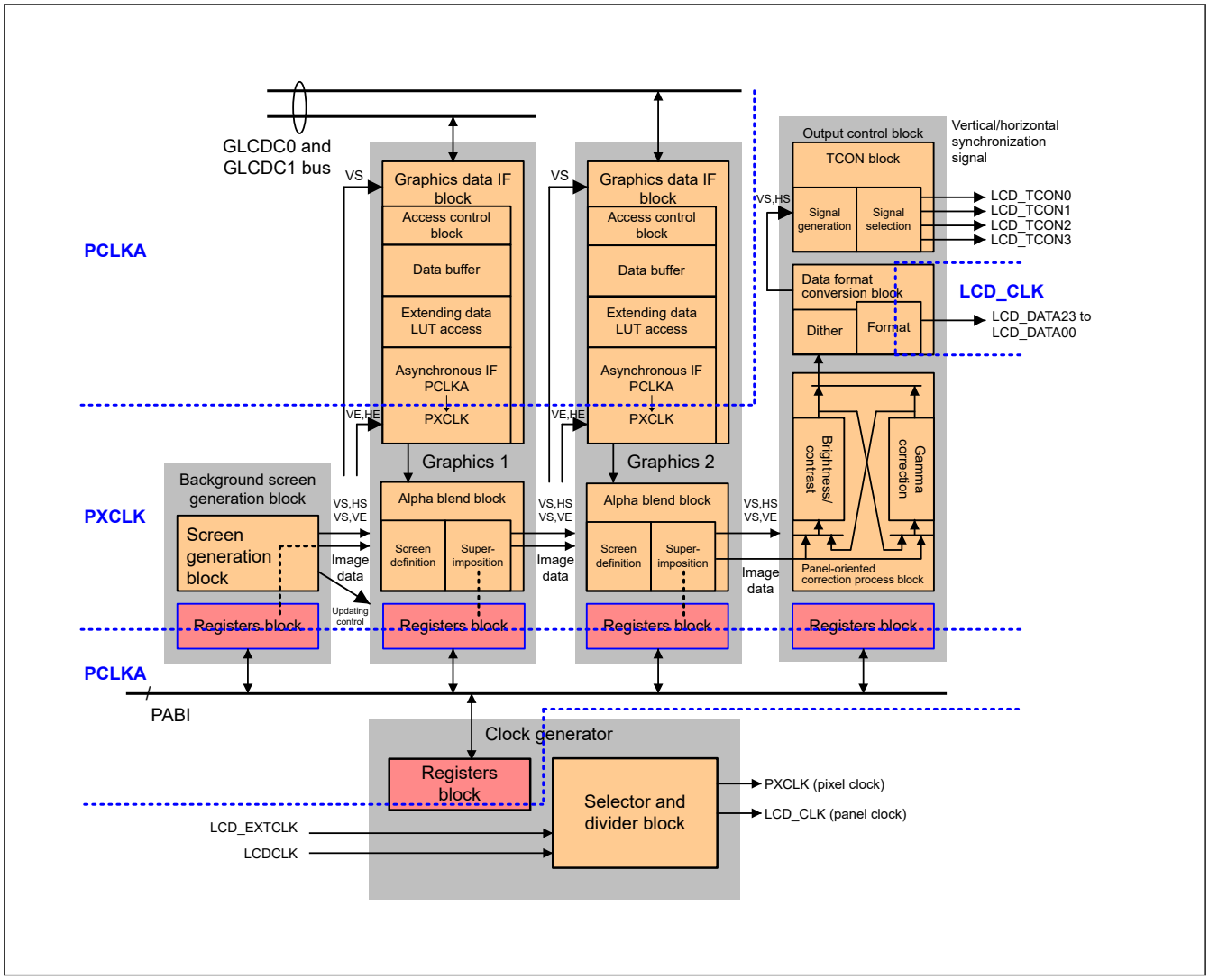


Figure 56.2 GLCDC configuration

56.1.2 Screen Format

Figure 56.3 shows the screen format overview of the GLCDC.

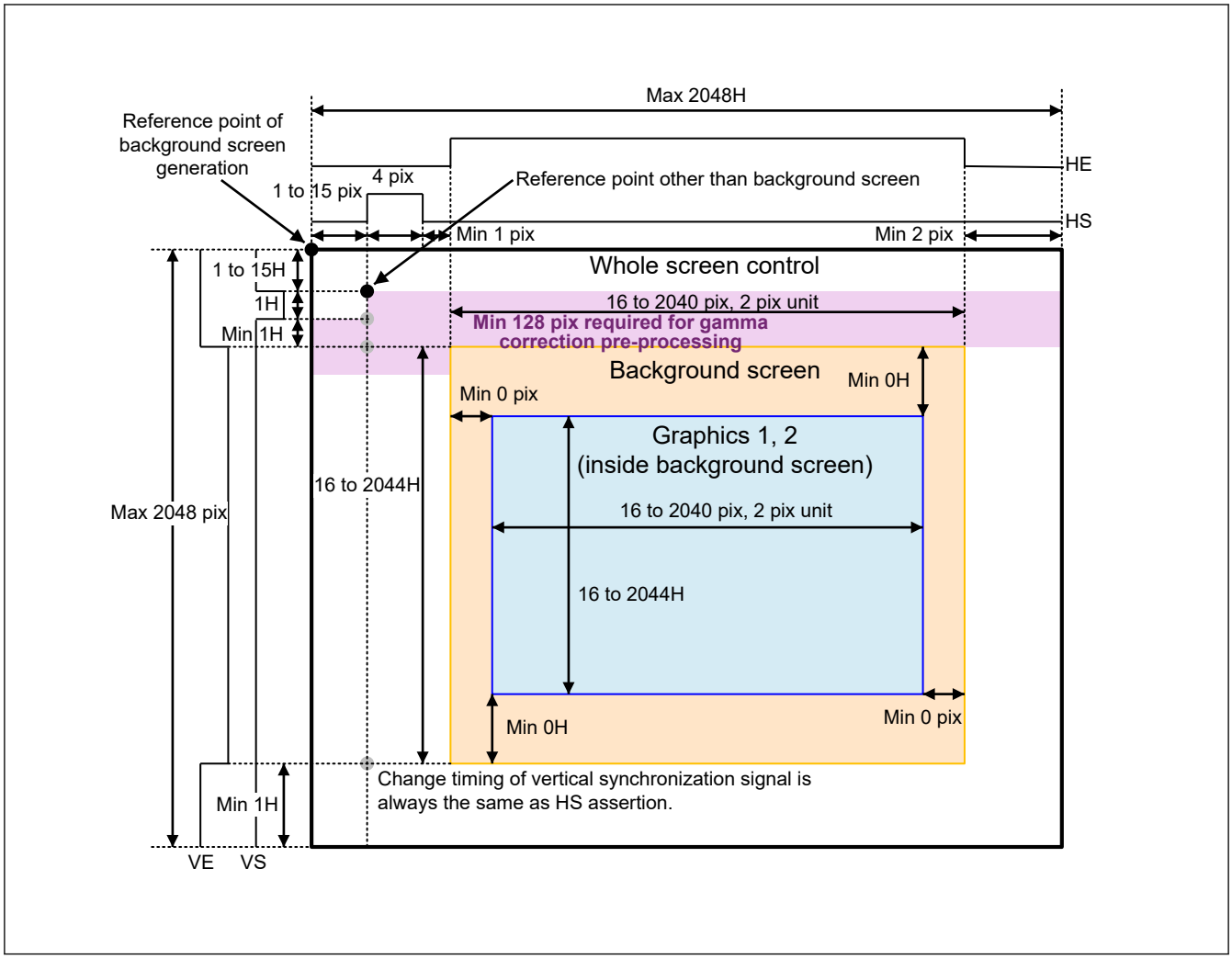


Figure 56.3 Screen format

The background screen generation block generates the essential timing signals for operations in the module as a whole. The graphics data interface blocks, alpha blending blocks, gamma correction block, output control block, and TCON block operate based on the synchronization and enable signals, which are sequentially transferred from the background screen generation block.

56.1.3 Graphics and Color Palette (CLUT) Data Formats

The GLCDC handles three display screens, one of which is a background screen. For the background screen, frame data is RGB888 graphics data stored in the registers, and for the other two screens, frame data is stored in the memory connected to the GLCDC0 and GLCDC1 bus as 32- or 16-bit/pixel graphics data or 8-, 4-, or 1-bit/pixel color palette (CLUT) data. The frame data of the relevant screen is read by the graphics data interface block, read into the GLCDC, and extended (converted) into ARGB8888 data for superimposition and blending. Figure 56.4 shows the frame data formats.

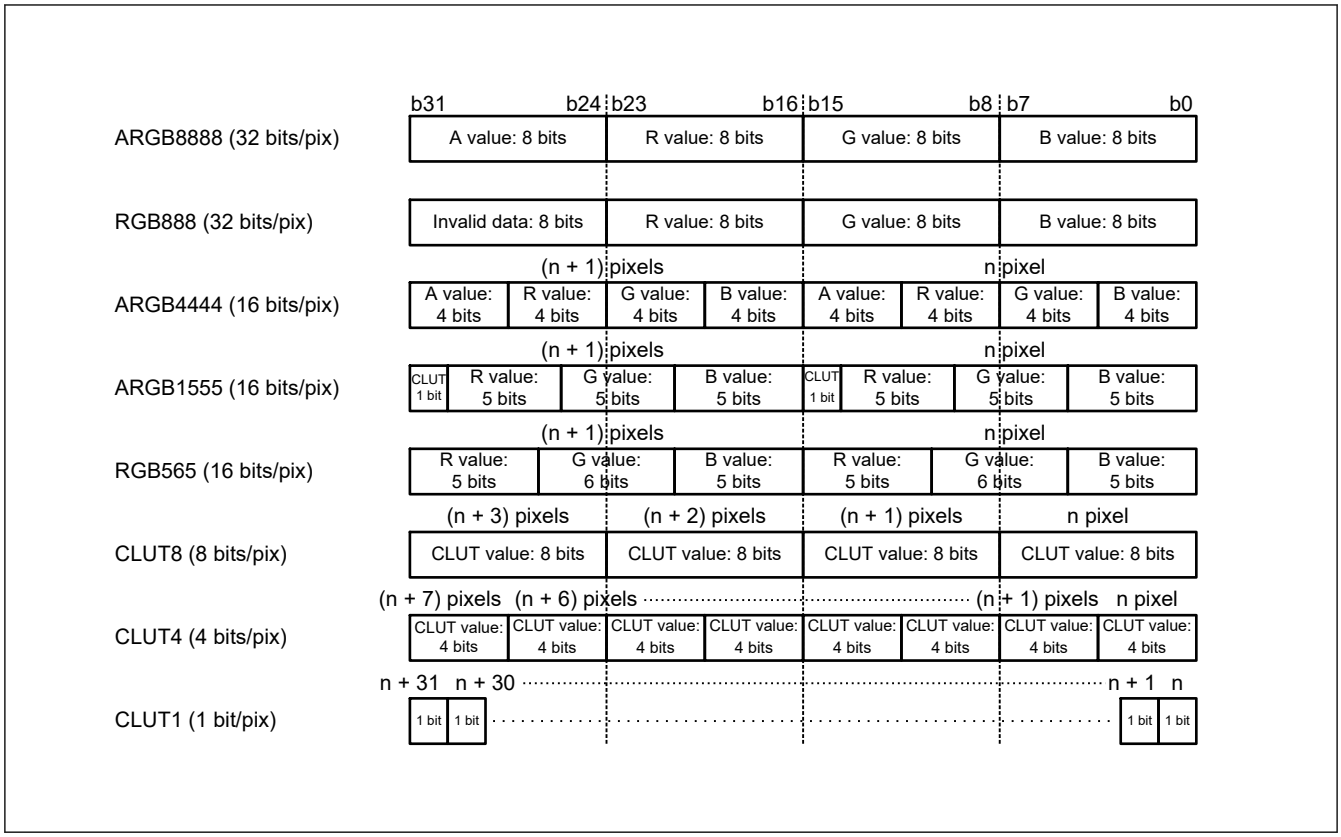


Figure 56.4 Frame data format

The A value (alpha blending value) represents the blending ratio of pixel data for displaying the lower-layer and current graphics screens after superimposition. The blending process is performed in accordance with the equations given in section 56.1.7. Graphics Data Interface. The CLUT value represents the color palette memory address (plane 0 or 1 is selected in the registers), and the data in the color palette is ARGB8888 (32 bits/pixel) graphics data. Addresses 0x01/0x00 are accessed with CLUT1, 0x0F to 0x00 with CLUT4, and 0xFF to 0x00 with CLUT8. Addresses 0x80/0x00 are accessed with ARGB1555 data.

56.1.4 Output Control for Data Format

The GLCDC can output data in accordance with the following formats and register settings:

- Data formats
  - Parallel: RGB888, RGB666, and RGB565
  - Serial: RGB888
  - Lower bits are processed in one of the following modes when 10-bit signals are reduced to 8-, 6-, and 5- bit signals:
    - Truncate mode
    - Round-off mode
    - 2×2 pattern dither processing + truncate mode
- Pixel arrangement
  - RGB
  - BGR
- Scan direction select for serial RGB888
  - Forward scan
  - Reverse scan
- Data output delay of serial RGB888

  - 0 to 3 clock cycles.

- Pin assignment
  - Little endian
  - Big endian

For details on dither processing, see section 56.2.48. OUT_PDTHA : Output Control Block Panel Dither Correction Register.

Bit assignment of LCD signals for parallel RGB888 format

Table 56.3 shows RGB signal inputs assigned to the LCD signal outputs for the parallel RGB888 output format.

Table 56.3 Bit assignment of RGB signal inputs for parallel RGB888 format

Pin assignment	Pixel arrangement			
	RGB, little endian	RGB, big endian	BGR, little endian	BGR, big endian
LCD_DATA23	R[7]	R[0]	B[7]	B[0]
LCD_DATA22	R[6]	R[1]	B[6]	B[1]
LCD_DATA21	R[5]	R[2]	B[5]	B[2]
LCD_DATA20	R[4]	R[3]	B[4]	B[3]
LCD_DATA19	R[3]	R[4]	B[3]	B[4]
LCD_DATA18	R[2]	R[5]	B[2]	B[5]
LCD_DATA17	R[1]	R[6]	B[1]	B[6]
LCD_DATA16	R[0]	R[7]	B[0]	B[7]
LCD_DATA15	G[7]	G[0]	G[7]	G[0]
LCD_DATA14	G[6]	G[1]	G[6]	G[1]
LCD_DATA13	G[5]	G[2]	G[5]	G[2]
LCD_DATA12	G[4]	G[3]	G[4]	G[3]
LCD_DATA11	G[3]	G[4]	G[3]	G[4]
LCD_DATA10	G[2]	G[5]	G[2]	G[5]
LCD_DATA09	G[1]	G[6]	G[1]	G[6]
LCD_DATA08	G[0]	G[7]	G[0]	G[7]
LCD_DATA07	B[7]	B[0]	R[7]	R[0]
LCD_DATA06	B[6]	B[1]	R[6]	R[1]
LCD_DATA05	B[5]	B[2]	R[5]	R[2]
LCD_DATA04	B[4]	B[3]	R[4]	R[3]
LCD_DATA03	B[3]	B[4]	R[3]	R[4]
LCD_DATA02	B[2]	B[5]	R[2]	R[5]
LCD_DATA01	B[1]	B[6]	R[1]	R[6]
LCD_DATA00	B[0]	B[7]	R[0]	R[7]

Note: R[7:0], G[7:0], and B[7:0] are RGB pixel data that is internally processed.

Bit assignment of LCD signals for parallel RGB666 format

Table 56.4 shows RGB signal inputs assigned to the LCD signal outputs for the parallel RGB666 output format.

Table 56.4 Bit assignment of RGB signal inputs for parallel RGB666 format (1 of 2)

Pin assignment	Pixel arrangement			
	RGB, little endian	RGB, big endian	BGR, little endian	BGR, big endian
LCD_DATA23	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0

Table 56.4 Bit assignment of RGB signal inputs for parallel RGB666 format (2 of 2)

Pin assignment	Pixel arrangement			
	RGB, little endian	RGB, big endian	BGR, little endian	BGR, big endian
LCD_DATA22	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA21	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA20	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA19	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA18	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA17	R[7]	R[2]	B[7]	B[2]
LCD_DATA16	R[6]	R[3]	B[6]	B[3]
LCD_DATA15	R[5]	R[4]	B[5]	B[4]
LCD_DATA14	R[4]	R[5]	B[4]	B[5]
LCD_DATA13	R[3]	R[6]	B[3]	B[6]
LCD_DATA12	R[2]	R[7]	B[2]	B[7]
LCD_DATA11	G[7]	G[2]	G[7]	G[2]
LCD_DATA10	G[6]	G[3]	G[6]	G[3]
LCD_DATA09	G[5]	G[4]	G[5]	G[4]
LCD_DATA08	G[4]	G[5]	G[4]	G[5]
LCD_DATA07	G[3]	G[6]	G[3]	G[6]
LCD_DATA06	G[2]	G[7]	G[2]	G[7]
LCD_DATA05	B[7]	B[2]	R[7]	R[2]
LCD_DATA04	B[6]	B[3]	R[6]	R[3]
LCD_DATA03	B[5]	B[4]	R[5]	R[4]
LCD_DATA02	B[4]	B[5]	R[4]	R[5]
LCD_DATA01	B[3]	B[6]	R[3]	R[6]
LCD_DATA00	B[2]	B[7]	R[2]	R[7]

Note: R[7:2], G[7:2], and B[7:2] are RGB pixel data that is internally processed.

Bit assignment of LCD signals for parallel RGB565 format

Table 56.5 shows RGB signal inputs assigned to the LCD signal outputs for the parallel RGB565 output format.

Table 56.5 Bit assignment of RGB signal inputs for parallel RGB565 format (1 of 2)

Pin assignment	Pixel arrangement			
	RGB, little endian	RGB, big endian	BGR, little endian	BGR, big endian
LCD_DATA23	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA22	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA21	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA20	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA19	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA18	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA17	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA16	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA15	R[7]	R[3]	B[7]	B[3]
LCD_DATA14	R[6]	R[4]	B[6]	B[4]
LCD_DATA13	R[5]	R[5]	B[5]	B[5]

Table 56.5 Bit assignment of RGB signal inputs for parallel RGB565 format (2 of 2)

Pin assignment	Pixel arrangement			
	RGB, little endian	RGB, big endian	BGR, little endian	BGR, big endian
LCD_DATA12	R[4]	R[6]	B[4]	B[6]
LCD_DATA11	R[3]	R[7]	B[3]	B[7]
LCD_DATA10	G[7]	G[2]	G[7]	G[2]
LCD_DATA09	G[6]	G[3]	G[6]	G[3]
LCD_DATA08	G[5]	G[4]	G[5]	G[4]
LCD_DATA07	G[4]	G[5]	G[4]	G[5]
LCD_DATA06	G[3]	G[6]	G[3]	G[6]
LCD_DATA05	G[2]	G[7]	G[2]	G[7]
LCD_DATA04	B[7]	B[3]	R[7]	R[3]
LCD_DATA03	B[6]	B[4]	R[6]	R[4]
LCD_DATA02	B[5]	B[5]	R[5]	R[5]
LCD_DATA01	B[4]	B[6]	R[4]	R[6]
LCD_DATA00	B[3]	B[7]	R[3]	R[7]

Note: R[7:3], G[7:2], and B[7:3] are RGB pixel data that is internally processed.

Bit assignment of RGB signal inputs for serial RGB888 format

Table 56.6 and Table 56.7 show RGB signal inputs assigned to the LCD signal outputs for the serial RGB888 output format.

Table 56.6 Bit assignment of RGB signal inputs for serial RGB888 format, RGB arrangement

Pin assignment	Pixel arrangement, scan direction select, and cycle															
	RGB, little endian								RGB, big endian							
	Reverse scan				Forward scan				Reverse scan				Forward scan			
	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th
LCD_DATA23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
:	:				:				:				:			
LCD_DATA08	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LCD_DATA07	Undefined	B[7]	G[7]	R[7]	R[7]	G[7]	B[7]	Undefined	Undefined	B[0]	G[0]	R[0]	R[0]	G[0]	B[0]	Undefined
LCD_DATA06	Undefined	B[6]	G[6]	R[6]	R[6]	G[6]	B[6]	Undefined	Undefined	B[1]	G[1]	R[1]	R[1]	G[1]	B[1]	Undefined
LCD_DATA05	Undefined	B[5]	G[5]	R[5]	R[5]	G[5]	B[5]	Undefined	Undefined	B[2]	G[2]	R[2]	R[2]	G[2]	B[2]	Undefined
LCD_DATA04	Undefined	B[4]	G[4]	R[4]	R[4]	G[4]	B[4]	Undefined	Undefined	B[3]	G[3]	R[3]	R[3]	G[3]	B[3]	Undefined
LCD_DATA03	Undefined	B[3]	G[3]	R[3]	R[3]	G[3]	B[3]	Undefined	Undefined	B[4]	G[4]	R[4]	R[4]	G[4]	B[4]	Undefined
LCD_DATA02	Undefined	B[2]	G[2]	R[2]	R[2]	G[2]	B[2]	Undefined	Undefined	B[5]	G[5]	R[5]	R[5]	G[5]	B[5]	Undefined
LCD_DATA01	Undefined	B[1]	G[1]	R[1]	R[1]	G[1]	B[1]	Undefined	Undefined	B[6]	G[6]	R[6]	R[6]	G[6]	B[6]	Undefined
LCD_DATA00	Undefined	B[0]	G[0]	R[0]	R[0]	G[0]	B[0]	Undefined	Undefined	B[7]	G[7]	R[7]	R[7]	G[7]	B[7]	Undefined

Note: R[7:0], G[7:0], and B[7:0] are RGB pixel data that is internally processed.

Table 56.7 Bit assignment of RGB signal inputs for serial RGB888 format, BGR arrangement (1 of 2)

Pin assignment	Pixel arrangement, scan direction select, and cycle															
	BGR, little endian								BGR, big endian							
	Reverse scan				Forward scan				Reverse scan				Forward scan			
	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th
LCD_DATA23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
:	:				:				:				:			

Table 56.7 Bit assignment of RGB signal inputs for serial RGB888 format, BGR arrangement (2 of 2)

Pin assignment	Pixel arrangement, scan direction select, and cycle															
	BGR, little endian								BGR, big endian							
	Reverse scan				Forward scan				Reverse scan				Forward scan			
	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th
LCD_DATA08	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LCD_DATA07	Undefined	R[7]	G[7]	B[7]	B[7]	G[7]	R[7]	Undefined	Undefined	R[0]	G[0]	B[0]	B[0]	G[0]	R[0]	Undefined
LCD_DATA06	Undefined	R[6]	G[6]	B[6]	B[6]	G[6]	R[6]	Undefined	Undefined	R[1]	G[1]	B[1]	B[1]	G[1]	R[1]	Undefined
LCD_DATA05	Undefined	R[5]	G[5]	B[5]	B[5]	G[5]	R[5]	Undefined	Undefined	R[2]	G[2]	B[2]	B[2]	G[2]	R[2]	Undefined
LCD_DATA04	Undefined	R[4]	G[4]	B[4]	B[4]	G[4]	R[4]	Undefined	Undefined	R[3]	G[3]	B[3]	B[3]	G[3]	R[3]	Undefined
LCD_DATA03	Undefined	R[3]	G[3]	B[3]	B[3]	G[3]	R[3]	Undefined	Undefined	R[4]	G[4]	B[4]	B[4]	G[4]	R[4]	Undefined
LCD_DATA02	Undefined	R[2]	G[2]	B[2]	B[2]	G[2]	R[2]	Undefined	Undefined	R[5]	G[5]	B[5]	B[5]	G[5]	R[5]	Undefined
LCD_DATA01	Undefined	R[1]	G[1]	B[1]	B[1]	G[1]	R[1]	Undefined	Undefined	R[6]	G[6]	B[6]	B[6]	G[6]	R[6]	Undefined
LCD_DATA00	Undefined	R[0]	G[0]	B[0]	B[0]	G[0]	R[0]	Undefined	Undefined	R[7]	G[7]	B[7]	B[7]	G[7]	R[7]	Undefined

Note: R[7:0], G[7:0], and B[7:0] are RGB pixel data that is internally processed.

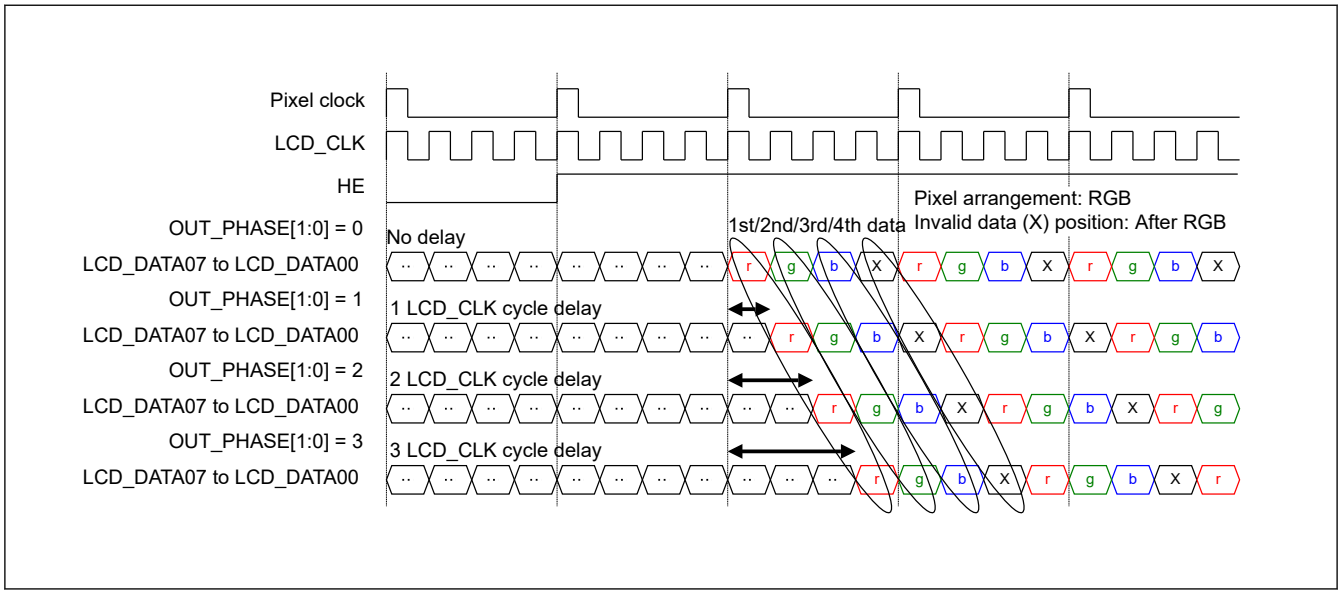


Figure 56.5 Serial RGB888 (4x speed) output timing

56.1.5 Output Control for Panel-Oriented Correction Process

The following panel-oriented correction processes are provided:

- Brightness correction
- Contrast correction
- RGB gamma correction

Brightness correction always precedes contrast correction, but RGB gamma correction can either precede or follow brightness and contrast correction, based on the register settings. In panel-oriented correction, 10-bit RGB data obtained by extending 8-bit RGB data output from graphics 2 is used, and 10-bit RGB data is also output to the output control (data format conversion) block. Figure 56.6 shows the configuration of the panel-oriented correction circuit.

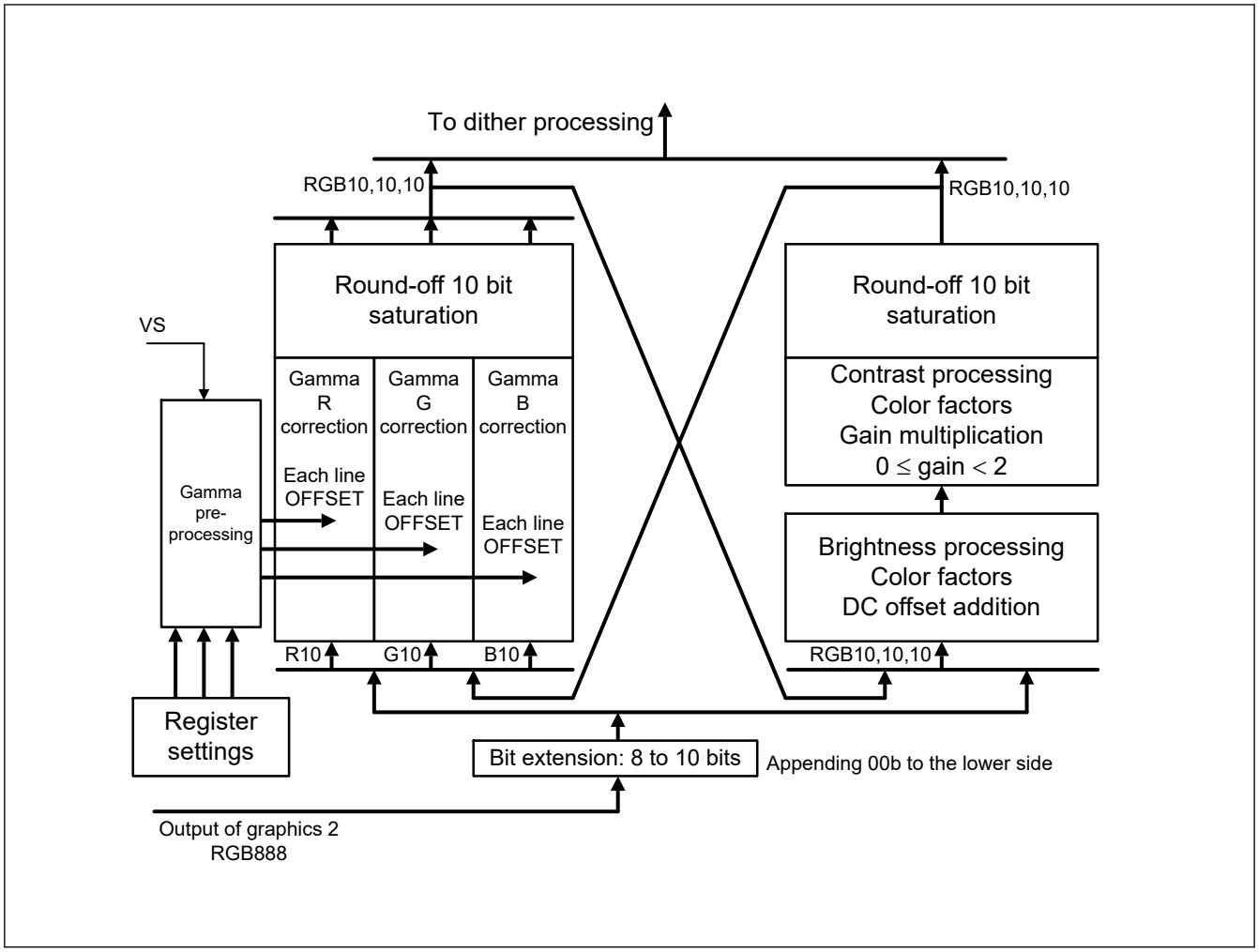


Figure 56.6 Configuration of panel-oriented correction circuit

56.1.6 Output Control for TCON

Any of the four signals generated from the internal vertical and horizontal synchronization signals (STVA, STVB, STHA, and STHB) that have passed through the data format conversion block can be selected for output on four pins, LCD_TCON0, LCD_TCON1, LCD_TCON2, and LCD_TCON3. The generated signals are completely independent of the image data. They are not affected by the output image format or any internal data process, and no register settings for signal generation affect the output image format or any internal data process. The data enable signal DE, which is to be generated by the TCON block, is the logical AND of the two signals STVB and STHB, which were previously generated by the TCON block. Consequently, three signals in total can actually be generated if DE is to be output.

56.1.7 Graphics Data Interface

Two circuit systems are provided for reading graphics data (graphics 1 and 2), each of which incorporates an access control block, data buffer, CLUT memory, data extension block, and asynchronous interface block.

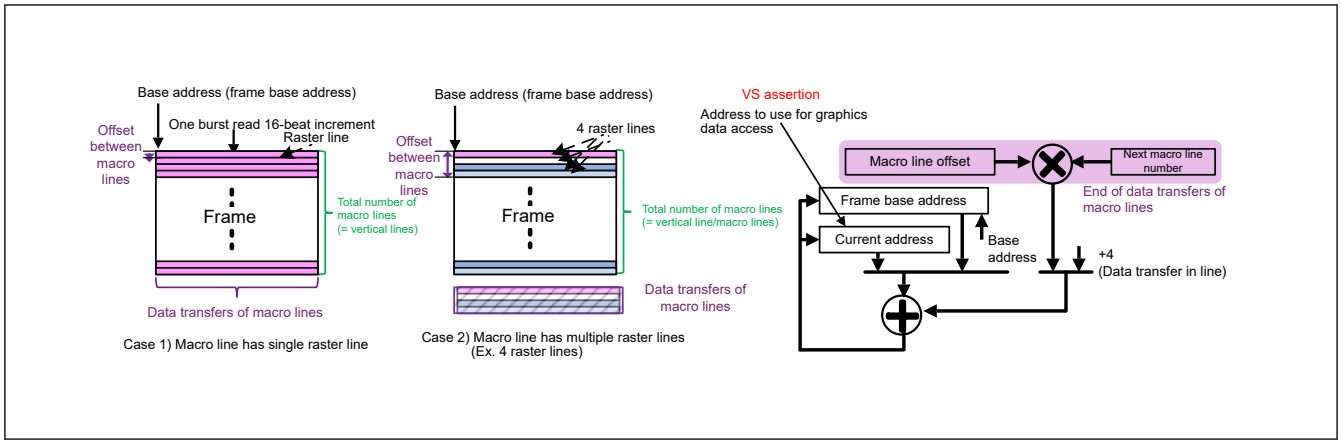


Figure 56.7 Calculation of graphics data access address

Graphics (or CLUT) data is accessed and output to the pixel operation block as ARGB8888 data (32 bits/pixel).

The GLCDC0 and GLCDC1 bus is accessed in word (32-bit) units in 16-beat increment burst reads, in accordance with the preset parameters in the two-dimensional addressing mode, in which the macro line structure is accounted for as shown in Figure 56.7, and data is stored in the data buffer. Even if invalid data is at the raster line end, all the data is stored in the data buffer, and the invalid data is skipped when data is read internally. Regardless of the format of the graphics data, data is extended to ARGB8888 data before being output to the alpha blending blocks.

56.1.8 Blending

The following processes are performed for the graphics areas specified in the registers. The lower-layer graphics plane is output without any processing to the display area outside the graphics area.

- Display plane selection
- RGB-index chroma-key
- Alpha blending

Figure 56.8 shows the relationship between the graphics display selection and rectangular alpha blending area.

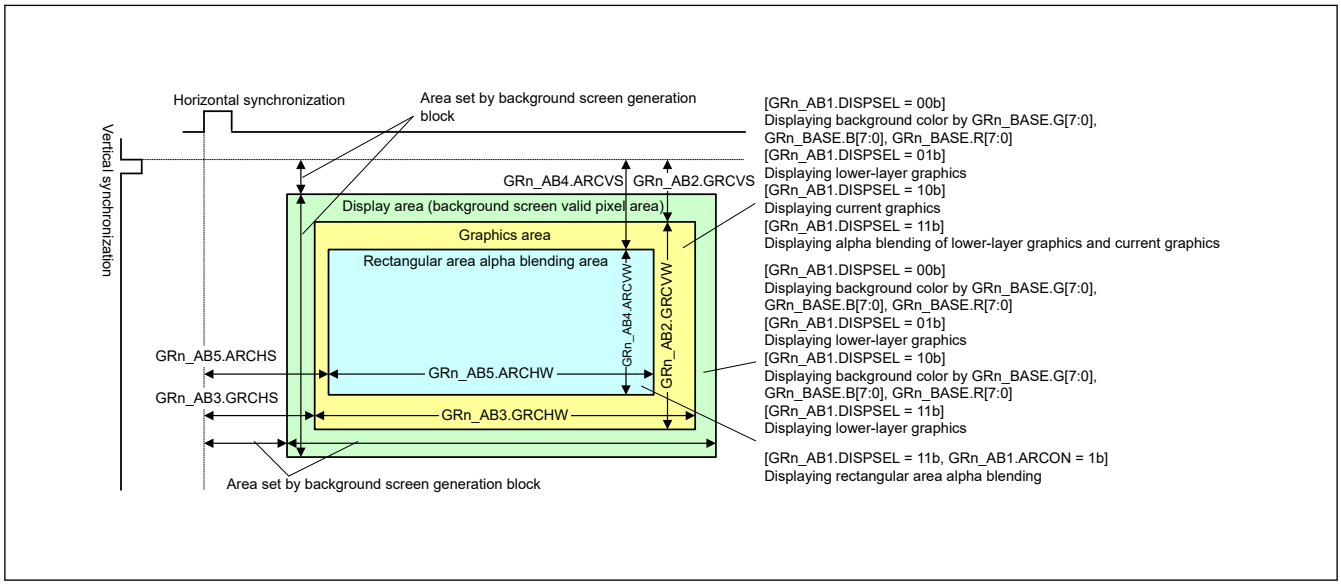


Figure 56.8 Graphics display selection

(1) Display plane selection

Based on the register settings, the following data is output to the graphics area:

- Background color:

  - RGB data specified in the registers.

- Lower-layer graphics:
  - RGB data input from the previous module
  - Output from the background screen generation block for graphics 1
  - Output from graphics 1 for graphics 2

- Current graphics:
  - RGB data obtained by the graphics data interface block extending the graphics data read by the graphics 1 or 2 module through the GLCDC0 and GLCDC1 bus or CLUT data to ARGB8888 format data.

- Blending:
  - RGB data obtained by blending the lower-layer graphics data and current graphics data obtained by the graphics data interface block extension to ARGB8888 format data, based on the alpha blending values or the register settings.

It is possible to modify the register values related to these functions while display operation is in progress, and to allow reflection of the updates to the internal operations when the VS (vertical synchronization signal) is asserted, if reflection of the register settings to the internal operations on vertical synchronization is enabled.

(2) RGB-index chroma-key

If the RGB value of the ARGB8888 data input from the graphics data interface block agrees with the value set in the GRn_AB8 register, the ARGB8888 data is entirely replaced with the value preset in the GRn_AB9 register. All the ARGB8888 data input from the graphics data interface block is subject to this process. If your application excludes CLUT memory output from this process, you must disable the corresponding process in GRn_AB7.CKON.

(3) Alpha blending

If blending is selected in the selected display plane, lower-layer graphics and current graphics are alpha-blended based on the register settings using either of the following two functions.

Alpha-blending in a rectangular area

The following process is performed for the rectangular area (in the graphics area) preset in the registers:

1. Lower-layer graphics and current graphics planes are blended in accordance with the A value set in the registers.
2. After the number of frames set in the registers passes, the A value is updated to the A value + Δ (register setting).
3. The lower-layer graphics and current graphics planes are blended in accordance with this updated A value.
4. The process of updating the A value after the number of frames set in the registers passes is repeated (A value: min/max value saturation).

Alpha-blending in pixel units

The lower-layer graphics and current graphics planes are blended in accordance with the A value of the ARGB8888 data input from the graphics data interface block.

Figure 56.9 shows the update of the alpha blending value in the rectangular alpha blending area.

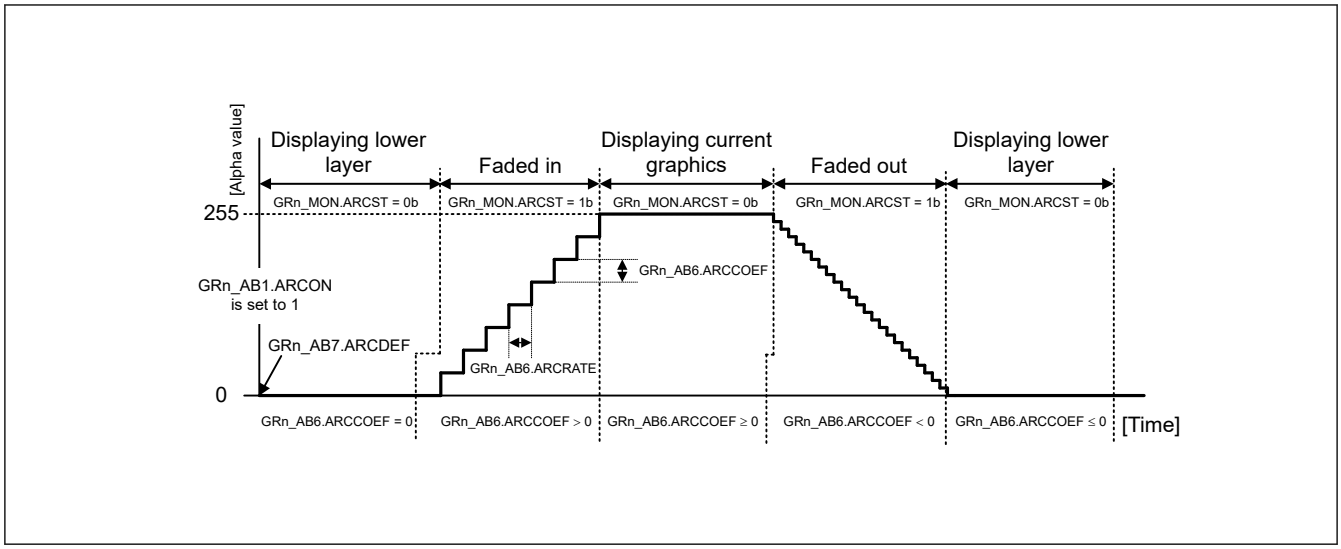


Figure 56.9 Updating of alpha blending value

Alpha blending is based on the following formulas:

When A value = 255

Rout/Gout/Bout = current graphics data

When A value ≠ 255

Rout = (Rin1 × A + Rin0 × (256 - A))/256

Gout = (Gin1 × A + Gin0 × (256 - A))/256

Bout = (Bin1 × A + Bin0 × (256 - A))/256

where,

A: alpha blending value

Rin1/Gin1/Bin1: current graphics data

Rin0/Gin0/Bin0: lower-layer graphics data

56.2 Register Descriptions

56.2.1 GRm_CLUTn[y] : Color Palette (m = 1 to 2, n = 0 to 1, y = 0 to 255)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x0+0x4 × y (GR1_CLUT0[y])
0x0400+0x4 × y (GR1_CLUT1[y])
0x0800+0x4 × y (GR2_CLUT0[y])
0x0C00+0x4 × y (GR2_CLUT1[y])

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	A[7:0]								R[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	G[7:0]								B[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	B[7:0]	B Value of Color Palette n Plane for Graphics m Plane B value of color palette n plane for graphics m plane. Unsigned 8-bit integer.	R/W

Bit	Symbol	Function	R/W
15:8	G[7:0]	G Value of Color Palette n Plane for Graphics m Plane G value of color palette n plane for graphics m plane. Unsigned 8-bit integer.	R/W
23:16	R[7:0]	R Value of Color Palette n Plane for Graphics m Plane R value of color palette n plane for graphics m plane. Unsigned 8-bit integer.	R/W
31:24	A[7:0]	Alpha Blending Value of Color Palette n Plane for Graphics m Plane Alpha blending value of color palette n plane for graphics m plane. Unsigned 8-bit integer.	R/W

Note: S-TYPE-3, P-TYPE-3

B[7:0] bits (B Value of Color Palette n Plane for Graphics m Plane)

The B[7:0] bits set the B value when this color palette is used.

G[7:0] bits (G Value of Color Palette n Plane for Graphics m Plane)

The G[7:0] bits set the G value when this color palette is used.

R[7:0] bits (R Value of Color Palette n Plane for Graphics m Plane)

The R[7:0] bits set the R value when this color palette is used.

A[7:0] bits (Alpha Blending Value of Color Palette n Plane for Graphics m Plane)

The A[7:0] bits set the alpha blending value when this color palette is used.

All the planes can always be accessed through the register access bus (PABI), regardless of the plane specified to be used by the graphics data access block. The updates are reflected to the internal operations directly, not in synchronization with the vertical synchronization signal. To keep reflection of the CLUT memory contents to the internal operations in synchronization with the vertical synchronization signal, first write data through the register access bus to the plane that is not being used for internal operations, and then modify the GRn_CLUTINT.SEL bit for controlling the plane that is to be used.

56.2.2 BG_EN : Background Plane Setting Operation Control Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	VEN	—	—	—	—	—	—	—	EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EN	Background Plane Operation Enable 0: Disable background plane operation 1: Enable background plane operation	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	VEN	Control of GLCDC Internal Register Value Reflection to Internal Operations This bit is set to 0 by an internal source. 0: Disable GLCDC register values from being reflected in internal operations at start of screen generation 1: Enable GLCDC register values to be reflected in internal operations at start of screen generation	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	SWRST	Software Reset Control 0: Place entire module in software reset state 1: Release entire module from software reset state	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

EN bit (Background Plane Operation Enable)

The EN bit enables or disables the operation of the background plane generation module. When the operation is stopped (this bit is cleared to 0) after the operation is enabled (set to 1), the operation is stopped at the end of screen generation, unlike when a software reset occurs.

When setting this bit to 1, set the BG_EN.VEN bit to 1 simultaneously to enable register value reflection to internal operations. When clearing this bit to 0, also clear the BG_EN.VEN bit to 0 simultaneously. Before clearing this bit to 0, confirm that the BG_MON.VEN bit is cleared to 0, to make sure that the signal for controlling reflection of the register values to internal operations is negated. Operation is not guaranteed if this bit and the BG_EN.VEN bit are set to 1, or if the settings in other registers are modified before the BG_MON.EN bit is cleared to 0.

VEN bit (Control of GLCDC Internal Register Value Reflection to Internal Operations)

The VEN bit enables or disables reflection of the GLCDC internal register to the GLCDC internal operation on assertion of the vertical synchronization signal (input). When this bit is set to 1, the signal of GLCDC internal register values reflection control is set to 1 at the immediate start of the screen, and are automatically cleared to 0 at the end of the vertical valid pixel of the same screen. Set this bit to 1 only when 0. Operation is not guaranteed if this bit is set to 1 when 1. Also, while this bit is 1, do not modify any register whose value is reflected to the internal operations at the start of the screen in the GLCDC or on assertion of the vertical synchronization signal (input). Otherwise, operation is not guaranteed.

SWRST bit (Software Reset Control)

The SWRST bit controls a software reset of the entire GLCDC, not only the background plane generation module. When this bit is cleared to 0, the GLCDC enters the reset state from any operation state. This bit must be set to 1 before the registers are set or operation enabled. Although the registers (except the CLUT memory and the some of the operation control registers) can be set while this bit is set to 1 (immediately after), before accessing the CLUT memory, enabling operation, or reflecting the register values to the internal operation by the vertical synchronization signal, confirm that PXCLK/LCD_CLK and PCLKA are supplied and that the BG_MON.SWRST bit, which monitors the entire module software reset state, is set to 1. Operation is not guaranteed if the software accesses CLUT memory, enables operation, or reflects the register vales to the internal operation on the vertical synchronization signal while PXCLK/LCD_CLK and PCLKA are not supplied or the BG_MON.SWRST bit is not set to 1. The peripheral bus clock (PCLKA) must be supplied to the GLCDC when this bit is used to apply or cancel a software reset. If PCLKA is not supplied, writing to this bit is impossible.

The bits in this register control the GLCDC states. The internal states can be read from the associated bits in the status monitor registers and BG_EN.VEN (this register). Because the GLCDC, which operates on multiple clock signals, requires a certain period for state transition, you must confirm that the internal state has stabilized (state transition is complete) when settings are modified. Operation is not guaranteed if settings are modified again before the internal state stabilizes (state transition is complete). However, clearing a software reset to 0 immediately makes the whole GLCDC reset, setting it to 1 releases the reset state. These operations do not require the clock supply of PXCLK.

56.2.3 BG_PERI : Background Plane Setting Free-Running Period Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	FV[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	FH[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

Bit	Symbol	Function	R/W
10:0	FH[10:0]	Background Plane Horizontal Synchronization Signal Period Period based on pixel clocks (PXCLK). The valid range is 0x17 to 0x7FF. 0x017: 24 cycles (pixels) 0x3FF: 1024 cycles (pixels) 0x7FF: 2048 cycles (pixels)	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
26:16	FV[10:0]	Background Plane Vertical Synchronization Signal Period Period based on lines. The valid range is 0x013 to 0x7FF. 0x013: 20 lines 0x3FF: 1024 lines 0x7FF: 2048 lines	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

FH[10:0] bits (Background Plane Horizontal Synchronization Signal Period)

The FH[10:0] bits set the horizontal synchronization signal period of the background plane. This field contains 11 bits and can be set to any number from 0x000 to 0x7FF. However, the valid range is 0x017 to 0x7FF.

FV[10:0] bits (Background Plane Vertical Synchronization Signal Period)

The FV[10:0] bits set the vertical synchronization signal period of the background plane. This field contains 11 bits and can be set to any number from 0x000 to 0x7FF. However, the valid range is 0x013 to 0x7FF.

56.2.4 BG_SYNC : Background Plane Setting Synchronization Position Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	VP[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	HP[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	HP[3:0]	Background Plane Horizontal Synchronization Signal Assertion Position Position based on pixel clocks (PXCLK). 0x0: Setting prohibited Others: HPth cycle (pixel)	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
19:16	VP[3:0]	Background Plane Vertical Synchronization Assertion Position Position based on lines. 0x0: Setting prohibited Others: VPth line	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

HP[3:0] bits (Background Plane Horizontal Synchronization Signal Assertion Position)

The HP[3:0] bits set the horizontal synchronization signal assertion position of the background plane. The signal is held asserted for a 4-pixel clock width.

VP[3:0] bits (Background Plane Vertical Synchronization Assertion Position)

The VP[3:0] bits set the vertical synchronization signal assertion position of the background plane. The signal is held asserted for a 1H width, and the assertion timing within a single horizontal line is specified in BG_SYNC.HP[3:0].

56.2.5 BG_VSIZE : Background Plane Setting Full Image Vertical Size Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x100C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	VP[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	VW[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
10:0	VW[10:0]	Background Plane Vertical Valid Pixel Width Width based on lines. 0x010: 16 lines ⋮ 0x7FC: 2044 lines Others: Setting prohibited	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
26:16	VP[10:0]	Background Plane Vertical Valid Pixel Start Position Position based on of lines. 0x003: 3rd line ⋮ 0x7EF: 2031 lines Others: Setting prohibited	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

VW[10:0] bits (Background Plane Vertical Valid Pixel Width)

The VW[10:0] bits set the vertical valid pixel width of the background plane. This field contains 11 bits and can be set to any number from 0x000 to 0x7FF. However, the valid range is 0x010 to 0x7FC. Operation is not guaranteed if a value outside the valid range is set.

VP[10:0] bits (Background Plane Vertical Valid Pixel Start Position)

The VP[10:0] bits set the vertical valid pixel start position of the background plane. This field contains 11 bits and can be set to any number from 0x000 to 0x7FF. However, the valid range is 0x003 to 0x7EF. Operation is not guaranteed if a value outside the valid range is set.

Specify the vertical valid pixel area between the assertion position of the vertical synchronization signal + 2 and the (background plane end - 1)th line. Operation is not guaranteed if the area is specified outside this range.

56.2.6 BG_HSIZE : Background Plane Setting Full Image Horizontal Size Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	HP[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	HW[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
10:0	HW[10:0]	Background Plane Horizontal Valid Pixel Width Width based on pixel clocks (PXCLK). 0x010: 16 cycles ⋮ 0x7F8: 2040 cycles Others: Setting prohibited	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
26:16	HP[10:0]	Background Plane Horizontal Valid Pixel Start Position Position based on pixel clocks (PXCLK). 0x006: 6th cycle (pixel) ⋮ 0x7EE: 2030 cycles Others: Setting prohibited	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

HW[10:0] bits (Background Plane Horizontal Valid Pixel Width)

The HW[10:0] bits set the horizontal valid pixel width of the background plane. This field contains 11 bits and can be set to any number from 0x000 to 0x7FF. However, the valid range is 0x010 to 0x7F8. Operation is not guaranteed if a value outside the valid range is set. When serial RGB is selected as the output format for the output control block, add two to the horizontal valid pixel width and set the resulting value to these bits.

HP[10:0] bits (Background Plane Horizontal Valid Pixel Start Position)

The HP[10:0] bits set the horizontal valid pixel start position of the background plane. This field contains 11 bits and can be set to any number from 0x000 to 0x7FF. However, the valid range is 0x006 to 0x7EE. Operation is not guaranteed if a value outside the valid range is set.

Specify the horizontal valid pixel area between the assertion position of the horizontal synchronization signal + 5 and pixel number (line end – 2). Operation is not guaranteed if the area is specified outside this range.

The background plane generation module outputs the values that are specified in the Background Color Register (BG_BGC) for the area defined by the Full Image Vertical Size Register (BG_VSIZE) and Full Image Horizontal Size Register (BG_HSIZE), and it outputs 0x00 as the RGB values for the blanking interval area.

56.2.7 BG_BGC : Background Plane Setting Background Color Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	R[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	G[7:0]								B[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	B[7:0]	Background Plane Valid Pixel Area B Value B value for background plane valid pixel area. Unsigned 8-bit integer.	R/W
15:8	G[7:0]	Background Plane Valid Pixel Area G Value G value for background plane valid pixel area. Unsigned 8-bit integer.	R/W
23:16	R[7:0]	Background Plane Valid Pixel Area R Value R value for background plane valid pixel area. Unsigned 8-bit integer.	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

B[7:0] bits (Background Plane Valid Pixel Area B Value)

The B[7:0] bits set the B value of image data to be output to the valid pixel area of the background plane.

G[7:0] bits (Background Plane Valid Pixel Area G Value)

The G[7:0] bits set the G value of image data to be output to the valid pixel area of the background plane.

R[7:0] bits (Background Plane Valid Pixel Area R Value)

The R[7:0] bits set the R value of image data to be output to the valid pixel area of the background plane.

56.2.8 BG_MON : Background Plane Setting Status Monitor Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	VEN	—	—	—	—	—	—	—	EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EN	Background Plane Operation Monitor 0: Operation is stopped 1: Operation is in progress	R
7:1	—	These bits are read as 0. The write value should be 0.	R
8	VEN	Entire Module Internal Operation Reflection Control Signal Monitor 0: Signal for controlling reflection of the register values to internal operations on assertion of vertical synchronization signal is negated 1: Signal for controlling reflection of the register values to internal operations on assertion of vertical synchronization signal is asserted	R
15:9	—	These bits are read as 0.	R
16	SWRST	Entire Module SW Reset State Monitor 0: Entire module is in software reset state 1: Entire module is released from software reset state	R
31:17	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

EN bit (Background Plane Operation Monitor)

The EN bit indicates whether the background plane generation module is operating or not. To stop the operation of the background plane generation module by clearing BG_EN.EN to 0, read this bit to confirm that the operation of the background plane is complete. Clearing BG_EN.EN to 0 does not stop the operation until completion of the background plane, unlike when BG_EN.SWRST is cleared to 0.

VEN bit (Entire Module Internal Operation Reflection Control Signal Monitor)

The VEN bit indicates the value of the signal for controlling reflection of the GLCDC internal register values to the internal operations. This signal is asserted at the start of a screen immediately after setting BG_EN.VEN to 1 and negated at the VE negate timing output from the background screen generation block.

SWRST bit (Entire Module SW Reset State Monitor)

The SWRST bit indicates the software reset state of the entire module when PXCLK is supplied. This bit value indicates the result of the peripheral module clock A (PCLKA) resampling the result of the pixel clock (PXCLK) sampling the BG_EN.SWRST bit. Even if PXCLK is not supplied, clearing the BG_EN.SWRST to 0 clears this bit to 0.

56.2.9 GRn_VEN : Graphics n Register Update Control Register (n = 1 to 2)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1100 + 0x100 × (n-1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PVEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PVEN	Control of Graphics n Register Value Reflection to Internal Pixel Operations This bit is cleared to 0 by an internal source. 0: Disable reflection of register values to internal operations on assertion of vertical synchronization signal (VS) 1: Enable reflection of register values to internal operations on assertion of the vertical synchronization signal (VS)	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

PVEN bit (Control of Graphics n Register Value Reflection to Internal Pixel Operations)

The PVEN bit enables or disables reflection of the register values to the internal operations of the pixel operation circuit on assertion of the vertical synchronization signal (input). When this bit is set to 1, the register values are immediately reflected to the internal operations on assertion of the vertical synchronization signal (input), and are automatically cleared to 0. Set this bit to 1 only when 0. Operation is not guaranteed if this bit is set to 1 when 1. Also, if the signal output is asserted from the background plane generation module that controls reflection of the register values to the internal operations of all the modules, the register values are always reflected to the internal operations on assertion of the vertical synchronization signal (input), regardless of the value of this bit. While this bit is 1, do not modify any register whose value is reflected to the internal operations on assertion of the vertical synchronization signal (input) in this block. Otherwise, operation is not guaranteed.

This bit must not be 1 at the same time as the BG_EN.VEN bit (control of background plane register value reflection to internal operations) in the Operation Control Register (BG_EN), one of the background plane setting registers. Otherwise, operation is not guaranteed.

56.2.10 GRn_FLMRD : Graphics n Frame Buffer Read Control Register (n = 1, 2)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1104 + 0x100 × (n -1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RENB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RENB	Graphics Data Read Enable Graphics data is the frame buffer data. 0: Disable reading 1: Enable reading	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

RENB bit (Graphics Data Read Enable)

The RENB bit enables or disables reading of the graphics data (frame buffer data in memory connected through the GLCDC0 and GLCDC1 bus). If the current graphics data is required, data read must be enabled (GRn_FLMRD.RENB = 1) before the background panel operation is enabled, register value internal operation reflection control is enabled, or the register value internal operation reflection control of graphics 1 and 2 is enabled.

If the current graphics data is not required, data read must be disabled (GRn_FLMRD.RENB = 0) before the background panel operation is enabled, register value internal operation reflection control is enabled, or the register value internal operation reflection control of graphics 1 and 2 is enabled.

56.2.11 GRn_FLM1 : Graphics n Frame Buffer Control Register 1 (n = 1 to 2)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1108 + 0x100 × (n -1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BSTMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
1:0	BSTMD[1:0]	Burst Transfer Control for Graphics Data Access Graphics data is the frame buffer data. 1 1: 16-beat increment burst transfer (64-byte boundary) Others: Setting prohibited	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

BSTMD[1:0] bits (Burst Transfer Control for Graphics Data Access)

The BSTMD[1:0] bits control burst transfers for accessing the graphics data (frame buffer data in memory connected to memory through the GLCDC0 and GLCDC1 bus). In this GLCDC, these bits are fixed to 11b. Operation is not guaranteed if these bits are set to any other value.

56.2.12 GRn_FLM2 : Graphics n Frame Buffer Control Register 2 (n = 1 to 2)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x110C + 0x100 × (n -1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BASE[31:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BASE[31:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	BASE[31:0]	Base Address for Accessing Graphics Data Start address in the frame buffer where graphics data is to be stored. Fix GRn_FLM2.BASE [5:0] to 0 during 64-byte burst transfer.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

BASE[31:0] bits (Base Address for Accessing Graphics Data)

The BASE[31:0] bits specify the base address (start address in the first frame buffer) for graphics data access. Because the GLCDC only supports the 16-beat increment burst transfer mode, in which data is aligned with a 64-byte boundary, the lower 6 bits (GRn_FLM2.BASE[5:0]) must be fixed to 0.

56.2.13 GRn_FLM3 : Graphics n Frame Buffer Control Register 3 (n = 1 to 2)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1110 + 0x100 × (n -1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	LNOFF[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
31:16	LNOFF[15:0]	Macro Line Offset Address for Accessing Graphics Data Macro line offset address for accessing graphics data (frame buffer data). Signed, 16-bit integer.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

LNOFF[15:0] bits (Macro Line Offset Address for Accessing Graphics Data)

The LNOFF[15:0] bits specify the macro line offset address for accessing graphics data (offset to be added to the current address at the macro line end for calculating the start address of the next macro line). Because the GLCDC only supports the 16-beat increment burst transfer mode, in which data is aligned with a 64-byte boundary, the lower 6 bits (GRn_FLM3.LNOFF[5:0]) must be fixed to 0.

56.2.14 GRn_FLM5 : Graphics n Frame Buffer Control Register 5 (n = 1 to 2)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1118 + 0x100 × (n -1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	LNNUM[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DATANUM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DATANUM[15:0]	Number of Data Transfer Times Per Line for Accessing Graphics Data Number of data transfers per macro line for accessing graphics data (frame buffer data), where one transfer is defined as 16-beat burst access (64-byte boundary). 0x0000 Once : ⋮ 0xFFFF 65536 times :	R/W
26:16	LNNUM[10:0]	Number of Lines Per Frame for Accessing Graphics Data Number of macro lines per frame for accessing graphics data (frame buffer data). 0x000: 1 macro line ⋮ 0x7FB: 2044 macro lines Others: Setting prohibited	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

DATANUM[15:0] bits (Number of Data Transfer Times Per Line for Accessing Graphics Data)

The DATANUM[15:0] bits specify the number of data transfers per macro line for accessing graphics data, where one transfer is defined as a 16-beat burst access through the GLCDC0 and GLCDC1 bus. For all the data transfers, at least 2 bytes of valid pixel data are required (16 pixels in LUT1 format and 16 pixels or more in the other formats). If the number of bytes per macro line cannot be divided by 64 (4 bytes × 16-beat), DATANUM is obtained by rounding up to the whole number.

LNNUM[10:0] bits (Number of Lines Per Frame for Accessing Graphics Data)

The LNNUM[10:0] bits specify the number of lines per frame for accessing graphics data. When graphics data for the number of lines set to these bits is read, it signals the end of the frame and the base address is loaded.

The following are two use cases for macro lines. In these use cases, the frame size is 480 pixels × 272 lines and the pixel format is RGB565 (16 bpp).

Case 1) One macro line is configured to be equivalent to the frame raster width. The number of macro lines is equivalent to the number of vertical lines.

DATANUM: 0x000E (15 times = 16 × 480 / 512)

LNNUM: 0x10F (272 macro lines = 272 / 1)

Case 2) One macro line is configured to be repeated during the display frame. The number of macro lines is not equivalent to the number of vertical lines. In the example, one macro line has 16 times the raster width.

DATANUM: 0x00EF (240 times = 16 × 480 / 512 × 16)

LNNUM: 0x010 (17 macro lines = 272 / 16)

56.2.15 GRn_FLM6 : Graphics n Frame Buffer Control Register 6 (n = 1, 2)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x111C + 0x100 × (n -1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	FORMAT[2:0]			—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
27:0	—	These bits are read as 0. The write value should be 0.	R/W
30:28	FORMAT[2:0]	Data Format for Accessing Graphics Data Data format for accessing graphics data (frame buffer data). 0 0 0: RGB565 (16 bits/pixel) 0 0 1: RGB888 (32 bits/pixel, 8 bits on the MSB side are invalid) 0 1 0: ARGB1555 (16 bits/pixel, 1 bit of A is LUT data) 0 1 1: ARGB4444 (16 bits/pixel) 1 0 0: ARGB8888 (32 bits/pixel) 1 0 1: CLUT8 (8 bits/pixel) 1 1 0: CLUT4 (4 bits/pixel) 1 1 1: CLUT1 (1 bit/pixel)	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

FORMAT[2:0] bits (Data Format for Accessing Graphics Data)

The FORMAT[2:0] bits specify the data format for accessing graphics data. CLUT1, CLUT4, CLUT8 contain the addresses 0x01 or 0x00, 0x0F to 0x00, and 0xFF to 0x00, respectively, for accessing the color palette. ARGB1555 contains the address for accessing the color palette (0x80 or 0x00) in the MSB and RGB data in the other bits. ARGB8888 and ARGB4444 contain the upper 8-bit or 4-bit alpha blending values and RGB data. RGB888 and RGB565 contain RGB data only.

56.2.16 GRn_AB1 : Graphics n Alpha Blending Control Register 1 (n = 1, 2)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1120 + 0x100 × (n -1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	ARCON	—	—	—	ARCDISPON	—	—	—	GRCDISPON	—	—	DISPSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	DISPSEL[1:0]	Graphics Display Plane Control 0 0: Background color display (value set in the GRn_BASE register) 0 1: Lower-layer graphics display 1 0: Current graphics display 1 1: Blended display of lower-layer graphics (input image from the previous stage) and current graphics (data read from the GLCDC0 and GLCDC1 bus)	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	GRCDISPON	Graphics Image Area Border Display Control 0: Turn display off 1: Turn display on	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	ARCDISPON	Image Area Border Display Control for Rectangular Area Alpha Blending 0: Turn display off 1: Turn display on	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
12	ARCON	Rectangular Area Alpha Blending Control 0: Turn blending off 1: Turn blending on	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

DISPSEL[1:0] bits (Graphics Display Plane Control)

The DISPSEL[1:0] bits control the graphics display plane. When the plane is selected as the lower-layer graphics, the image input from the previous stage is displayed (the background plane for GR1, and output from GR1 for GR2); as the background color, the background color specified in the GRn_BASE registers is displayed; and as the current graphics, the ARGB8888 data obtained by expanding the graphics data read by the GLCDC from the GLCDC0 and GLCDC1 bus is displayed. When the current graphics display is selected (these bits are set to 10b), RGB888 data is displayed, regardless of the alpha blending value in the pixel. Table 56.8 and Figure 56.10 show the relationship between the register setting and display area.

GRCDISPON bit (Graphics Image Area Border Display Control)

The GRCDISPON bit turns on or off the border display for the graphics image area. When the display is turned on (this bit is set to 1), the graphics image area is bordered with the preset color. The border is 1 pixel wide on the outermost periphery of the area with the display data set as 0xFF for each RGB color.

ARCDISPON bit (Image Area Border Display Control for Rectangular Area Alpha Blending)

The ARCDISPON bit turns on or off the border display for the image area where rectangular area alpha blending is performed. When the display is turned on (this bit is set to 1), the image area for the rectangular alpha blending is bordered with the preset color. The border is 1 pixel wide on the outermost periphery of the area and the display data set as 0xFF for each RGB color.

ARCON bit (Rectangular Area Alpha Blending Control)

The ARCON bit turns on or off alpha blending in a rectangular area. When alpha blending is turned on (this bit is set to 1), pixels are processed in accordance with the alpha blending control specified in the relevant registers for the specified rectangular area, not in accordance with the alpha value input from the graphics data interface for each pixel. In the areas outside the rectangular area in the graphics area, pixels are processed in accordance with the alpha value input from the graphics data interface for each pixel.

Table 56.8 Display selections

GRn_AB1. DISPSEL[1:0] (display plane)	GRn_AB1. ARCON (rectangular)	GRn_AB7. CKON (chroma key)	Within rectangular alpha blending area	Outside rectangular alpha blending area within graphics area	Outside graphics area within display area	Outside display area
00b	0	0	—	Background color	Background color	Lower layer R = G = B = 0x00
01b	0	0	—	Lower layer	Lower layer	Lower layer R = G = B = 0x00
10b	0	0	—	Current	Background color	Lower layer R = G = B = 0x00
11b	0	0	—	Current + alpha blending in pixel units	Lower layer	Lower layer R = G = B = 0x00
11b	0	1	—	Current + RGB-index chroma key + alpha blending in pixel units	Lower layer	Lower layer R = G = B = 0x00
11b	1	0	Current + rectangular alpha blending	Current + alpha blending in pixel units	Lower layer	Lower layer R = G = B = 0x00
11b	1	1	Current + rectangular alpha blending	Current + RGB-index chroma key + alpha blending in pixel units	Lower layer	Lower layer R = G = B = 0x00

Note: Operation is not guaranteed when any other value is set.

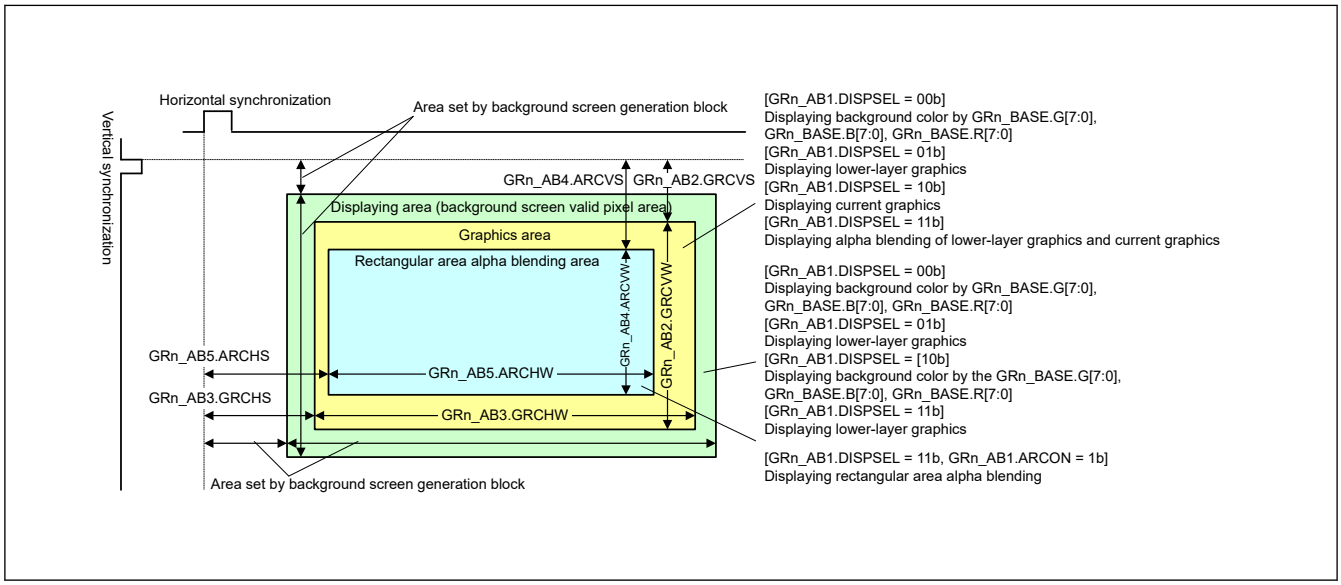


Figure 56.10 Selection of graphics display plane

56.2.17 GRn_AB2 : Graphics n Alpha Blending Control Register 2 (n = 1, 2)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1124 + 0x100 × (n -1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	GRCVS[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	GRCVW[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
10:0	GRCVW[10:0]	Vertical Width of Graphics Image Area Width in lines. 0x010: 16 lines ⋮ 0x7FC: 2044 lines Others: Setting prohibited	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
26:16	GRCVS[10:0]	Vertical Start Position of Graphics Image Area Position in lines. 0x002: 2nd line ⋮ 0x7EE: 2030th line Others: Setting prohibited	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GRCVW[10:0] bits (Vertical Width of Graphics Image Area)

The GRCVW[10:0] bits specify the vertical width of the graphics image area.

GRCVS[10:0] bits (Vertical Start Position of Graphics Image Area)

The GRCVS[10:0] bits specify the vertical start position of the graphics image area, in reference to assertion of the vertical synchronization signal (VS). For the relationship with the graphics display plane, see Figure 56.10.

56.2.18 GRn_AB3 : Graphics n Alpha Blending Control Register 3 (n = 1, 2)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1128 + 0x100 × (n -1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	GRCHS[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	GRCHW[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
10:0	GRCHW[10:0]	Horizontal Width of Graphics Image Area Width in pixels. 0x010: 16 pixels ⋮ 0x7F8: 2040 pixels Others: Setting prohibited	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/(W)
26:16	GRCHS[10:0]	Horizontal Start Position of Graphics Image Area Position in pixels. 0x005: 5th pixel ⋮ 0x7ED: 2029th pixel Others: Setting prohibited	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GRCHW[10:0] bits (Horizontal Width of Graphics Image Area)

The GRCHW[10:0] bits specify the horizontal width of the graphics image area.

GRCHS[10:0] bits (Horizontal Start Position of Graphics Image Area)

The GRCHS[10:0] bits specify the horizontal start position of the graphics image area, in reference to assertion of the horizontal synchronization signal (VS). For the relationship with the graphics display plane, see Figure 56.10.

56.2.19 GRn_AB4 : Graphics n Alpha Blending Control Register 4 (n = 1, 2)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x112C + 0x100 × (n -1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	ARCVS[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ARCVW[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
10:0	ARCVW[10:0]	Vertical Width of Rectangular Area Alpha Blending Image Area Width in lines. 0x001: 1 line ⋮ 0x7FC: 2044 lines Others: Setting prohibited	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
26:16	ARCVS[10:0]	Vertical Start Position of Rectangular Area Alpha Blending Image Area Position in lines. 0x002: 2nd line ⋮ 0x7EE: 2030th line Others: Setting prohibited	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

ARCVW[10:0] bits (Vertical Width of Rectangular Area Alpha Blending Image Area)

The ARCVW[10:0] bits specify the vertical width of the rectangular area alpha blending image area.

ARCVS[10:0] bits (Vertical Start Position of Rectangular Area Alpha Blending Image Area)

The ARCVS[10:0] bits specify the vertical start position of the rectangular area alpha blending image area, in reference to assertion of the vertical synchronization signal (VS). For the relationship with the graphics display plane, see Figure 56.10.

56.2.20 GRn_AB5 : Graphics n Alpha Blending Control Register 5 (n = 1, 2)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1130 + 0x100 × (n -1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	ARCHS[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ARCHW[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
10:0	ARCHW[10:0]	Horizontal Width of Rectangular Area Alpha Blending Image Area Width in pixels. 0x001: 1 pixel ⋮ 0x7F8: 2040 pixels Others: Setting prohibited	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
26:16	ARCHS[10:0]	Horizontal Start Position of Rectangular Area Alpha Blending Image Area Position in pixels. 0x005: 5th pixel ⋮ 0x7ED: 2029th pixel Others: Setting prohibited	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

ARCHW[10:0] bits (Horizontal Width of Rectangular Area Alpha Blending Image Area)

The ARCHW[10:0] bits specify the horizontal width of the rectangular area alpha blending image area.

ARCHS[10:0] bits (Horizontal Start Position of Rectangular Area Alpha Blending Image Area)

The ARCHS[10:0] bits specify the horizontal start position of the rectangular area alpha blending image area, in reference to assertion of the horizontal synchronization signal (HS). For the relationship with the graphics display plane, see Figure 56.10.

56.2.21 GRn_AB6 : Graphics n Alpha Blending Control Register 6 (n = 1, 2)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1134 + 0x100 × (n -1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	ARCCOEF[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	ARCRATE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	ARCRATE[7:0]	Frame Rate for Alpha Blending in Rectangular Area 0x00: 1 frame ⋮ 0xFF: 256 frames	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
24:16	ARCCOEF[8:0]	Alpha Coefficient for Alpha Blending in Rectangular Area Valid settings: -255 to 255. Bit [8]: Sign 0: Add 1: Subtract Bits [7:0]: Variation, as an absolute value	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

ARCRATE[7:0] bits (Frame Rate for Alpha Blending in Rectangular Area)

The ARCRATE[7:0] bits specify the frame rate for alpha blending in a rectangular area.

ARCCOEF[8:0] bits (Alpha Coefficient for Alpha Blending in Rectangular Area)

The ARCCOEF[8:0] bits specify the alpha coefficient for alpha blending in a rectangular area.

In alpha blending in a rectangular area, current graphics are faded in or out with the GRn_AB7.ARCDEF[7:0], GRn_AB6.ARCCOEF[8:0], and GRn_AB6.ARCRATE[7:0] settings. If the alpha value is set in the GRn_AB7.ARCDEF[7:0] bits, the GRn_AB7.ARCDEF[7:0] bits and the alpha blending in a rectangular area are turned on. Each time the vertical synchronization signal (VS) rises the number of times set in the GRn_AB6.ARCRATE[7:0] bits, the value in GRn_AB6.ARCCOEF[8:0] is added to or subtracted from the alpha value. Figure 56.11 shows change in the alpha value.

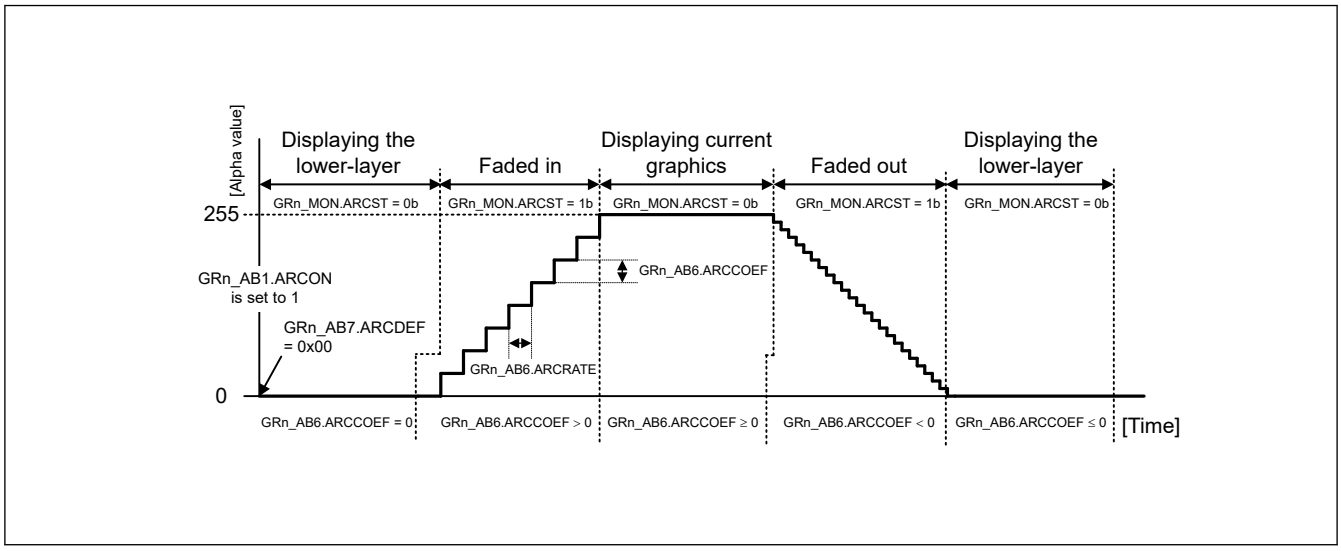


Figure 56.11 Changes in alpha value

56.2.22 GRn_AB7 : Graphics n Alpha Blending Control Register 7 (n = 1, 2)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1138 + 0x100 × (n -1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	ARCDEF[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CKON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CKON	RGB-Index Chroma-Key Processing Control 0: Disable chroma-key processing 1: Enable chroma-key processing	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
23:16	ARCDEF[7:0]	Initial Alpha Value for Alpha Blending in Rectangular Area	R/W

Bit	Symbol	Function	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

CKON bit (RGB-Index Chroma-Key Processing Control)

The CKON bit turns RGB-index chroma-key processing on or off. This function is enabled only if this bit is 1 when the graphics display plane is the blended display (GRn_AB1.DISPSEL[1:0] = 11b). And it is reflected to the graphics area except alpha blending in a rectangular area. For details, see Table 56.8.

ARCDEF[7:0] bits (Initial Alpha Value for Alpha Blending in Rectangular Area)

The ARCDEF[7:0] bits specify the initial alpha value for alpha blending in a rectangular area. For changes in the alpha value during fade-in or fade-out of the current graphics using this bit, see Figure 56.11.

56.2.23 GRn_AB8 : Graphics n Alpha Blending Control Register 8 (n = 1, 2)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x113C + 0x100 × (n -1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	CKKG[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CKKB[7:0]								CKKR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CKKR[7:0]	R Signal for RGB-Index Chroma-Key Processing R signal for RGB-index chroma-key processing. Unsigned 8-bit value.	R/W
15:8	CKKB[7:0]	B Signal for RGB-Index Chroma-Key Processing B signal for RGB-index chroma-key processing. Unsigned 8-bit value.	R/W
23:16	CKKG[7:0]	G Signal for RGB-Index Chroma-Key Processing G signal for RGB-index chroma-key processing. Unsigned 8-bit value.	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

If RGB-index chroma-key processing is enabled when the RGB values of the current bit graphics agree with the values of the associated chroma-key bits, the image data of the current graphics (ARGB8888, including the alpha blending values) is replaced by the values in the GRn_AB9 register. In alpha blending in pixel units at later stages, the latest alpha values are used.

CKKR[7:0] bits (R Signal for RGB-Index Chroma-Key Processing)

The CKKR[7:0] bits specify the value to be compared with the R value of the current graphics in the RGB-index chromakey processing.

CKKB[7:0] bits (B Signal for RGB-Index Chroma-Key Processing)

The CKKB[7:0] bits specify the value to be compared with the B value of the current graphics in the RGB-index chromakey processing.

CKKG[7:0] bits (G Signal for RGB-Index Chroma-Key Processing)

The CKKG[7:0] bits specify the value to be compared with the G value of the current graphics in the RGB-index chroma-key processing.

56.2.24 GRn_AB9 : Graphics n Alpha Blending Control Register 9 (n = 1, 2)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1140 + 0x100 × (n -1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CKA[7:0]								CKG[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CKB[7:0]								CKR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CKR[7:0]	R Value after RGB-Index Chroma-Key Processing Replacement R value after RGB-index chroma-key processing replacement. Unsigned 8-bit value.	R/W
15:8	CKB[7:0]	B Value after RGB-Index Chroma-Key Processing Replacement B value after RGB-index chroma-key processing replacement. Unsigned 8-bit value.	R/W
23:16	CKG[7:0]	G Value after RGB-Index Chroma-Key Processing Replacement G value after RGB-index chroma-key processing replacement. Unsigned 8-bit value.	R/W
31:24	CKA[7:0]	A Value after RGB-Index Chroma-Key Processing Replacement A value after RGB-index chroma-key processing replacement.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

If the RGB-index chroma-key processing is enabled and the pixel data is to be replaced (the RGB values of the current graphics agree with the GRn_AB8 values), alpha blending in pixel units is performed in later stages in accordance with this image data.

CKR[7:0] bits (R Value after RGB-Index Chroma-Key Processing Replacement)

The CKR[7:0] bits specify the R value after RGB-index chroma-key processing replacement.

CKB[7:0] bits (B Value after RGB-Index Chroma-Key Processing Replacement)

The CKB[7:0] bits specify the B value after RGB-index chroma-key processing replacement.

CKG[7:0] bits (G Value after RGB-Index Chroma-Key Processing Replacement)

The CKG[7:0] bits specify the G value after RGB-index chroma-key processing replacement.

CKA[7:0] bits (A Value after RGB-Index Chroma-Key Processing Replacement)

The CKA[7:0] bits specify the A value after RGB-index chroma-key processing replacement.

56.2.25 GRn_BASE : Graphics n Background Color Control Register (n = 1, 2)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x114C + 0x100 × (n -1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	G[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	B[7:0]								R[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	R[7:0]	Background Color R Value Background color R value. Unsigned 8-bit value.	R/W
15:8	B[7:0]	Background Color B Value Background color B value. Unsigned 8-bit value.	R/W
23:16	G[7:0]	Background Color G Value Background color G value. Unsigned 8-bit value.	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

When the background color is selected in the display selection (GRn_AB1.DISPSEL[1:0] = 00b), this RGB data is output to the entire display image area. When the current graphics setting is selected (GRn_AB1.DISPSEL[1:0] = 10b), the RGB data is output to the outside of the graphics image area within the display image area.

R[7:0] bits (Background Color R Value)

The R[7:0] bits specify the background color R value.

B[7:0] bits (Background Color B Value)

The B[7:0] bits specify the background color B value.

G[7:0] bits (Background Color G Value)

The G[7:0] bits specify the background color G value.

56.2.26 GRn_CLUTINT : Graphics n CLUT Table Interrupt Control Register (n = 1, 2)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1150 + 0x100 × (n -1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	LINE[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	LINE[10:0]	Number of Detection Lines 0x000: 1 line ⋮ 0x7FF: 2048 lines	R/W
15:11	—	These bits are read as 0. The write value should be 0./	R/W
16	SEL	CLUT Table Control 0: Select CLUT table 0 1: Select CLUT table 1	R/W
31:17	—	These bits are read as 0. The write value should be 0./	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

LINE[10:0] bits (Number of Detection Lines)

The LINE[10:0] bits specify the number of lines to be detected. When the number of lines specified in this bit are detected, the event is recognized outside the module on the HS assertion. To retain the status of the recognized event and assert the GLCDC interrupt request signal, set the prescribed value to the State Detection Control Register (SYSCNT_DTCTEN) and Interrupt Request Enable Control Register (SYSCNT_INTEN), which are system control registers. Although this function is provided to both graphics 1 and 2, it is only enabled in graphics 2 in this GLCDC.

SEL bit (CLUT Table Control)

The SEL bit controls the CLUT plane to be used for internal operations. Access to the color palette (CLUT) through the register access bus is always valid for both planes 0 and 1, regardless of the setting in this bit, and the written value is immediately reflected to the internal operations (not in synchronization with the vertical synchronization signal).

To keep reflection of the CLUT memory contents to the internal operations in synchronization with the vertical synchronization signal, first write data through the register access bus to the plane that is not being used for the internal operations, and then modify the bits intended for controlling the plane that is to be used for the internal operations.

56.2.27 GRn_MON : Graphics n Status Monitor Register (n = 1, 2)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1154 + 0x100 × (n -1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UNDFLST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARCST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ARCST	Status Monitor for Alpha Blending in Rectangular Area 0: Fade-in/fade-out not in progress 1: Fade-in/fade-out in progress	R
15:1	—	These bits are read as 0.	R
16	UNDFLST	Underflow Status Monitor 0: No underflow occurred in internal operations 1: Underflow occurred in internal operations	R
31:17	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

ARCST bit (Status Monitor for Alpha Blending in Rectangular Area)

The ARCST bit indicates whether or not alpha blending (fade-in/fade-out) in a rectangular area is in progress. When alpha blending in a rectangular area is turned on (GRn_AB1.ARCON is set to 1) and the register value is to be reflected to the internal operations on assertion of the vertical synchronization signal (VS), this bit sets to 1 immediately on assertion of the vertical synchronization signal. When alpha blending in a rectangular area is turned off (GRn_AB1.ARCON cleared to 0) or when the alpha blending (fade-in/fade-out) in a rectangular area is complete (alpha blended value reaches the minimum or maximum value), this bit clears to 0. If the alpha coefficient for the alpha blending in a rectangular area (GRn_AB6.ARCCOEF[8:0]) is set to 0x000 and the initial alpha value for alpha blending in a rectangular area (GRn_AB7.ARCDEF[7:0]) is set to any value other than 0xFF or 0x00, the alpha blending value does not reach the minimum or maximum value, and this bit remains 1 (no timeout processing is performed).

UNDFLST bit (Underflow Status Monitor)

The UNDFLST bit indicates whether or not an underflow has occurred in the internal operations. The underflow interrupt request flag sets when an underflow occurs, and retains its value until it is cleared by software. However, this bit monitors the internal status, and so the flag automatically clears to 0 when the graphics data bus interface initializes on assertion of the vertical synchronization signal. Even when the current graphics data is not required (GRn_AB1.DISPSEL[1:0] = 0xb), this bit sets to 1 during the period from the graphics image valid area start to the next frame vertical synchronization signal (VS) assertion timing set in the registers.

56.2.28 GAMn_LATCH : Gamma n Register Update Control Register (n = G, B, R)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1300 (GAMG_LATCH)
0x1340 (GAMB_LATCH)
0x1380 (GAMR_LATCH)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VEN	Control of Gamma Correction × Module Register Value Reflection to Internal Operations 0: Disable reflection of register values to internal operations on assertion of vertical synchronization signal (VS) 1: Enable reflection of register values to internal operations on assertion of the vertical synchronization signal (VS)	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

VEN bit (Control of Gamma Correction × Module Register Value Reflection to Internal Operations)

The VEN bit enables or disables reflection of the register values to the internal operations in the gamma correction circuit on assertion of the vertical synchronization signal (input). When this bit is set to 1, the register values are immediately reflected to the internal operations on assertion of the vertical synchronization signal (input), and then this bit automatically clears to 0. Also, if the signal is asserted that controls reflection of the register values to the internal operations of all the modules output from the background plane generation module, the register values are reflected to the internal operations on assertion of the vertical synchronization signal (input), regardless of the value of this bit. While this bit is 1, do not modify any register whose value is reflected to the internal operations on assertion of the vertical synchronization signal (input) in the GLCDC. Otherwise, operation is not guaranteed.

This bit must not be 1 at the same time as the BG_EN.VEN bit (control of background plane register value reflection to internal operations) in the Operation Control Register (BG_EN), one of the background plane setting registers. Otherwise, operation is not guaranteed.

Although there are three VEN bits, one for each G, R, and B color, only the GAMG_LATCH.VEN bit controls the gamma correction with the reflection of the GAM_SW.GAMON bit. To enable gamma correction, set the GAMG_LATCH.VEN bit once after setting the GAM_SW.GAMON bit.

The VEN bit is set to 1 by writing 1, and automatically clears to 0 immediately on assertion of the vertical synchronization signal.

56.2.29 GAM_SW : Gamma Correction Block Function Switch Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1304

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GAMON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	GAMON	Gamma Correction On/Off Control 0: Turn off gamma correction 1: Turn on gamma correction	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMG_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GAMON bit (Gamma Correction On/Off Control)

The GAMON bit turns on or off gamma correction.

56.2.30 GAMn_LUT1 : Gamma n Correction Block Table Setting Register 1 (n = G, B, R)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1308 (GAMG_LUT1)
0x1348 (GAMB_LUT1)
0x1388 (GAMR_LUT1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	GAIN00[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	GAIN01[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	GAIN01[10:0]	Gain Value of Area 1 Unsigned 11-bit fixed point value. 0x000: 0.000 (0/1024) ⋮ 0x400: 1.000 (1024/1024) ⋮ 0x7FF: 1.999 (2047/1024)	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
26:16	GAIN00[10:0]	Gain Value of Area 0 0x000: 0.000 (0/1024) ⋮ 0x400: 1.000 (1024/1024) ⋮ 0x7FF: 1.999 (2047/1024)	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

Gamma correction for each color is carried out as follows:

- Din (input signal): 10 bits. To correct the output of graphics 2, 00b is appended as the lower 2 bits for extension
- Dout (output signal): 10 bits
- TH (threshold): 10 bits (register setting); up to 15 can be set
- GAIN (gain): 0/1024 to 2047/1024 for each area (register setting), can be set for up to 16 areas
- OFFSET (offset value): 21 bits (result of internal calculation; calculation of up to 15 points)

The following is automatically calculated from the assertion of the vertical synchronization signal (VS) to the start of valid pixel data internally:

- $D_{out} = ((D_{in} - TH(n)) \times GAIN(n) + OFFSET(n))$
- $OFFSET(n) = OFFSET(n-1) + (TH(n) - TH(n-1)) \times GAIN(n-1),\ \text{where}\ OFFSET(n) = 0$

Note: Because the gain is a positive number (≥ 0), the correction line shows a monotonic increase.

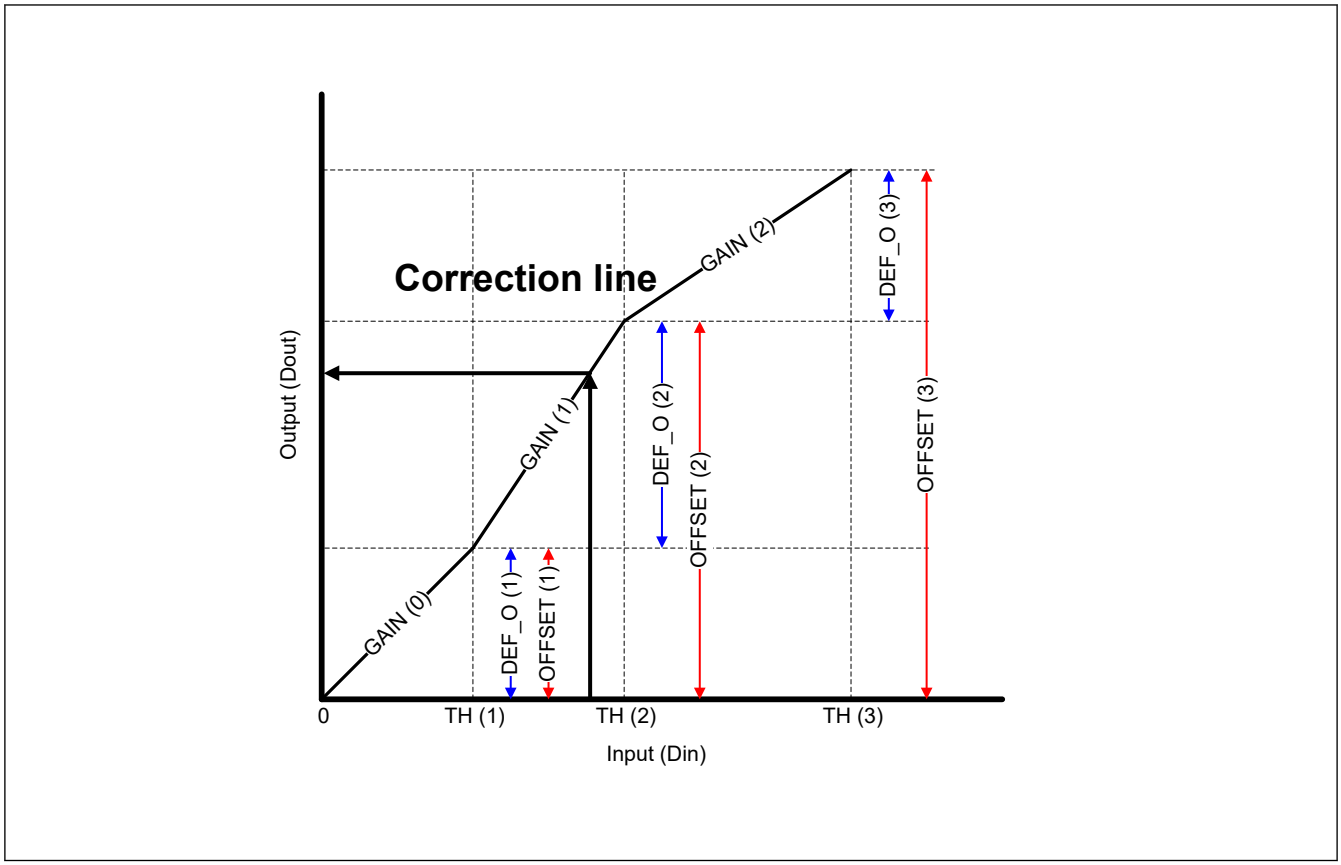


Figure 56.12 Calculation of gamma correction value

GAIN01[10:0] bits (Gain Value of Area 1)

The GAIN01[10:0] bits specify the gain value of area 1 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [9] and [10].

GAIN00[10:0] bits (Gain Value of Area 0)

The GAIN00[10:0] bits specify the gain value of area 0 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [9] and [10].

56.2.31 GAMn_LUT2 : Gamma n Correction Block Table Setting Register 2 (n = G, B, R)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x130C (GAMG_LUT2)
0x134C (GAMB_LUT2)
0x138C (GAMR_LUT2)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	GAIN02[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	GAIN03[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	GAIN03[10:0]	Gain Value of Area 3 Unsigned 11-bit fixed point value. 0x000: 0.000 (0/1024) ⋮ 0x400: 1.000 (1024/1024) ⋮ 0x7FF: 1.999 (2047/1024)	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
26:16	GAIN02[10:0]	Gain Value of Area 2 Unsigned 11-bit fixed point value. 0x000: 0.000 (0/1024) ⋮ 0x400: 1.000 (1024/1024) ⋮ 0x7FF: 1.999 (2047/1024)	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GAIN03[10:0] bits (Gain Value of Area 3)

The GAIN03[10:0] bits specify the gain value of area 3 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [9] and [10].

GAIN02[10:0] bits (Gain Value of Area 2)

The GAIN02[10:0] bits specify the gain value of area 2 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [9] and [10].

56.2.32 GAMn_LUT3 : Gamma n Correction Block Table Setting Register 3 (n = G, B, R)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1310 (GAMG_LUT3)
0x1350 (GAMB_LUT3)
0x1390 (GAMR_LUT3)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	GAIN04[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	GAIN05[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	GAIN05[10:0]	Gain Value of Area 5 Unsigned 11-bit fixed point value. 0x000: 0.000 (0/1024) ⋮ 0x400: 1.000 (1024/1024) ⋮ 0x7FF: 1.999 (2047/1024)	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
26:16	GAIN04[10:0]	Gain Value of Area 4 Unsigned 11-bit fixed point value. 0x000: 0.000 (0/1024) ⋮ 0x400: 1.000 (1024/1024) ⋮ 0x7FF: 1.999 (2047/1024)	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GAIN05[10:0] bits (Gain Value of Area 5)

The GAIN05[10:0] bits specify the gain value of area 5 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [9] and [10].

GAIN04[10:0] bits (Gain Value of Area 4)

The GAIN04[10:0] bits specify the gain value of area 4 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [9] and [10].

56.2.33 GAMn_LUT4 : Gamma n Correction Block Table Setting Register 4 (n = G, B, R)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1314 (GAMG_LUT4)
0x1354 (GAMB_LUT4)
0x1394 (GAMR_LUT4)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	GAIN06[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	GAIN07[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	GAIN07[10:0]	Gain Value of Area 7 Unsigned 11-bit fixed point value. 0x000: 0.000 (0/1024) ⋮ 0x400: 1.000 (1024/1024) ⋮ 0x7FF: 1.999 (2047/1024)	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
26:16	GAIN06[10:0]	Gain Value of Area 6 Unsigned 11-bit fixed point value. 0x000: 0.000 (0/1024) ⋮ 0x400: 1.000 (1024/1024) ⋮ 0x7FF: 1.999 (2047/1024)	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GAIN07[10:0] bits (Gain Value of Area 7)

The GAIN07[10:0] bits specify the gain value of area 7 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [9] and [10].

GAIN06[10:0] bits (Gain Value of Area 6)

The GAIN06[10:0] bits specify the gain value of area 6 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [9] and [10].

56.2.34 GAMn_LUT5 : Gamma n Correction Block Table Setting Register 5 (n = G, B, R)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1318 (GAMG_LUT5)
0x1358 (GAMB_LUT5)
0x1398 (GAMR_LUT5)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	GAIN08[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	GAIN09[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	GAIN09[10:0]	Gain Value of Area 9 Unsigned 11-bit fixed point value. 0x000: 0.000 (0/1024) ⋮ 0x400: 1.000 (1024/1024) ⋮ 0x7FF: 1.999 (2047/1024)	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
26:16	GAIN08[10:0]	Gain Value of Area 8 Unsigned 11-bit fixed point value. 0x000: 0.000 (0/1024) ⋮ 0x400: 1.000 (1024/1024) ⋮ 0x7FF: 1.999 (2047/1024)	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GAIN09[10:0] bits (Gain Value of Area 9)

The GAIN09[10:0] bits specify the gain value of area 9 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [9] and [10].

GAIN08[10:0] bits (Gain Value of Area 8)

The GAIN08[10:0] bits specify the gain value of area 8 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [9] and [10].

56.2.35 GAMn_LUT6 : Gamma n Correction Block Table Setting Register 6 (n = G, B, R)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x131C (GAMG_LUT6)
0x135C (GAMB_LUT6)
0x139C (GAMR_LUT6)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	GAIN10[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	GAIN11[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	GAIN11[10:0]	Gain Value of Area 11 Unsigned 11-bit fixed point value. 0x000: 0.000 (0/1024) ⋮ 0x400: 1.000 (1024/1024) ⋮ 0x7FF: 1.999 (2047/1024)	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
26:16	GAIN10[10:0]	Gain Value of Area 10 Unsigned 11-bit fixed point value. 0x000: 0.000 (0/1024) ⋮ 0x400: 1.000 (1024/1024) ⋮ 0x7FF: 1.999 (2047/1024)	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GAIN11[10:0] bits (Gain Value of Area 11)

The GAIN11[10:0] bits specify the gain value of area 11 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [9] and [10].

GAIN10[10:0] bits (Gain Value of Area 10)

The GAIN10[10:0] bits specify the gain value of area 10 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [9] and [10].

56.2.36 GAMn_LUT7 : Gamma n Correction Block Table Setting Register 7 (n = G, B, R)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1320 (GAMG_LUT7)
0x1360 (GAMB_LUT7)
0x13A0 (GAMR_LUT7)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	GAIN12[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	GAIN13[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	GAIN13[10:0]	Gain Value of Area 13 Unsigned 11-bit fixed point value. 0x000: 0.000 (0/1024) ⋮ 0x400: 1.000 (1024/1024) ⋮ 0x7FF: 1.999 (2047/1024)	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
26:16	GAIN12[10:0]	Gain Value of Area 12 Unsigned 11-bit fixed point value. 0x000: 0.000 (0/1024) ⋮ 0x400: 1.000 (1024/1024) ⋮ 0x7FF: 1.999 (2047/1024)	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GAIN13[10:0] bits (Gain Value of Area 13)

The GAIN13[10:0] bits specify the gain value of area 13 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [9] and [10].

GAIN12[10:0] bits (Gain Value of Area 12)

The GAIN12[10:0] bits specify the gain value of area 12 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [9] and [10].

56.2.37 GAMn_LUT8 : Gamma n Correction Block Table Setting Register 8 (n = G, B, R)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1324 (GAMG_LUT8)
0x1364 (GAMB_LUT8)
0x13A4 (GAMR_LUT8)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	GAIN14[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	GAIN15[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	GAIN15[10:0]	Gain Value of Area 15 Unsigned 11-bit fixed point value. 0x000: 0.000 (0/1024) ⋮ 0x400: 1.000 (1024/1024) ⋮ 0x7FF: 1.999 (2047/1024)	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
26:16	GAIN14[10:0]	Gain Value of Area 14 Unsigned 11-bit fixed point value. 0x000: 0.000 (0/1024) ⋮ 0x400: 1.000 (1024/1024) ⋮ 0x7FF: 1.999 (2047/1024)	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GAIN15[10:0] bits (Gain Value of Area 15)

The GAIN15[10:0] bits specify the gain value of area 15 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [9] and [10].

GAIN14[10:0] bits (Gain Value of Area 14)

The GAIN14[10:0] bits specify the gain value of area 14 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [9] and [10].

56.2.38 GAMn_AREA1 : Gamma n Correction Block Area Setting Register 1 (n = G, B, R)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1328 (GAMG_AREA1)
0x1368 (GAMB_AREA1)
0x13A8 (GAMR_AREA1)

Bit position:	31	29	20	19	10	9	0
Bit field:	—	—	TH01[9:0]	TH02[9:0]	TH03[9:0]		
Value after reset:	0 0	0 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
9:0	TH03[9:0]	Start Threshold of Area 3 Unsigned 10-bit integer.	R/W
19:10	TH02[9:0]	Start Threshold of Area 2 Unsigned 10-bit integer.	R/W
29:20	TH01[9:0]	Start Threshold of Area 1 Unsigned 10-bit integer.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

Set the start threshold TH (n) of area n (n = 0 to 15) to satisfy the following conditions. Otherwise, operation is not guaranteed.

TH (n) < TH (n + 1)

n = 0 to 15 and TH (0) = 0x000, TH (16) = 0x3FF

TH (n) = TH (n + 1) is valid only if TH (n) = 0x3FF.

For details on calculation of the gamma correction value, see section 56.2.30. GAMn_LUT1 : Gamma n Correction Block Table Setting Register 1 (n = G, B, R) and Figure 56.12

TH03[9:0] bits (Start Threshold of Area 3)

The TH03[9:0] bits specify the start threshold of area 3 to be used for gamma correction in terms of an unsigned 10-bit integer.

TH02[9:0] bits (Start Threshold of Area 2)

The TH02[9:0] bits specify the start threshold of area 2 to be used for gamma correction in terms of an unsigned 10-bit integer.

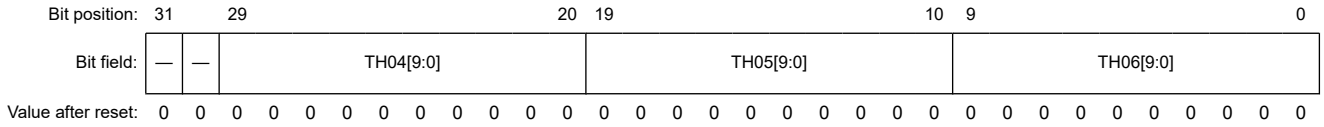
TH01[9:0] bits (Start Threshold of Area 1)

The TH01[9:0] bits specify the start threshold of area 1 to be used for gamma correction in terms of an unsigned 10-bit integer.

56.2.39 GAMn_AREA2 : Gamma n Correction Block Area Setting Register 2 (n = G, B, R)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x132C (GAMG_AREA2)
0x136C (GAMB_AREA2)
0x13AC (GAMR_AREA2)



Bit	Symbol	Function	R/W
9:0	TH06[9:0]	Start Threshold of Area 6 Unsigned 10-bit integer.	R/W
19:10	TH05[9:0]	Start Threshold of Area 5 Unsigned 10-bit integer.	R/W
29:20	TH04[9:0]	Start Threshold of Area 4 Unsigned 10-bit integer.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

TH06[9:0] bits (Start Threshold of Area 6)

The TH06[9:0] bits specify the start threshold of area 6 to be used for gamma correction in terms of an unsigned 10-bit integer.

TH05[9:0] bits (Start Threshold of Area 5)

The TH05[9:0] bits specify the start threshold of area 5 to be used for gamma correction in terms of an unsigned 10-bit integer.

TH04[9:0] bits (Start Threshold of Area 4)

The TH04[9:0] bits specify the start threshold of area 4 to be used for gamma correction in terms of an unsigned 10-bit integer.

56.2.40 GAMn_AREA3 : Gamma n Correction Block Area Setting Register 3 (n = G, B, R)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1330 (GAMG_AREA3)
0x1370 (GAMB_AREA3)
0x13B0 (GAMR_AREA3)

Bit position:	31	30	29 ... 20	19 ... 10	9 ... 0
Bit field:	—	—	TH07[9:0]	TH08[9:0]	TH09[9:0]
Value after reset:	0	0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
9:0	TH09[9:0]	Start Threshold of Area 9 Unsigned 10-bit integer.	R/W
19:10	TH08[9:0]	Start Threshold of Area 8 Unsigned 10-bit integer.	R/W
29:20	TH07[9:0]	Start Threshold of Area 7 Unsigned 10-bit integer.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

TH09[9:0] bits (Start Threshold of Area 9)

The TH09[9:0] bits specify the start threshold of area 9 to be used for gamma correction in terms of an unsigned 10-bit integer.

TH08[9:0] bits (Start Threshold of Area 8)

The TH08[9:0] bits specify the start threshold of area 8 to be used for gamma correction in terms of an unsigned 10-bit integer.

TH07[9:0] bits (Start Threshold of Area 7)

The TH07[9:0] bits specify the start threshold of area 7 to be used for gamma correction in terms of an unsigned 10-bit integer.

56.2.41 GAMn_AREA4 : Gamma n Correction Block Area Setting Register 4 (n = G, B, R)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1334 (GAMG_AREA4)
0x1374 (GAMB_AREA4)
0x13B4 (GAMR_AREA4)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TH10[9:0]										TH11[9:0]										TH12[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	TH12[9:0]	Start Threshold of Area 12 Unsigned 10-bit integer.	R/W
19:10	TH11[9:0]	Start Threshold of Area 11 Unsigned 10-bit integer.	R/W
29:20	TH10[9:0]	Start Threshold of Area 10 Unsigned 10-bit integer.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

TH12[9:0] bits (Start Threshold of Area 12)

The TH12[9:0] bits specify the start threshold of area 12 to be used for gamma correction in terms of an unsigned 10-bit integer.

TH11[9:0] bits (Start Threshold of Area 11)

The TH11[9:0] bits specify the start threshold of area 11 to be used for gamma correction in terms of an unsigned 10-bit integer.

TH10[9:0] bits (Start Threshold of Area 10)

The TH10[9:0] bits specify the start threshold of area 10 to be used for gamma correction in terms of an unsigned 10-bit integer.

56.2.42 GAMn_AREA5 : Gamma n Correction Block Area Setting Register 5 (n = G, B, R)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1338 (GAMG_AREA5)
0x1378 (GAMB_AREA5)
0x13B8 (GAMR_AREA5)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TH13[9:0]										TH14[9:0]										TH15[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	TH15[9:0]	Start Threshold of Area 15 Unsigned 10-bit integer.	R/W
19:10	TH14[9:0]	Start Threshold of Area 14 Unsigned 10-bit integer.	R/W
29:20	TH13[9:0]	Start Threshold of Area 13 Unsigned 10-bit integer.	R/W

Bit	Symbol	Function	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

TH15[9:0] bits (Start Threshold of Area 15)

The TH15[9:0] bits specify the start threshold of area 15 to be used for gamma correction in terms of an unsigned 10-bit integer.

TH14[9:0] bits (Start Threshold of Area 14)

The TH14[9:0] bits specify the start threshold of area 14 to be used for gamma correction in terms of an unsigned 10-bit integer.

TH13[9:0] bits (Start Threshold of Area 13)

The TH13[9:0] bits specify the start threshold of area 13 to be used for gamma correction in terms of an unsigned 10-bit integer.

56.2.43 OUT_VLATCH : Output Control Block Register Update Control Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x13C0

Bit position:	31																													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VEN	Control of Output Control Module Register Value Reflection to Internal Operations 0: Disable reflection of register values to internal operations on assertion of vertical synchronization signal (VS) 1: Enable reflection of register values to internal operations on assertion of vertical synchronization signal (VS).	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

VEN bit (Control of Output Control Module Register Value Reflection to Internal Operations)

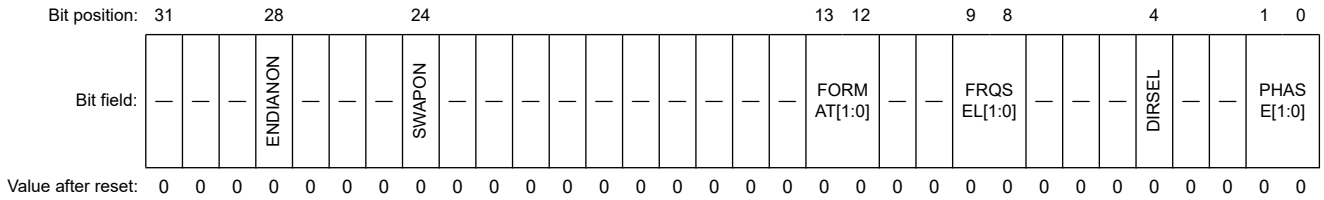
The VEN bit enables or disables reflection of the register values to the internal operations in the output control circuit on assertion of the vertical synchronization signal (input). When this bit is set to 1, the register values are immediately reflected to the internal operations on assertion of the vertical synchronization signal (input), and then this bit automatically clears to 0. Also, if the signal is asserted that controls reflection of the register values to the internal operations of all the modules output from the ground plane generation module, the register values are reflected to the internal operations on assertion of the vertical synchronization signal (input), regardless of the value of this bit. While this bit is 1, do not modify any register whose value is reflected to the internal operations on assertion of the vertical synchronization signal (input) in the GLCDC. Otherwise, operation is not guaranteed.

This bit must not be 1 at the same time as the BG_EN.VEN bit (control of background plane register value reflection to internal operations) in the Operation Control Register (BG_EN), or one of the background plane setting registers. Otherwise, operation is not guaranteed.

56.2.44 OUT_SET : Output Control Block Output Interface Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x13C4



Bit	Symbol	Function	R/W
1:0	PHASE[1:0]	Data Output Delay Control in Serial RGB Format Data delay in LCD_CLK cycles. 0 0: 0 cycle 0 1: 1 cycle 1 0: 2 cycles 1 1: 3 cycles	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	DIRSEL	Scan Direction Select of Serial RGB Format 0: Forward scan 1: Reverse scan	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
9:8	FRQSEL[1:0]	Clock Frequency Division Control 0 0: No frequency division, parallel RGB 0 1: Setting prohibited 1 0: Quarter frequency (serial RGB) 1 1: Setting prohibited	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	FORMAT[1:0]	Output Format Select 0 0: RGB888 — select RGB888 as dither output format 0 1: RGB666 — select RGB666 as dither output format 1 0: RGB565 — select RGB565 as dither output format 1 1: Serial RGB — select RGB888 as dither output format. Select dither output format in OUT_PDTHA.FORM[1:0]	R/W
23:14	—	These bits are read as 0. The write value should be 0.	R/W
24	SWAPON	Pixel Order Control 0: RGB order 1: BGR order	R/W
27:25	—	These bits are read as 0. The write value should be 0.	R/W
28	ENDIANON	Bit Endian Control 0: Descending order (little endian) 1: Ascending order (big endian)	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Rewriting these bits is prohibited during operation. Set the settings before operation is enabled. Operation is not guaranteed if the bit is rewritten during operation.

For details on assignment of pixel data to the output pins (LCD_DATA23 to LCD_DATA00) by setting these bits (except for OUT_SET.FRQSEL[1:0]), see Figure 56.3 to Figure 56.6.

PHASE[1:0] bits (Data Output Delay Control in Serial RGB Format)

The PHASE[1:0] bits control data output delay in serial RGB format. When the delay is 0 cycle (these bits are 00b), pixel data (R, B, or invalid data, depending on the setting in this register) is output one pixel clock (PXCLK) cycle after the

horizontal data enable signal (HE). When any value other than 00b is set to OUT_SET.PHASE[1:0], pixel data output is delayed for a preset number of LCD_CLK cycles.

DIRSEL bit (Scan Direction Select of Serial RGB Format)

The DIRSEL bit controls the data arrangement of the serial RGB format. When this bit is set to 1, the serial RGB data is arranged in reverse direction, and when it is 0, the serial RGB is arranged in forward direction.

FRQSEL[1:0] bits (Clock Frequency Division Control)

The FRQSEL[1:0] bits control clock frequency division of LCD_CLK (panel output clock) and PXCLK (pixel clock for internal operations). Set these bits to 10b only for the serial RGB format (OUT_SET.FORMAT[1:0] = 11b), so that PXCLK has a 1/4 frequency of the LCD_CLK frequency and synchronizes with LCD_CLK. Set these bits to 00b for the parallel RGB format (OUT_SET.FORMAT[1:0] = 10b, 01b, or 00b), so that PXCLK has the same frequency as the LCD_CLK frequency and synchronizes with LCD_CLK. Otherwise, operation is not guaranteed.

FORMAT[1:0] bits (Output Format Select)

The FORMAT[1:0] bits select the output format of RGB data. Set these bits in accordance with the output format select bits in the Panel Dither Correction Register (OUT_PDTHA.FORM[1:0]). For serial RGB format (these bits are 11b), set OUT_PDTHA.FORM[1:0] to 00b. Otherwise, operation is not guaranteed.

SWAPON bit (Pixel Order Control)

The SWAPON bit controls the pixel order of RGB data output. When this bit is set to 1, internally processed data is assigned to the output pins in BGR order, and when this bit is 0, data is assigned in the RGB order. Data is assigned to the output pins (LCD_DATA23 to LCD_DATA00) with the MSB first for the RGB parallel format, and serially for the RGB serial format.

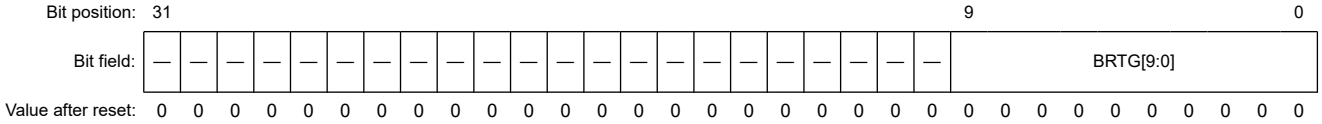
ENDIANON bit (Bit Endian Control)

The ENDIANON bit controls the bit order of RGB data output. When this bit is set to 1, internally processed data is assigned to the output pins in ascending order (big endian), and when this bit is 0, data is assigned in descending order (little endian).

56.2.45 OUT_BRIGHT1 : Output Control Block Brightness Correction Register 1

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x13C8



Bit	Symbol	Function	R/W
9:0	BRTG[9:0]	Brightness Adjustment of G Signal Brightness (DC) adjustment of G signal. Unsigned 10-bit integer, +512 with offset.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when OUT_VLATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

BRTG[9:0] bits (Brightness Adjustment of G Signal)

The BRTG[9:0] bits specify the brightness (DC) adjustment of the G signal.

Brightness correction of the G signal is performed as follows:

- Gout: Output of brightness correction (input of contrast correction). Unsigned 10 bits
- Gin: Input of brightness correction. Unsigned 10 bits
- BRTG: Setting in this bit
- Gout = Gin + BRTG - 512.

56.2.46 OUT_BRIGHT2 : Output Control Block Brightness Correction Register 2

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x13CC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	BRTB[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	BRTR[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	BRTR[9:0]	Brightness Adjustment of R Signal Brightness (DC) adjustment of R signal. Unsigned 10-bit integer, +512 with offset.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	BRTB[9:0]	Brightness Adjustment of B Signal Brightness (DC) adjustment of B signal. Unsigned 10-bit integer, +512 with offset.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when OUT_VLATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

BRTR[9:0] bits (Brightness Adjustment of R Signal)

The BRTR[9:0] bits specify the brightness (DC) adjustment of the R signal.

Brightness correction of the R signal is performed as follows:

- Rout: Output of brightness correction (input of contrast correction), unsigned, 10 bits
- Rin: Input of brightness correction, unsigned, 10 bits
- BRTR: Setting in this bit
- Rout = Rin + BRTR - 512.

BRTB[9:0] bits (Brightness Adjustment of B Signal)

The BRTB[9:0] bits specify the brightness (DC) adjustment of the B signal.

Brightness correction of the B signal is performed as follows:

- Bout: Output of brightness correction (input of contrast correction), unsigned, 10 bits
- Bin: Input of brightness correction, unsigned, 10 bits
- BRTB: Setting in this bit
- Bout = Bin + BRTB - 512.

56.2.47 OUT_CONTRAST : Output Control Block Contrast Correction Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x13D0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	CONTG[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CONTB[7:0]								CONTR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CONTR[7:0]	Contrast Adjustment of R Signal Unsigned 8-bit fixed point value adjusting GAIN on R signal. 0x00: 0/128 = 0.000 ⋮ 0x80: 128/128 = 1.000 ⋮ 0xFF: 255/128 = 1.992	R/W
15:8	CONTB[7:0]	Contrast Adjustment of B Signal Unsigned 8-bit fixed point value adjusting GAIN on B signal. 0x00: 0/128 = 0.000 ⋮ 0x80: 128/128 = 1.000 ⋮ 0xFF: 255/128 = 1.992	R/W
23:16	CONTG[7:0]	Contrast Adjustment of G Signal Unsigned 8-bit fixed point value adjusting GAIN on G signal. 0x00: 0/128 = 0.000 ⋮ 0x80: 128/128 = 1.000 ⋮ 0xFF: 255/128 = 1.992	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when OUT_VLATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

CONTR[7:0] bits (Contrast Adjustment of R Signal)

The CONTR[7:0] bits specify the contrast (GAIN) adjustment of R signal. The location of the decimal point is between bits [7] and [6].

CONTB[7:0] bits (Contrast Adjustment of B Signal)

The CONTB[7:0] bits specify the contrast (GAIN) adjustment of B signal. The location of the decimal point is between bits [7] and [6].

CONTG[7:0] bits (Contrast Adjustment of G Signal)

The CONTG[7:0] bits specify the contrast (GAIN) adjustment of G signal. The location of the decimal point is between bits [7] and [6].

Contrast correction of each pixel is performed as follows (x = R, G, B):

- Dxout: Output of contrast correction, unsigned, 10 bits
- Dxin: Input of contrast correction (output of brightness correction), unsigned, 10 bits
- CONTx: Setting in this bit

- Dxout = Dxin × CONTx.

56.2.48 OUT_PDTHA : Output Control Block Panel Dither Correction Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x13D4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	SEL[1:0]		—	—	FORM[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	PA[1:0]		—	—	PB[1:0]		—	—	PC[1:0]		—	—	PD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PD[1:0]	Pattern Value (D) of 2×2 Pattern Dither Unsigned 2-bit integer.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
5:4	PC[1:0]	Pattern Value (C) of 2×2 Pattern Dither Unsigned 2-bit integer.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
9:8	PB[1:0]	Pattern Value (B) of 2×2 Pattern Dither Unsigned 2-bit integer.	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	PA[1:0]	Pattern Value (A) of 2×2 Pattern Dither Unsigned 2-bit integer.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
17:16	FORM[1:0]	Output Format Select 0 0: RGB888; select RGB888 or serial RGB as output interface format 0 1: RGB666; select RGB666 as output interface format 1 0: RGB565; select RGB565 as output interface format 1 1: Setting prohibited Select output interface format in OUT_SET.FORMAT[1:0].	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
21:20	SEL[1:0]	Operation Mode 0 0: Truncate 0 1: Round-off 1 0: 2×2 pattern dither 1 1: Setting prohibited	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when OUT_VLATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

PA[1:0], PB[1:0], PC[1:0], PD[1:0] bits (Pattern Value (A, B, C, D) of 2×2 Pattern Dither)

The PA[1:0], PB[1:0], PC[1:0], and PD[1:0] bits specify the pattern value A, B, C, and D of 2×2 pattern dither. Figure 56.13 shows the configuration of the dither correction block.

FORM[1:0] bits (Output Format Select)

The FORM[1:0] bits specify the output format of the dither process. These bits must be set in accordance with the OUT_SET.FORMAT[1:0] bits of the Output Interface Register. For serial RGB (OUT_SET.FORMAT[1:0] = 11b), set these bits to 00b. Otherwise, operation is not guaranteed.

SEL[1:0] bits (Operation Mode)

The SEL[1:0] bits specify the dither operation mode. The dither process is performed for the bits equal to or shorter than the pixel data length selected in the output format select bits (OUT_PDTHA.FORM[1:0]). OUT_PDTHA.PA[1:0], PB[1:0], PC[1:0], and PD[1:0] are used for 2×2 pattern dither.

Figure 56.13 shows the configuration of the dither correction block.

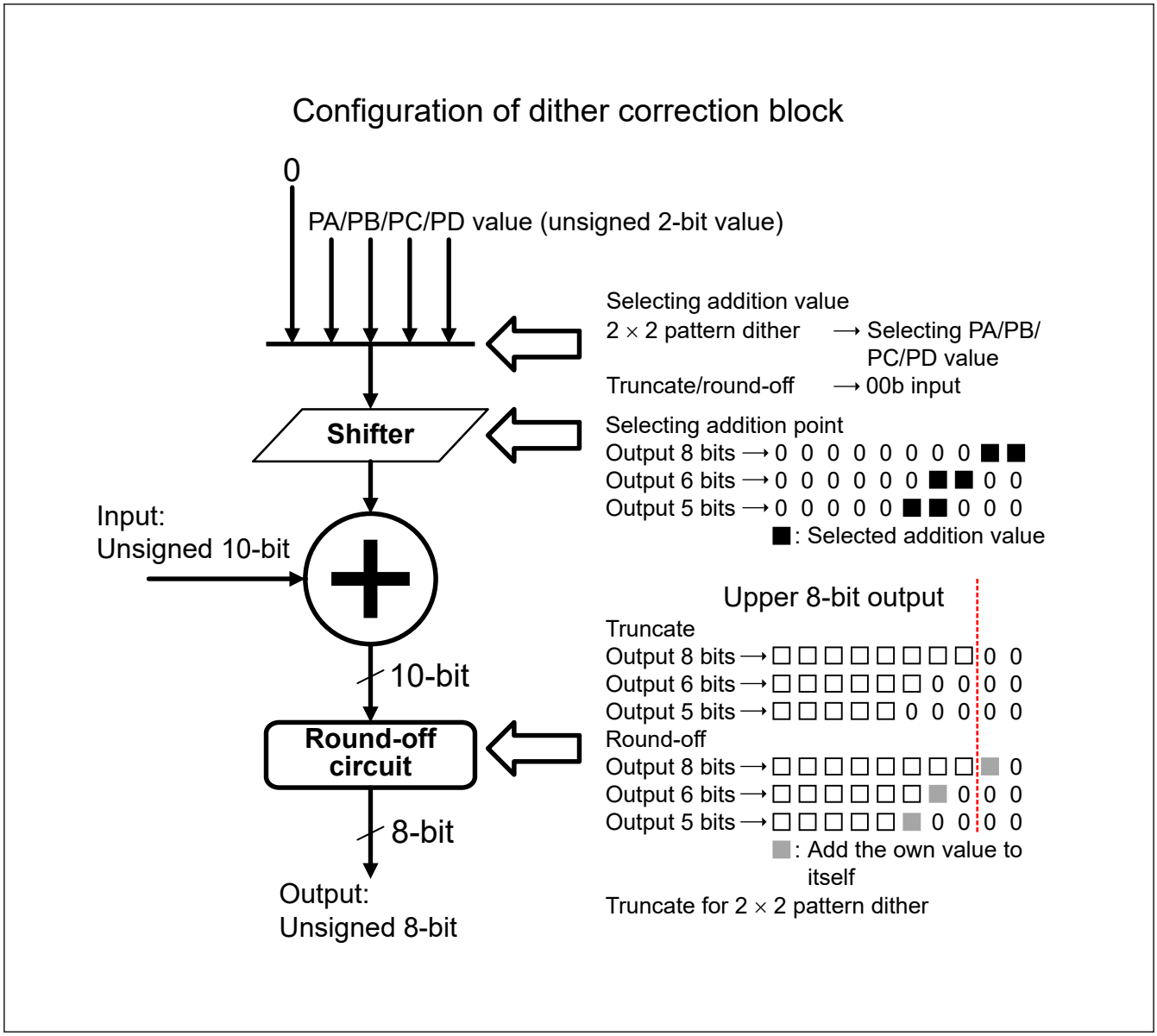


Figure 56.13 Configuration of dither correction block

Figure 56.14 shows the addition value selection method for 2×2 pattern dither.

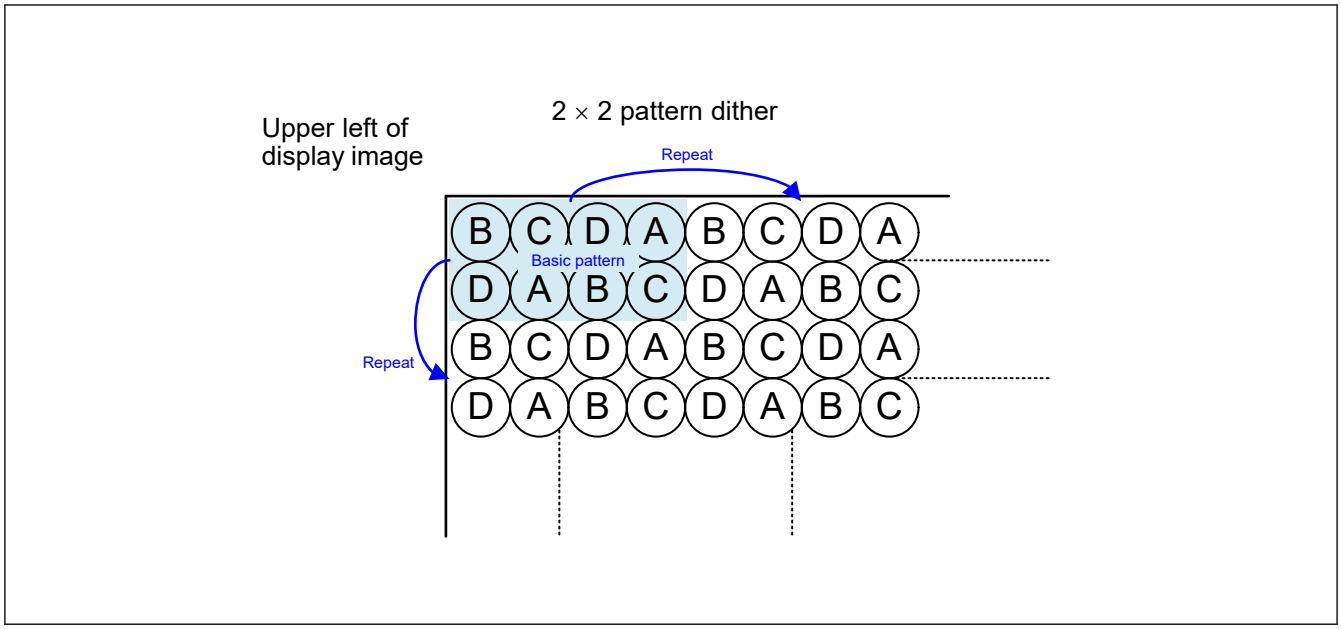


Figure 56.14 Addition value selection method for 2 × 2 pattern dither

A basic pattern is repeated as follows. Four frames constitute 1 cycle.

BCDA DABC	→	CDAB ABCD	→	DABC BCDA	→	ABCD CDAB	Shift to the left by one pixel Shift to the left by one pixel	
1st Frame		2nd Frame		3rd Frame		4th Frame		

- A: Value obtained after OUT_PDTHA.PA[1:0] bit value is shifted in accordance with the output format
- B: Value obtained after OUT_PDTHA.PB[1:0] bit value is shifted in accordance with the output format
- C: Value obtained after OUT_PDTHA.PC[1:0] bit value is shifted in accordance with the output format
- D: Value obtained after OUT_PDTHA.PD[1:0] bit value is shifted in accordance with the output format

Renesas recommends setting the bits as follows: PA[1:0] = 11b, PB[1:0] = 00b, PC[1:0] = 10b, PD[1:0] = 01b

When 2 × 2 pattern dither (OUT_PDTHA.SEL[1:0] = 10b) is to be set, the valid pixel area of the background plane must be an integer multiple of the basic pattern. If serial RGB is selected as the output format for the output control block, add two to the horizontal valid pixel width and set the resulting value to the background plane horizontal valid pixel width bits (BG_HSIZE.HW[10:0]).

56.2.49 OUT_CLKPHASE : Output Control Block Output Phase Control Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x13E4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	FRONTGAM	—	—	—	LCDEDGE	—	TCON0EDGE	TCON1EDGE	TCON2EDGE	TCON3EDGE	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	TCON3EDGE	LCD_TCON3 Output Phase Control 0: Synchronize output with rising edge of LCD_CLK 1: Synchronize output with falling edge of LCD_CLK	R/W
4	TCON2EDGE	LCD_TCON2 Output Phase Control 0: Synchronize output with rising edge of LCD_CLK 1: Synchronize output with falling edge of LCD_CLK	R/W
5	TCON1EDGE	LCD_TCON1 Output Phase Control 0: Synchronize output with rising edge of LCD_CLK 1: Synchronize output with falling edge of LCD_CLK	R/W
6	TCON0EDGE	LCD_TCON0 Output Phase Control 0: Synchronize output with rising edge of LCD_CLK 1: Synchronize output with falling edge of LCD_CLK	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	LCDEDGE	LCD_DATA Output Phase Control 0: Synchronize output with rising edge of LCD_CLK 1: Synchronize output with falling edge of LCD_CLK	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
12	FRONTGAM	Correction Control 0: Process brightness/contrast correction followed by gamma correction 1: Process gamma correction followed by brightness/contrast correction	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when OUT_VLATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

TCON3EDGE bit (LCD_TCON3 Output Phase Control)

The TCON3EDGE bit controls the output phase of LCD_TCON3. When this bit is set to 1, LCD_TCON3 is output in synchronization with the falling edge of LCD_CLK, and when it is set to 0, LCD_TCON3 is output in synchronization with the rising edge.

TCON2EDGE bit (LCD_TCON2 Output Phase Control)

The TCON2EDGE bit controls the output phase of LCD_TCON2. When this bit is set to 1, LCD_TCON2 is output in synchronization with the falling edge of LCD_CLK, and when it is set to 0, LCD_TCON2 is output in synchronization with the rising edge.

TCON1EDGE bit (LCD_TCON1 Output Phase Control)

The TCON1EDGE bit controls the output phase of LCD_TCON1. When this bit is set to 1, LCD_TCON1 is output in synchronization with the falling edge of LCD_CLK, and when it is set to 0, LCD_TCON1 is output in synchronization with the rising edge

TCON0EDGE bit (LCD_TCON0 Output Phase Control)

The TCON0EDGE bit controls the output phase of LCD_TCON0. When this bit is set to 1, LCD_ TCON0 is output in synchronization with the falling edge of LCD_CLK, and when it is cleared to 0, LCD_TCON0 is output in synchronization with the rising edge.

LCDEDGE bit (LCD_DATA Output Phase Control)

The LCDEDGE bit controls the output phase of the LCD_DATA pins (LCD_DATA23 to LCD_DATA00). When this bit is set to 1, the LCD_DATA pins are output in synchronization with the falling edge of LCD_CLK, and when it is set to 0, the LCD_DATA pins are output in synchronization with the rising edge.

FRONTGAM bit (Correction Control)

The FRONTGAM bit controls the correction sequence. When this bit is set to 1, gamma correction is followed by brightness and contrast correction, and when it is set to 0, gamma correction follows brightness and contrast correction. In both cases, each RGB data output from the graphics 2 module is extended from 8 bits to 10 bits (with 00b appended to

the lower side), and is input to the preceding stage of the correction circuit. The output is rounded to 10 bits and is input to the dither correction circuit. Although the sequence of RGB gamma correction and brightness/contrast correction can be reversed using this bit, brightness correction always precedes contrast correction.

56.2.50 TCON_TIM : TCON Reference Timing Setting Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1404

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	HALF[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	OFFSET[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	OFFSET[10:0]	Horizontal Synchronization Signal Generation Reference Timing Offset from the assertion of the internal horizontal synchronization signal in pixels. 0x000: 1 pixel ⋮ 0x7FF: 2048 pixels	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
26:16	HALF[10:0]	Vertical Synchronization Signal Generation Change Timing Delay from the assertion of the internal horizontal synchronization signal in pixels. 0x000: 1 pixel (no delay) ⋮ 0x7FF: 2048 pixels	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

OFFSET[10:0] bits (Horizontal Synchronization Signal Generation Reference Timing)

The OFFSET[10:0] bits specify the reference timing to be used when the horizontal synchronization signal is generated in the TCON. Set the offset from the assertion of the internal horizontal synchronization signal (HS) in terms of pixels. Figure 56.15 shows the horizontal synchronization signal generation reference timing in the TCON.

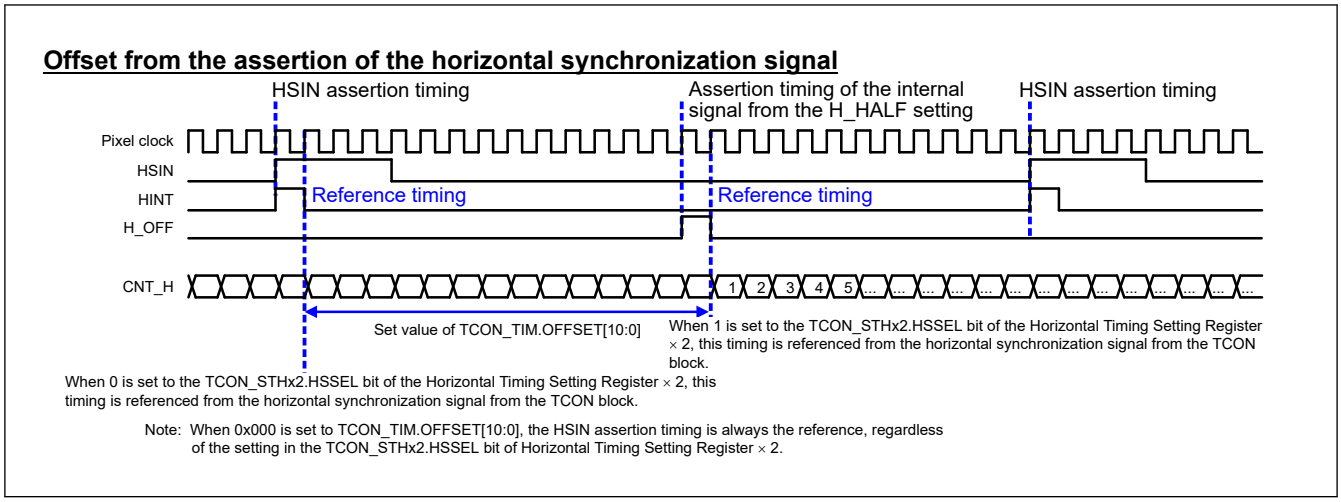


Figure 56.15 Reference timing in the TCON for horizontal synchronization signal generation

HALF[10:0] bits (Vertical Synchronization Signal Generation Change Timing)

The HALF[10:0] bits specify the vertical synchronization signal change timing when the signal is generated in the TCON. Set the change timing as a delay from the assertion of the internal horizontal synchronization signal (HS) in terms of pixels. Figure 56.16 shows the vertical synchronization signal change timing in the TCON.

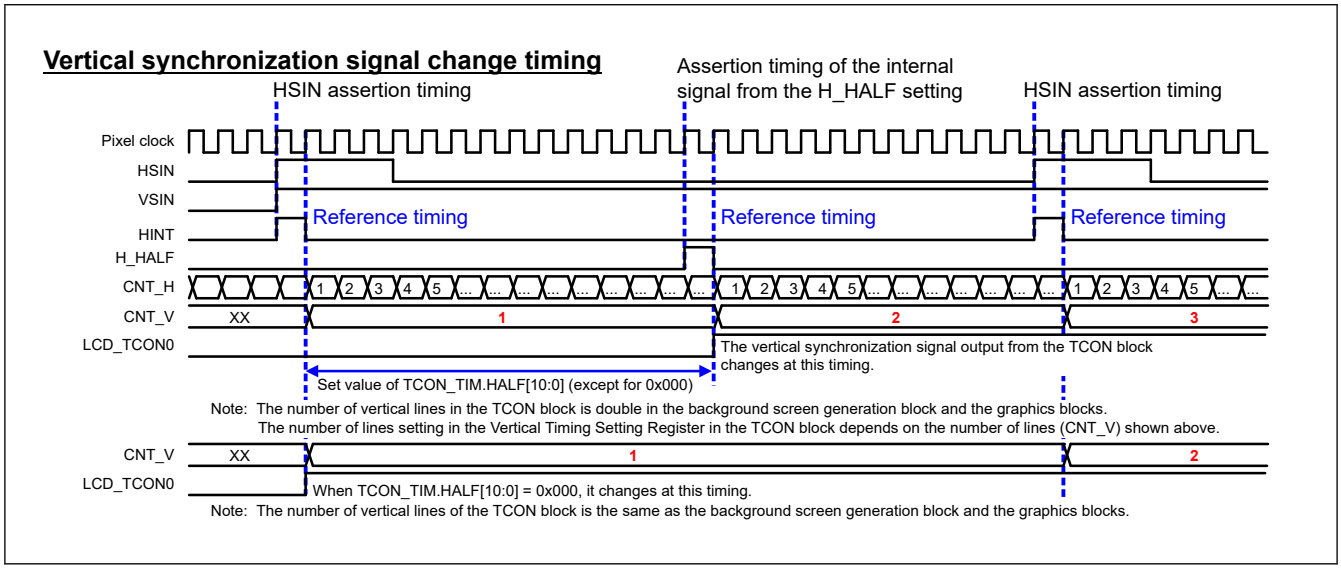


Figure 56.16 Timing in the TCON for vertical synchronization signal change

Also, Figure 56.17 shows the relationship between the TCON block and the output format block in the output control block. These blocks are based on shared control signals (same timing) and image data, and the internal delay is the same. The delay difference specified in the register settings is the source of the timing differences on the external pins.

- TCON block: 3 delay cycles of the pixel clock (PXCLK)
- Data format block:
  - Parallel RGB: 3 delay cycles of the pixel clock (PXCLK)
  - Serial RGB: 3 delay cycles of the pixel clock (PXCLK):
    - The delay of the pixel head data including invalid data
    - No delay of the serial RGB data.

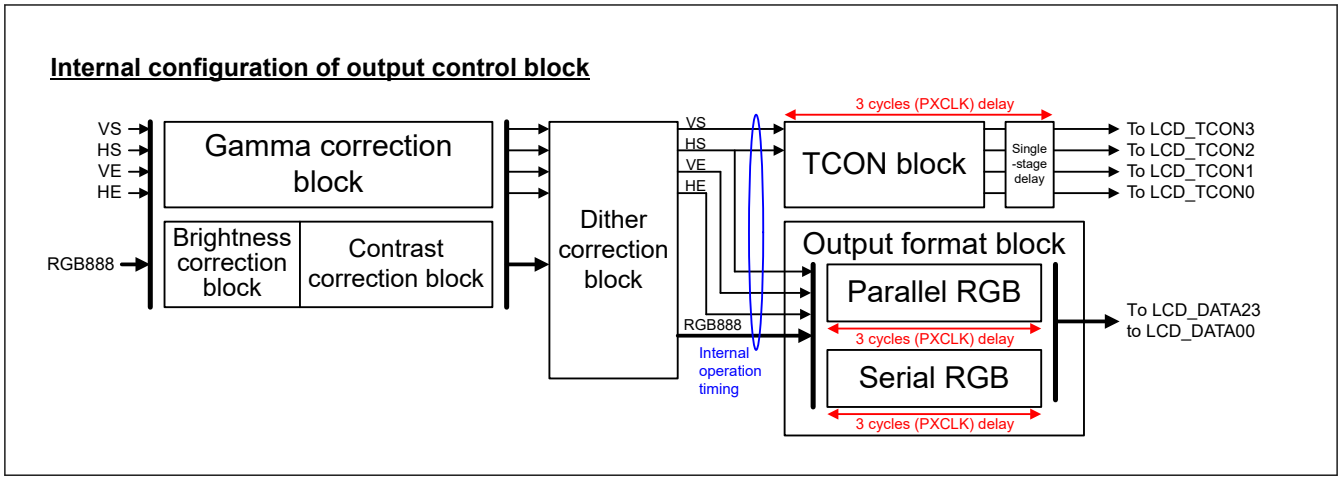


Figure 56.17 Internal configuration of output control block

56.2.51 TCON_STVx1 : TCON Vertical Timing Setting Register x1 (x = A, B)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1408 (TCON_STVA1)
0x1410 (TCON_STVB1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	VS[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	VW[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	VW[10:0]	Vertical Synchronization Signal STVx1 Second Change Timing Signal assertion width in lines. 0x000: 0 line (no vertical synchronization signal assertion) ⋮ 0x7FF: 2047 lines.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
26:16	VS[10:0]	Vertical Synchronization Signal STVx1 First Change Timing Signal delay in lines. 0x000: 0 line (no delay) ⋮ 0x7FF: 2047 lines	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

The vertical timing setting registers (TCON_STVA1/TCON_STVB1 and TCON_STVA2/TCON_STVB2) have the same configuration, and x is either A or B in the descriptions.

VW[10:0] bits (Vertical Synchronization Signal STVx1 Second Change Timing)

The VW[10:0] bits specify the second change (negation) timing of the vertical synchronization signal STVx1, which is generated in the TCON. Set the second change timing as a delay from the first change point in terms of lines. The change position in a horizontal line is defined in TCON_TIM.HALF[10:0] in the Reference Timing Setting Register (TCON_TIM), as is the first change timing.

VS[10:0] bits (Vertical Synchronization Signal STVx1 First Change Timing)

The VS[10:0] bits specify the first change (assertion) timing of the vertical synchronization signal STVx1, which is generated in the TCON. Set the change timing as a delay from the input vertical synchronization signal (VSIN) in terms of lines. The change position in a horizontal line is defined in TCON_TIM.HALF[10:0] in the Reference Timing Setting Register (TCON_TIM), as is the first change timing. Figure 56.18 shows the change timing of vertical synchronization signal to be generated.

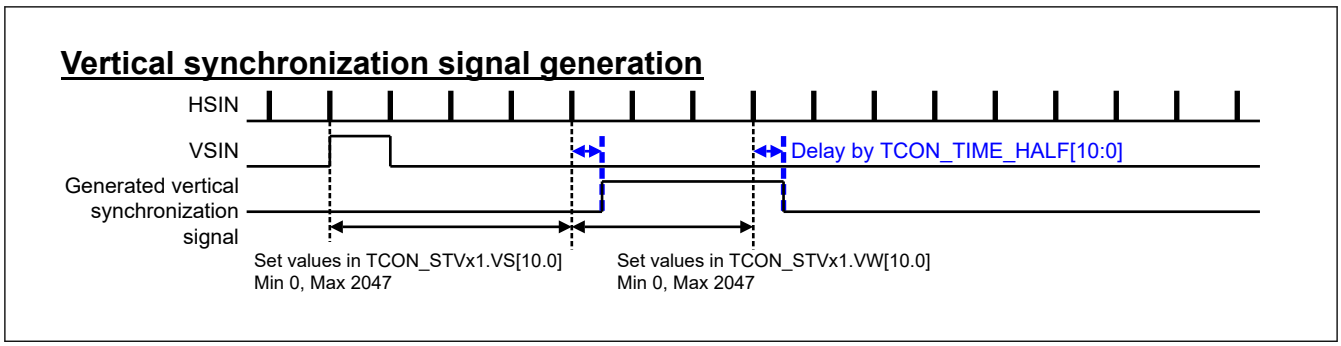


Figure 56.18 Generation of vertical synchronization signal

56.2.52 TCON_STVx2 : TCON Vertical Timing Setting Register x2 (x = A, B)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x140C (TCON_STVA2)
0x1414 (TCON_STVB2)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	INV	—	SEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SEL[2:0]	Output Signal Select Control for LCD_TCON0/LCD_TCON1 Pin Output signal select for LCD_TCON0 pin (controlled in TCON_STVA2 register) and LCD_TCON1 pin (controlled in TCON_STVB2 register). 0 0 0: STVA 0 0 1: STVB 0 1 0: STHA 0 1 1: STHB 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: DE	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	INV	Vertical Synchronization Signal STVx Polarity Inversion Control 0: Do not invert 1: Invert	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

The vertical timing setting registers (TCON_STVA1/TCON_STVB1 and TCON_STVA2/TCON_STVB2) have the same configuration, and x is either A or B in the descriptions.

SEL[2:0] bits (Output Signal Select Control for LCD_TCON0/LCD_TCON1 Pin)

The SEL[2:0] bits control output signal select for the LCD_TCON0/LCD_TCON1 pin. Figure 56.19 shows the configuration of the inversion control and output signal select.

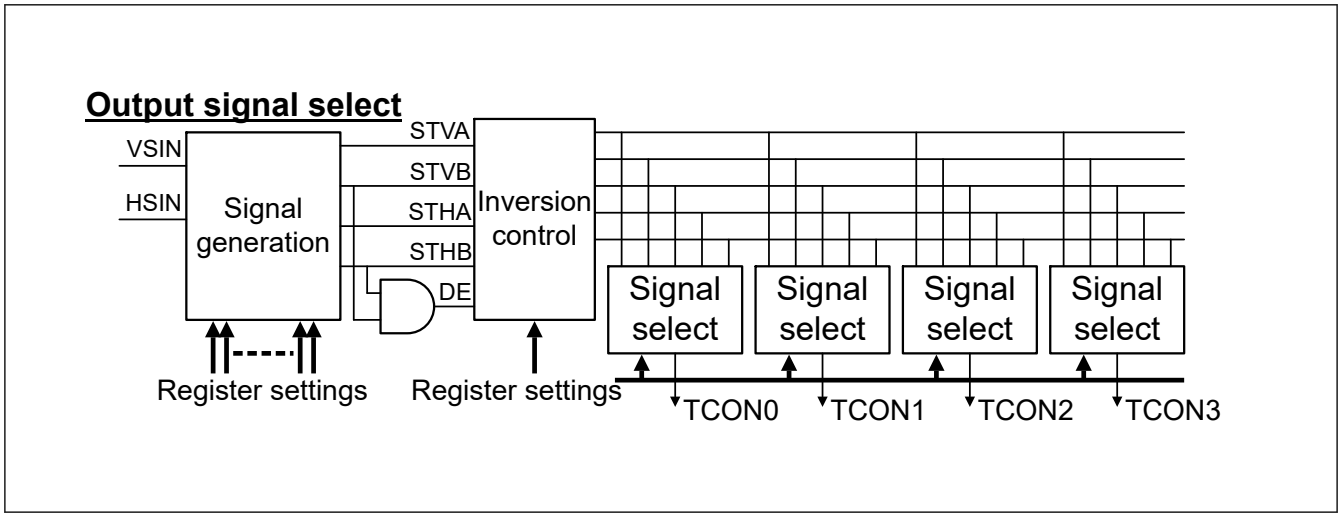


Figure 56.19 Configuration of inversion control and output signal selects

INV bit (Vertical Synchronization Signal STVx Polarity Inversion Control)

The INV bit controls polarity inversion of the vertical synchronization signal (STVx).

56.2.53 TCON_STHx1 : TCON Horizontal Timing Setting Register x1 (x = A, B)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1418 (TCON_STHA1)
0x1420 (TCON_STHB1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	HS[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	HW[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	HW[10:0]	Horizontal Synchronization Signal STHx1 Second Change Timing Signal assertion width in pixels. 0x000: 0 pixel (no horizontal synchronization signal assertion) ⋮ 0x7FF: 2047 pixels	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
26:16	HS[10:0]	Horizontal Synchronization Signal STHx1 First Change Timing Signal delay in pixels. 0x000: 0 pixel (no delay) ⋮ 0x7FF: 2047 pixels	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

The horizontal timing setting registers (TCON_STHA1/TCON_STHB1 and TCON_STHA2/TCON_STHB2) have the same configuration, and x is either A or B in the descriptions.

HW[10:0] bits (Horizontal Synchronization Signal STHx1 Second Change Timing)

The HW[10:0] bits specify the second change (negation) timing of the horizontal synchronization signal STHx1, which is generated in the TCON. Set the second change timing as a distance from the first change point in terms of pixels.

HS[10:0] bits (Horizontal Synchronization Signal STHx1 First Change Timing)

The HS[10:0] bits specify the first change (assertion) timing of the horizontal synchronization signal STHx1, which is generated in the TCON. Set the change timing as a distance from the input horizontal synchronization signal (HSIN) or the reference timing based on the offset specified in the TCON_TIM.OFFSET[10:0] bit (horizontal synchronization signal generation reference timing) in terms of pixels.

Figure 56.20 shows the horizontal synchronization signal generation timing if the input horizontal synchronization signal (HSIN) is based on the negated edge reference of the HINT signal. Figure 56.21 shows the horizontal synchronization signal generation timing after offset. By setting the TCON_TIM.OFFSET[10:0] bit (horizontal synchronization generation reference timing) and these horizontal synchronization bits appropriately, it is possible to generate a signal that is asserted before HSIN and negated after HSIN, where HSIN is the horizontal synchronization signal input to the TCON.

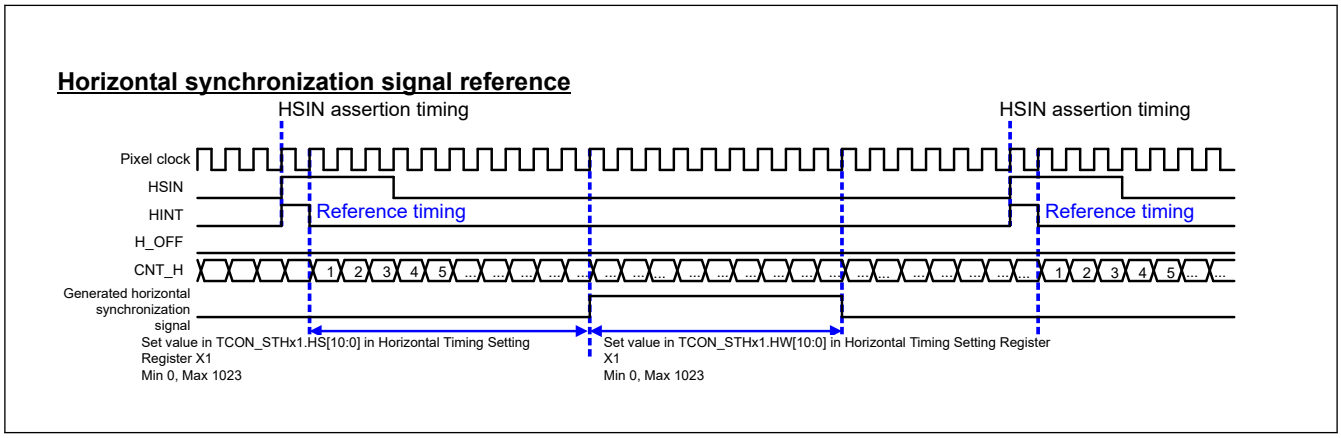


Figure 56.20 Signal generation based on input horizontal synchronization signal (HSIN)

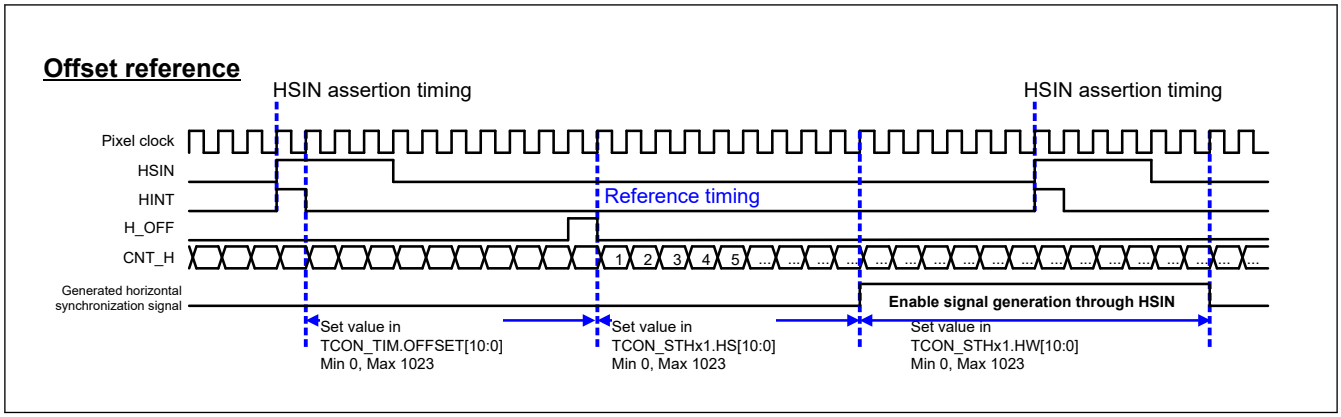


Figure 56.21 Horizontal synchronization signal generation based on offset

When generating the signal across HSIN on the offset reference, the horizontal synchronization signal of the last line of the frame spans the first line of the next frame. Even if the BG_EN.EN bit is cleared to 0 and the GLCDC operation is stopped, the horizontal synchronization signal across HSIN is not cleared at the frame end and retains the predetermined value until the second change timing set in the registers. If BG_EN.SWRST is cleared to 0, the signal returns to the initial state immediately.

56.2.54 TCON_STHx2 : TCON Horizontal Timing Setting Register x2 (x = A, B)

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x141C (TCON_STHA2)
0x1424 (TCON_STHB2)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	HSSEL	—	—	—	INV	—	SEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SEL[2:0]	Output Signal Select Control for LCD_TCON2/LCD_TCON3 Pin Output signal select for LCD_TCON2 pin (controlled in TCON_STHA2 register) and LCD_TCON3 pin (controlled in TCON_STHB2 register). 0 0 0: STVA 0 0 1: STVB 0 1 0: STHA 0 1 1: STHB 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: DE	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	INV	Horizontal Synchronization Signal STHx Polarity Inversion Control 0: Do not invert 1: Invert	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	HSSEL	Horizontal Synchronization Signal STHx Reference Timing Control 0: Select input horizontal synchronization signal (HSIN) as reference for signal generation 1: Select offset specified in TCON_TIM.OFFSET[10:0] (horizontal synchronization generation reference timing) as reference for signal generation	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

The horizontal timing setting registers (TCON_STHA1/TCON_STHB1 and TCON_STHA2/TCON_STHB2) have the same configuration, and x is either A or B in the descriptions.

SEL[2:0] bits (Output Signal Select Control for LCD_TCON2/LCD_TCON3 Pin)

The SEL[2:0] bits control the output signal select for the LCD_TCON2/LCD_TCON3 pins.

INV bit (Horizontal Synchronization Signal STHx Polarity Inversion Control)

The INV bit controls polarity inversion of the horizontal synchronization signal (STHx).

HSSEL bit (Horizontal Synchronization Signal STHx Reference Timing Control)

The HSSEL bit selects the reference timing for generating the horizontal synchronization signal STHx. For details on the signal to be generated, see Figure 56.20 and Figure 56.21. For the configuration of the inversion control and output signal select, see Figure 56.19.

56.2.55 TCON_DE : TCON Data Enable Polarity Setting Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1428

Bit position:	31																													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INV
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	INV	Data Enable Signal DE Polarity Inversion Control 0: Do not invert 1: Invert	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

INV bit (Data Enable Signal DE Polarity Inversion Control)

The INV bit controls polarity inversion of the data enable signal DE. The data enable signal DE generated in the TCON is the logical AND of the STVB and STHB signals.

56.2.56 SYSCNT_DTCTEN : System Control Block State Detection Control Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1440

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	L2UNDFDTC	L1UNDFDTC	VPOSDTC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VPOSDTC	Specified Line Detection Control 0: Disable detection of specified line 1: Enable detection of specified line*1	R/W
1	L1UNDFDTC	Graphics 1 Underflow Detection Control 0: Disable detection of graphics 1 underflow 1: Enable detection of graphics 1 underflow*2	R/W
2	L2UNDFDTC	Graphics 2 Underflow Detection Control 0: Disable detection of graphics 2 underflow 1: Enable detection of graphics 2 underflow*2	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Set the VPOSDTC bit to 1 after setting the BG_EN.EN bit to 1.
Note 2. When setting the LnUNDFDTC (n = 1, 2) bit to 1 and when the BG_SYNC.VP[3:0] bits are set to a value greater than 0x5, an unexpected GLCDC_LnUNDF (n = 1, 2) interrupt is generated after the GLCDC starts.Therefore, set the SYSCNT_STCLR.LnUNDFCLR (n = 1, 2) bit to 1, then set the SYSCNT_STMON.LnUNDF (n = 1, 2) bit to 0 to clear the unexpected GLCDC_LnUNDF (n = 1, 2) interrupt.

VPOSDTC bit (Specified Line Detection Control)

The VPOSDTC bit enables or disables detection of the specified line. When this bit is set to 1, the associated bit in the SYSCNT_STMON register sets to 1 on event notification from graphics 2. When it is set to 0, the associated bit in the SYSCNT_STMON register does not set to 1 even on event notification from graphics 2.

L1UNDFDTC bit (Graphics 1 Underflow Detection Control)

The L1UNDFDTC bit enables or disables detection of the graphics 1 underflow. When this bit is set to 1, the associated bit in the SYSCNT_STMON register sets to 1 when an underflow occurs in graphics 1. When it is set to 0, the associated bit in the SYSCNT_STMON register does not set to 1 even when an underflow occurs in graphics 1. The underflow state in graphics 1 is automatically cleared on assertion of the vertical synchronization signal (VS) regardless of the value of this bit, and normal operation is recovered.

L2UNDFDTC bit (Graphics 2 Underflow Detection Control)

The L2UNDFDTC bit enables or disables detection of the graphics 2 underflow. When this bit is set to 1, the associated bit in the SYSCNT_STMON register sets to 1 when an underflow occurs in graphics 2. When it is set to 0, the associated bit in the SYSCNT_STMON register does not set to 1 even when an underflow occurs in graphics 2. The underflow state in graphics 2 is automatically cleared on assertion of the vertical synchronization signal (VS) regardless of the value of this bit, and normal operation is recovered.

56.2.57 SYSCNT_INTEN : System Control Block Interrupt Request Enable Control Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1444

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	L2UNDFINTEN	L1UNDFINTEN	VPOSINTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VPOSINTEN	Interrupt Request Signal GLCDC_VPOS Enable Control 0: Disable GLCDC_VPOS output 1: Enable GLCDC_VPOS output	R/W
1	L1UNDFINTEN	Interrupt Request Signal GLCDC_L1UNDF Enable Control 0: Disable GLCDC_L1UNDF output 1: Enable GLCDC_L1UNDF output	R/W
2	L2UNDFINTEN	Interrupt Request Signal GLCDC_L2UNDF Enable Control 0: Disable GLCDC_L2UNDF output 1: Enable GLCDC_L2UNDF output	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

VPOSINTEN bit (Interrupt Request Signal GLCDC_VPOS Enable Control)

The VPOSINTEN bit enables or disables the interrupt request signal GLCDC_VPOS. When this bit is set to 1, the interrupt request signal GLCDC_VPOS is asserted when the associated status monitor flag SYSCNT_STMON[0] sets. When it is cleared to 0, the interrupt request signal GLCDC_VPOS is not asserted even when the associated status monitor flag SYSCNT_STMON[0] sets. If this bit is cleared during GLCDC_VPOS assertion, the associated status monitor flag SYSCNT_STMON[0] does not change, but the interrupt request signal GLCDC_VPOS is negated.

L1UNDFINTEN bit (Interrupt Request Signal GLCDC_L1UNDF Enable Control)

The L1UNDFINTEN bit enables or disables the interrupt request signal GLCDC_L1UNDF. When this bit is set to 1, the interrupt request signal GLCDC_L1UNDF is asserted when the associated status monitor flag SYSCNT_STMON[1] sets. When it is cleared to 0, the interrupt request signal GLCDC_L1UNDF is not asserted even when the associated status monitor flag SYSCNT_STMON[1] sets. If this bit is cleared during GLCDC_L1UNDF assertion, the associated status monitor flag SYSCNT_STMON[1] does not change, but the interrupt request signal GLCDC_L1UNDF is negated.

L2UNDFINTEN bit (Interrupt Request Signal GLCDC_L2UNDF Enable Control)

The L2UNDFINTEN bit enables or disables the interrupt request signal GLCDC_L2UNDF. When this bit is set to 1, the interrupt request signal GLCDC_L2UNDF is asserted when the associated status monitor flag SYSCNT_STMON[2] sets. When it is cleared to 0, the interrupt request signal GLCDC_L2UNDF is not asserted even when the associated status monitor flag SYSCNT_STMON[2] sets. If this bit is cleared during GLCDC_L2UNDF assertion, the associated status monitor flag SYSCNT_STMON[2] does not change, but the interrupt request signal GLCDC_L2UNDF is negated.

56.2.58 SYSCNT_STCLR : System Control Block Status Clear Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1448

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	L2UNDFCLR	L1UNDFCLR	VPOSCLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VPOSCLR	Graphics 2 Specified Line Detection Flag Clear 0: No operation 1: Clear the graphics 2 specified line detection flag	R/W*1
1	L1UNDFCLR	Graphics 1 Underflow Detection Flag Clear 0: No operation 1: Clear the graphics 1 underflow detection flag	R/W*1
2	L2UNDFCLR	Graphics 2 Underflow Detection Flag Clear 0: No operation 1: Clears the graphics 2 underflow detection flag	R/W*1
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. These bits are read as 0.

VPOSCLR bit (Graphics 2 Specified Line Detection Flag Clear)

Writing 1 to the VPOSCLR bit clears the graphics 2 specified line detection flag. Clearance of the flag by this bit is only valid for the associated flag, and does not directly affect the other states and interrupt request signals. However, an interrupt request signal might be negated by clearing the detection flag.

L1UNDFCLR bit (Graphics 1 Underflow Detection Flag Clear)

Writing 1 to the L1UNDFCLR bit clears the graphics 1 underflow detection flag. Clearance of the flag by this bit is only valid for the associated flag, and does not directly affect the other states and interrupt request signals. However, an interrupt request signal might be negated by clearing the detection flag.

L2UNDFCLR bit (Graphics 2 Underflow Detection Flag Clear)

Writing 1 to the L2UNDFCLR bit clears the graphics 2 underflow detection flag. Clearance of the flag by this bit is only valid for the associated flag, and does not directly affect the other states and interrupt request signals. However, an interrupt request signal might be negated by clearing the detection flag.

56.2.59 SYSCNT_STMON : System Control Block Status Monitor Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x144C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	L2UNDF	L1UNDF	VPOS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VPOS	Graphics 2 Specified Line Detection Flag 0: Specified line notification not detected in graphics 2 1: Specified line notification detected in graphics 2	R
1	L1UNDF	Graphics 1 Underflow Detection Flag 0: No underflow detected in graphics 1 1: Underflow detected in graphics 1	R
2	L2UNDF	Graphics 2 Underflow Detection Flag 0: No underflow detected in graphics 2 1: Underflow detected in graphics 2	R
31:3	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

VPOS flag (Graphics 2 Specified Line Detection Flag)

The VPOS flag indicates that the specified line notification was detected in graphics 2. When this flag is 1, it indicates that the specified line notification was detected in graphics 2 at some time in the past. It does not necessarily mean that graphics 2 is currently processing the specified line. When this flag is 0, it indicates that no specified line notification was detected after the module operation was enabled. When the interrupt request signal is enabled and is cleared while this flag is 1, the associated interrupt request signal GLCDC_VPOS is negated, but this does not affect the state of graphics 2.

L1UNDF flag (Graphics 1 Underflow Detection Flag)

The L1UNDF flag indicates that an underflow was detected in graphics 1. When this flag is 1, it indicates that an underflow was detected in graphics 1 at some time in the past. It does not necessarily mean that graphics 1 is currently in the underflow state. When this flag is 0, it indicates that no underflow was detected after the module operation was enabled. When the interrupt request signal is enabled and is cleared while this flag is 1, the associated interrupt request signal GLCDC_L1UNDF is negated, but this does not affect the state of graphics 1.

Even if the current graphics data is not required (GR1_AB1.DISPSEL[1:0] = 0xb), if the SYSCNT_DTCTEN.L1UNDFDTC flag (graphics 1 underflow detection control) is 1 and detection is enabled, this flag is set to 1 at the start of the graphics image valid area. To avoid unnecessary detection flag settings or interrupt request signal assertions, when the display does not require the current graphics data, set the SYSCNT_DTCTEN.L1UNDFDTC flag and the SYSCNT_INTEN.L1UNDFINTEN bit (interrupt request signal GLCDC_L1UNDF enable control) to 0.

L2UNDF flag (Graphics 2 Underflow Detection Flag)

The L2UNDF flag indicates that an underflow was detected in graphics 2. When this flag is 1, it indicates that an underflow was detected in graphics 2 at some time in the past. It does not necessarily mean that graphics 2 is currently in the

underflow state. When this flag is 0, it indicates that no underflow was detected after the module operation was enabled. When the interrupt request signal is enabled and is cleared while this flag is 1, the associated interrupt request signal GLCDC_L2UNDF is negated, but this does not affect the state of graphics 2.

Even if the current graphics data is not required (GR2_AB1.DISPSEL[1:0] = 0xb), if the SYSCNT_DTCTEN.L2UNDFDTC flag (graphics 2 underflow detection control) is 1 and detection is enabled, this flag is set to 1 at the start of the graphics image valid area. To avoid the unnecessary detection flag settings or interrupt request signal assertions, when the display does not require the current graphics data, set the SYSCNT_DTCTEN.L2UNDFDTC flag and the SYSCNT_INTEN.L2UNDFINTEN bit (interrupt request signal GLCDC_L2UNDF enable control) to 0.

56.2.60 SYSCNT_PANEL_CLK : System Control Block Version and Panel Clock Control Register

Base address: GLCDC = 0x4034_2000
GLCDC_NS = 0x5034_2000

Offset address: 0x1450

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	VER[15:0]															
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PIXSEL	—	—	—	CLKSEL	—	CLKEN	DCDR[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	DCDR[5:0]	Clock Division Ratio Setting Control See Table 56.9 for details on these settings.	R/W
6	CLKEN	Panel Clock Output Enable Control 0: Disable panel clock output 1: Enable panel clock output Before changing the PIXSEL, CLKSEL, or DCDR bit, this bit must be set to 0.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	CLKSEL	Panel Clock Supply Source Control 0: Select external clock (LCD_EXTCLK) 1: Select LCDCLK	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
12	PIXSEL	Pixel Clock Select Control 0: Select no frequency division, parallel RGB 1: Select quarter frequency, serial RGB This setting must have the same value as OUT_SET.FRQSEL[1].	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
31:16	VER[15:0]	Version Information Version information of the GLCDC.	R

Note: S-TYPE-3, P-TYPE-3

The configuration of the pixel and panel clock generator circuits are shown in section 8, Clock Generation Circuit.

DCDR[5:0] bits (Clock Division Ratio Setting Control)

The DCDR[5:0] bits control the setting of the panel clock division ratio. The division ratio bit can only be set to the values listed in Table 56.9. Operation is not guaranteed for values not listed.

Table 56.9 Input clock division (1 of 2)

DCDR[5:0]	Clock division ratio	CLKSEL = 0, LCD_EXTCLK (≤ 60 MHz)	CLKSEL = 1, LCDCLK (Up to 240 MHz)
000000b	1/2	LCD_EXTCLK/2	LCDCLK/2*1

Table 56.9 Input clock division (2 of 2)

DCDR[5:0]	Clock division ratio	CLKSEL = 0, LCD_EXTCLK (≤ 60 MHz)	CLKSEL = 1, LCDCLK (Up to 240 MHz)
000001b	1/1	LCD_EXTCLK*1	LCDCLK*1
000010b	1/2	LCD_EXTCLK/2	LCDCLK/2*1
000011b	1/3	LCD_EXTCLK/3	LCDCLK/3*1
000100b	1/4	LCD_EXTCLK/4	LCDCLK/4*1
000101b	1/5	LCD_EXTCLK/5	LCDCLK/5
000110b	1/6	LCD_EXTCLK/6	LCDCLK/6
000111b	1/7	LCD_EXTCLK/7	LCDCLK/7
001000b	1/8	LCD_EXTCLK/8	LCDCLK/8
001001b	1/9	LCD_EXTCLK/9	LCDCLK/9
001100b	1/12	LCD_EXTCLK/12	LCDCLK/12
010000b	1/16	LCD_EXTCLK/16	LCDCLK/16
011000b	1/24	LCD_EXTCLK/24	LCDCLK/24
100000b	1/32	LCD_EXTCLK/32	LCDCLK/32

Note 1. The panel clock is output as the LCD_CLK output clock. This setting may be prohibited because LCD_EXTCLK and LCD_CLK have limited frequencies. See section 60, Electrical Characteristics.

To set the panel clock:

1. After setting the input source of the panel clock in the CLKSEL bit, set the division ratio in the DCDR[5:0] bits and the pixel clock selection.
2. Set the CLKEN bit to 1.

CLKEN bit (Panel Clock Output Enable Control)

The CLKEN bit enables or disables the panel clock output. When enabling the panel clock output and operating the panel clock block, set this bit to 1. When changing the PIXSEL, CLKSEL, or DCDR bit, you must set this bit to 0 once and stop the panel clock output. Operation is not guaranteed if any setting is changed while the panel clock is being output.

CLKSEL bit (Panel Clock Supply Source Control)

The CLKSEL bit controls the selection of the panel clock supply source from either the external clock pin (LCD_EXTCLK) or LCDCLK. When the external clock is selected, set this bit to 0. When LCDCLK is selected, set this bit to 1.

PIXSEL bit (Pixel Clock Select Control)

The PIXSEL bit controls the selection of the pixel clock output. When selecting parallel RGB, set this bit to 0 to output the same frequency as the panel clock (no frequency division). When selecting serial RGB, set this bit to 1 to output the quarter frequency of the panel clock as the pixel clock.

This bit must be synchronized with the OUT_SET.FRQSEL[1:0] setting. You must set the same value as in FRQSEL[1]. Otherwise, operation is not guaranteed.

VER[15:0] bits (Version Information)

The VER[15:0] bits provide GLCDC version information.

56.3 Operation

56.3.1 Overall Control

The GLCDC consists of six modules as shown in Figure 56.22, each of which functions independently. The four modules handling image data are interconnected by the vertical and horizontal synchronization signals VS, HS, VE, and HE, and image data (RGB888), as shown in Figure 56.23. The processing of image data is carried out with the pixel clock (PXCLK). LCD_CLK synchronizes with PXCLK (also in phase with each other), and has the same or quadruple frequency (for serial RGB888 output). The registers controlling operation and setting parameters are connected with the PABI and operate on PCLKA. The circuits, including the data buffer and the CLUT memory, operate on PCLKA to read graphics data from

the GLCDC0 and GLCDC1 bus, access the color palette (CLUT) memory, and expand graphics data into the ARGB8888 format.

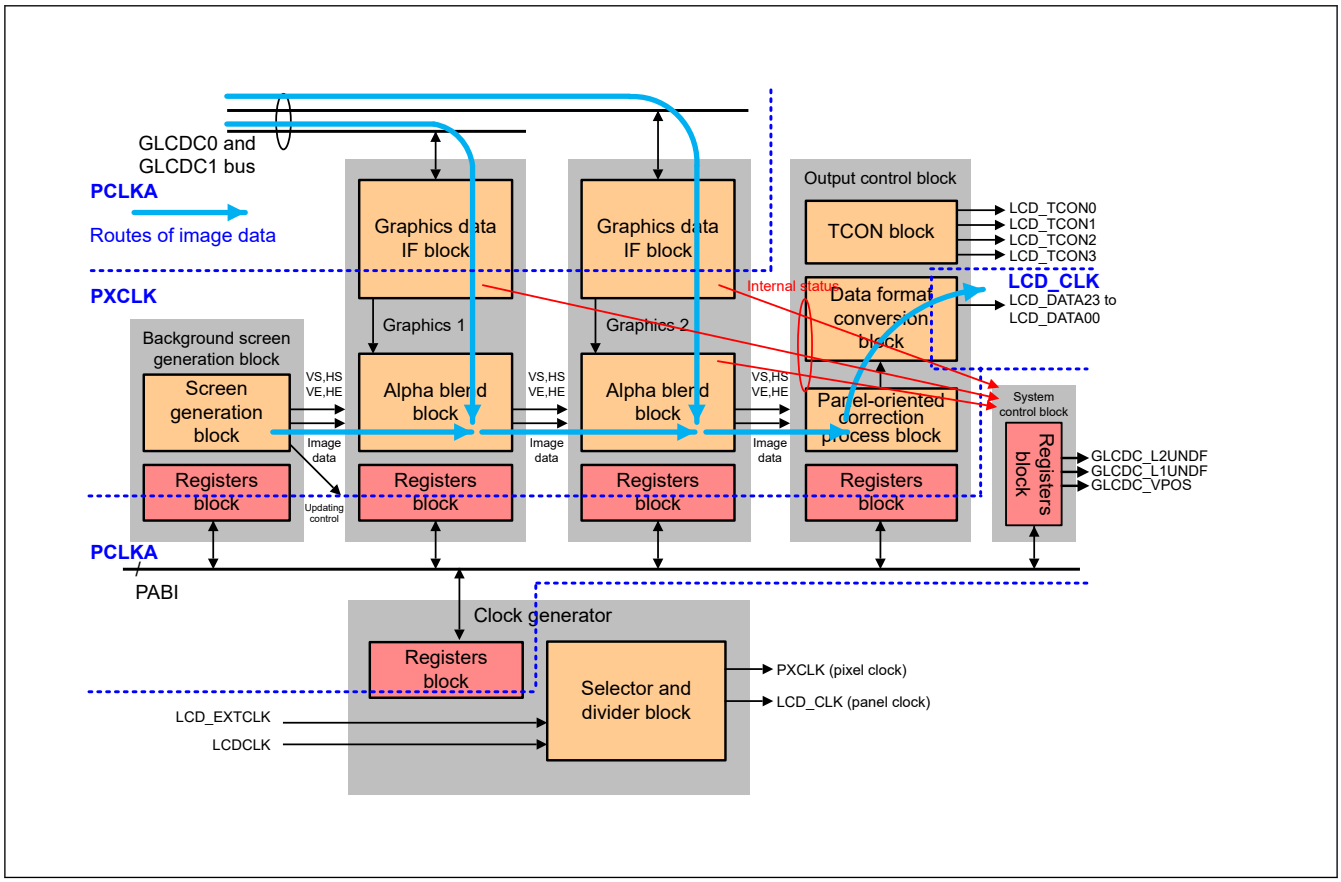


Figure 56.22 GLCDC configuration

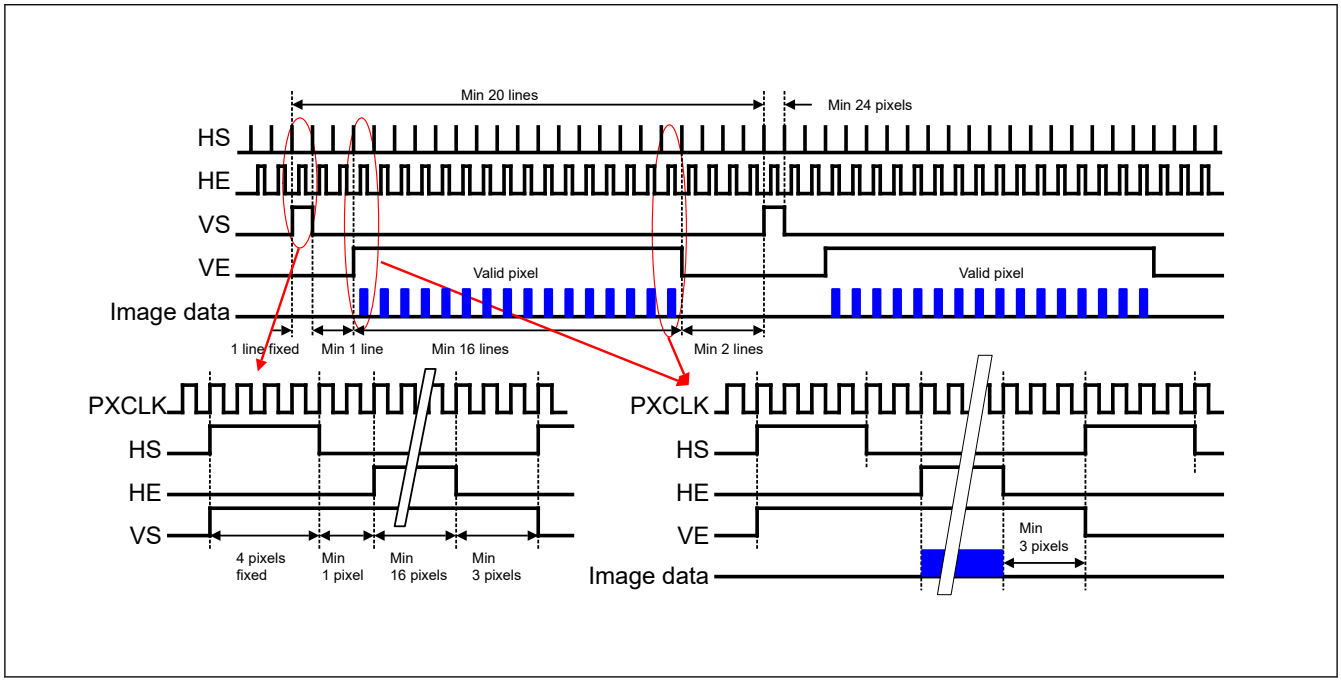


Figure 56.23 Image data carrier signals in the GLCDC

The graphics 1, graphics 2, and output control modules have no individual operation enable bits. By clearing the software reset state in the background screen generation module, these modules wait for vertical and horizontal synchronization

signals VS, HS, VE, and HE, and image data (RGB888) to be input, detect assertion of the vertical synchronization signal VS, and start operation for each frame based on the values preset in the registers. For the system control module, clearing the software reset state in the background screen generation module allows it to monitor the status of graphics 1 and 2 in accordance with the register settings and assert interrupt request signals. These status flags are intended for observing the internal status and exert no influence on the internal operation. The operation between startup and stop of the GLCDC is outlined in this section.

(1) Startup

To start GLCDC operation:

1. Confirm that PCLKA is supplied to the GLCDC and that the reset is negated. When setting LCD_CLK to the same frequency as PXCLK or LCD_CLK to the quadruple speed clock, the setting must be consistent with the value in the register specified in the subsequent stage. The quadruple speed clock is required only when the output format is serial RGB888. Operation is not guaranteed if the quadruple speed clock is supplied to LCD_CLK in other output formats than serial RGB888.
2. Set 1 to the BG_EN.SWRST bit to release the entire GLCDC from a software reset.
3. Set parameters necessary for operation in each register. Although registers of any module can be set first, while the BG_EN.VEN bit (control of GLCDC internal register value reflection to internal operations), BG_EN.EN bit (background plane operation enable) of the background screen generation module, and the VEN bit in the register update control register of each module remain cleared to 0, confirm that PCLKA and PXCLK/LCD_CLK are supplied and the BG_MON.SWRST bit (entire module software reset state monitor) is set to 1.
4. When displaying the graphics data read by graphics 1 and 2 through the GLCDC0 and GLCDC1 bus, set 1 to the GRn_FLMRD.RENB bit (graphics data read enable).
5. Write the color palette data of graphics 1 and 2 (0 and 1 planes) to the CLUT memory through the register access bus (PABI) as required. This is necessary for the LUT1, LUT4, LUT8, and ARGB1555 data formats. ARGB1555 uses addresses 0x80 and 0x00 on the two planes, LUT1 uses 0x01 and 0x00, LUT4 uses the addresses from 0x0F to 0x00, and LUT8 uses the addresses from 0xFF to 0x00.
6. Set 1 to the BG_EN.VEN bit (control of background plane register value reflection to internal operations) and the BG_EN.EN bit (background plane operation enable) of the background screen generation module at the same time. This setting allows output of the vertical and horizontal synchronization signals VS, HS, VE, and HE, and the image data (RGB888) from the background screen generation module. When the pixel data is valid (both VE and HE are 1), output from the background screen generation module uses the value in the BG_BGC register. The data value is 0x00000000 for pixels that are not valid.
7. Each module detects the assertion of the vertical synchronization signal (VS) output from its previous stage (background screen generation module for graphics 1, graphics 1 for graphics 2, and graphics 2 for the output control module) and starts operation in accordance with its register settings. All the modules control the operation in frame units. When the assertion of the vertical synchronization signal (VS) is detected, the current frame is taken as the frame head and the status is initialized. If necessary (when the VEN bits in the register update control registers are 1), the register values read through the register access bus are reflected to the internal operation.

(2) Changing parameters during operation

With the GLCDC, it is possible to update parameters of each module and reflect the updates to the internal operations during operation, without preventing graphics data to be read by the background screen generation module, graphics 1, or graphics 2. By setting 1 to the VEN bits of the modules, including the background screen generation module, almost all parameters are reflected to the internal operations at the start of the following frame (at the start of the control screen for the background screen generation module, and immediately after the VS assertion of the previous stage for the other modules). However, if the CLUT plane being used for internal operations (determined in the GRn_CLUTINT.SEL[1:0] bits) is modified, the updates are reflected to the internal operations immediately without waiting for the following VS assertion. To circumvent this immediate reflection of the CLUT plane modification to the internal operations, first write all the image data (ARGB8888) necessary for the CLUT plane that is not being used for internal operations, next modify the GRn_CLUTINT.SEL[1:0] bits, and finally set 1 to the VEN bit of the background screen generation module or the target module.

To modify parameters during operation:

1. Confirm that the VEN bit of each module is 0. Operation is not guaranteed if the target registers are modified when the VEN bit of the module to which the register values are to be reflected to the internal operations, or the VEN bit of the background screen generation module is 1.
2. Modify the value of the target registers.
3. If only a particular module is to be the target, set 1 to the VEN bit of the target module. If multiple modules or the background screen generation module are to be the targets, set 1 to the VEN bit of the background screen generation module. In this case, all the modules are included as targets, not only the background screen generation module.
4. Confirm that the VEN bit to which 1 was set is 0. If the bit is cleared to 0, the target register contents are reflected to the internal operations. If the bit remains 1, however, the target register contents might not yet be reflected to the internal operations. The VEN bit of each module is cleared to 0 immediately after the target register values are internally reflected. However, the VEN bit of the background screen generation module is not cleared to 0 until the module output VE is negated (with all the modules, sufficient delay is secured in reference to the VS assertion of the background screen generation module so that the register values are reflected to the internal operations for the same frame). Figure 56.24 shows the operation of the signals output by the background screen generation module and the monitor bits.

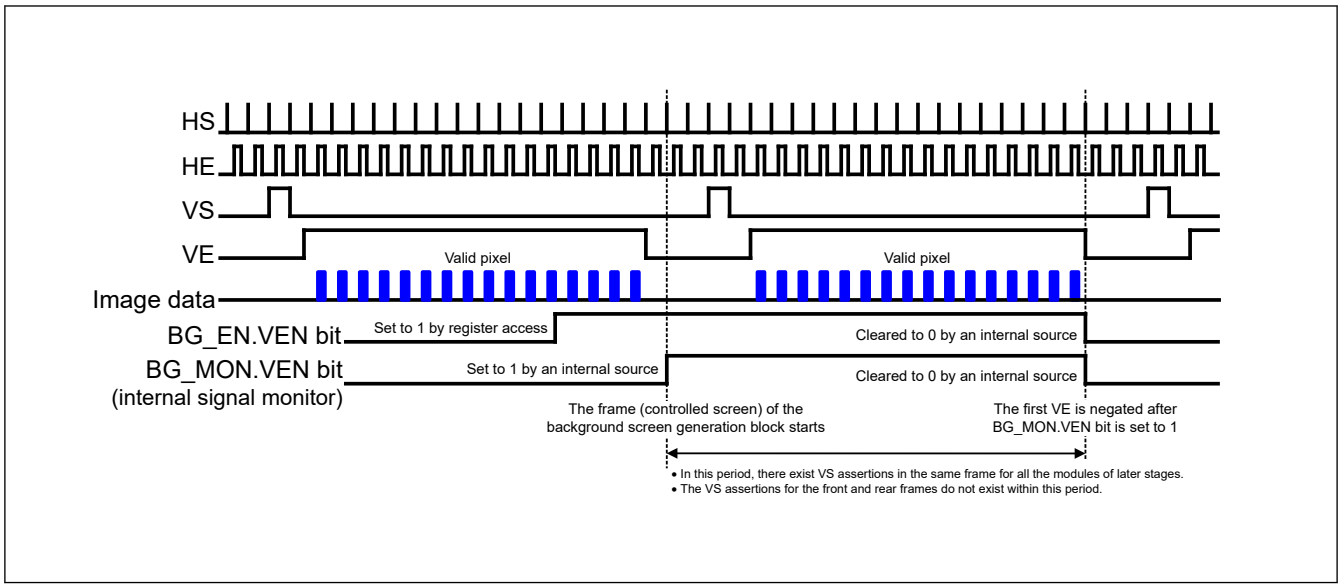


Figure 56.24 Control signals for register value reflection to internal operations

(3) Stopping and restarting in normal operation

To stop the GLCDC:

1. Confirm that the VEN bit of each module is cleared to 0.
2. Clear the BG_EN.EN bit (background plane operation enable) of the background screen generation module to 0.
3. Confirm that the BG_MON.EN bit (background plane operation monitor) of the background screen generation module has changed to 0. This bit is cleared to 0 only after operation has stopped. It is cleared at the frame end of the background screen generation module, not when the operations of all the modules are complete (not at the frame end of the output control module). If it is necessary to wait for all the modules to complete operations, a certain period (for example, a period equivalent to one line) is required.
4. Usually, a software reset can be safely applied (clearing of the BG_EN.SWRST bit) after confirming that the BG_MON.EN bit of the background screen generation module has changed to 0. (Even if the output signal returns to the initial value, no problems occur because the GLCDC has already entered the vertical blanking interval.)
5. When restarting the GLCDC by setting the relevant registers without applying a software reset, wait until sufficient time elapses after the BG_MON.EN bit becomes 0 (when the output control module output is the frame end) before starting the GLCDC. The GLCDC itself is not affected by this because the GLCDC starts operating after recognizing the assertion of the VS of the previous stage as the frame head. However, operation of some connected devices might be affected if a blanking interval or a period equivalent to one line is too short. For details, check the specifications of the connected device. When a software reset is applied, the register values are also initialized and almost all of the registers

must be set again. Only the color palette data (CLUT memory content) is retained (only for a software reset after normal end).

Figure 56.25 shows the changes in signal lines and bits for the stop and restart in normal operation. Even if the background screen generation module is stopped by a clearing of the BG_EN.EN bit to 0, the GLCDC stops because of an abnormal operation sequence, not normal operation, if the GLCDC0 and GLCDC1 bus access is not complete at the frame end because of an underflow in graphics 1 or 2, inappropriate setting of graphics data access, or other undesirable conditions.

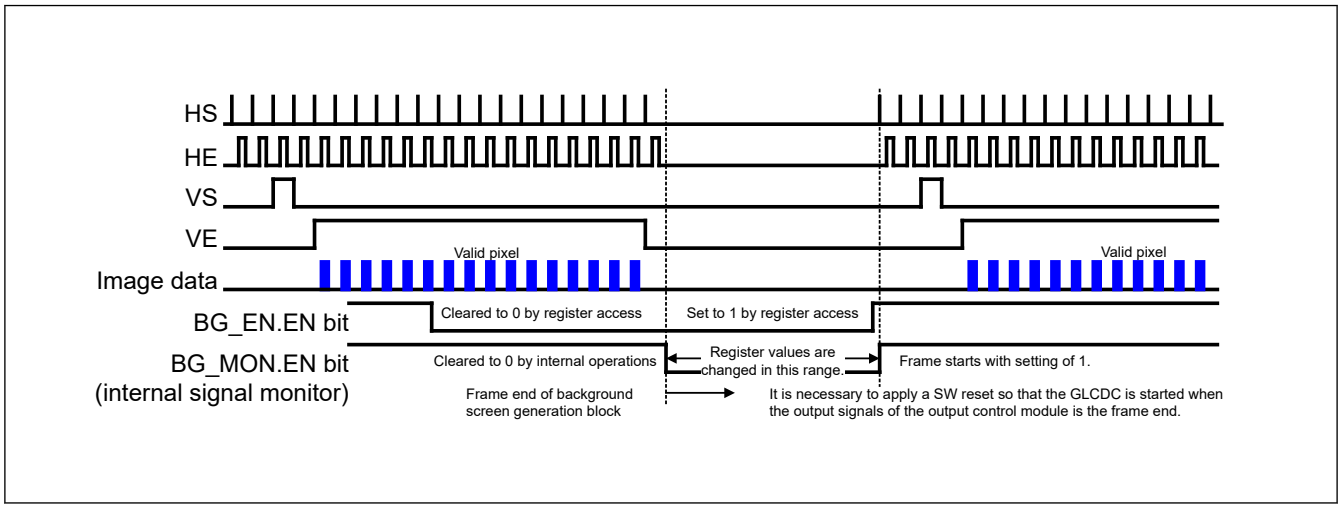


Figure 56.25 Stop and restart in normal operation

(4) Stopping and restarting in abnormal operation

Stop in abnormal operation occurs in the following cases:

- When during operation (BG_EN.EN or BG_MON.EN is set to 1), a software reset (BG_EN.SWRST bit is cleared to 0) or a reset is applied
- When unnecessary (unintended) access to the GLCDC0 and GLCDC1 bus occurs and a data cycle is not completed, even though the BG_EN.EN bit is cleared to 0 and so the BG_MON.EN bit is also cleared to 0

In both cases, the GLCDC is internally initialized and the register access bus can be accessed normally (except for a hardware reset). However, the GLCDC0 and GLCDC1 bus might write unintended graphics data to the GLCDC. Therefore, it is advisable to apply a software reset (clear the BG_EN.SWRST bit to 0) and confirm that there is no unintended access to the GLCDC0 and GLCDC1 bus, even after confirming that both the BG_EN.EN and BG_MON.EN bits are cleared to 0. Next release the GLCDC from the software reset (set the BG_EN.SWRST bit to 1), set the relevant registers, and set the BG_EN.En and BG_EN.VEN bits in the background screen generation module to 1 to restart the GLCDC. For the procedure for checking the GLCDC0 and GLCDC1 bus state, see the *ARM® AHB Specification*. Use this procedure even after a reset when the GLCDC0 and GLCDC1 bus, a target device, or a target controller is not initialized. If these are initialized on a reset assertion for the GLCDC, a normal startup procedure can be used without checking the GLCDC0 and GLCDC1 bus state.

56.3.2 Screen Definition

The essential signals for GLCDC operations are generated by the background screen generation module, and the graphics 1 and 2 modules and output control module operate based on the sequentially transferred vertical and horizontal synchronization signals (VS and HS) and vertical and horizontal pixel enable signals (VE and HE). The reference point (frame start point) of the background screen cannot be determined by the output signals. The point shown in the figures is virtual and provided only for register settings. The reference points (frame start points) of the graphics 1 and 2 modules and output control module are the assertion of the vertical synchronization signal (VS), which is input (output) from the previous stage. Each module defines the valid pixel display area and special processing area for pixel data (rectangular areas for graphics 1 and 2) according to the position and width based on this reference point. Figure 56.26 shows the definition of the background screen, and Figure 56.27 shows the definition of the graphics 1 and 2 screen. The output control module performs correction for the entire valid pixel area (when both VE and HE are 1) output from the previous stage (graphics 2). (No registers are provided in the output control module for setting the correction area.) TCON in the output control module

generates the control signals to be output based on the internal vertical and horizontal synchronization signals (VS and HS), and the change timing of the output signals can be freely modified within the valid setting range specifiable in the registers.

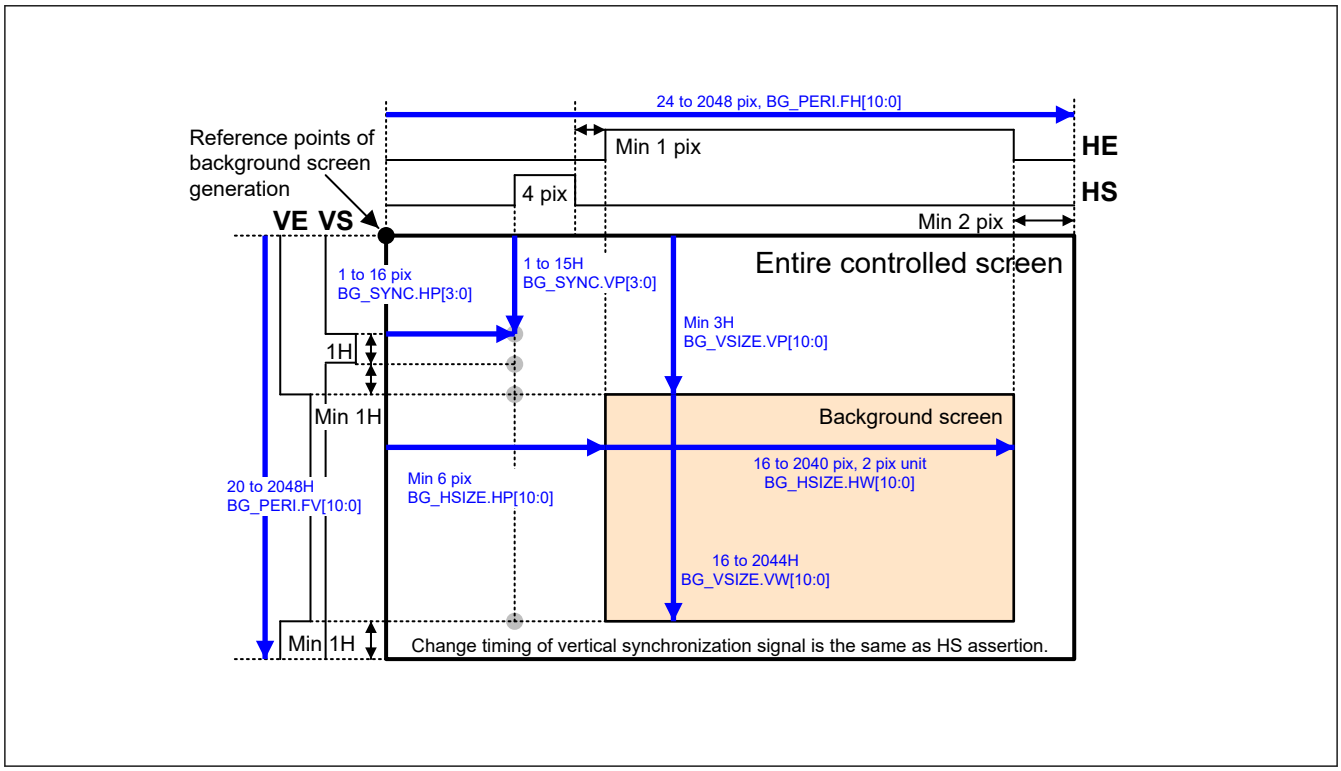


Figure 56.26 Definition of background screen

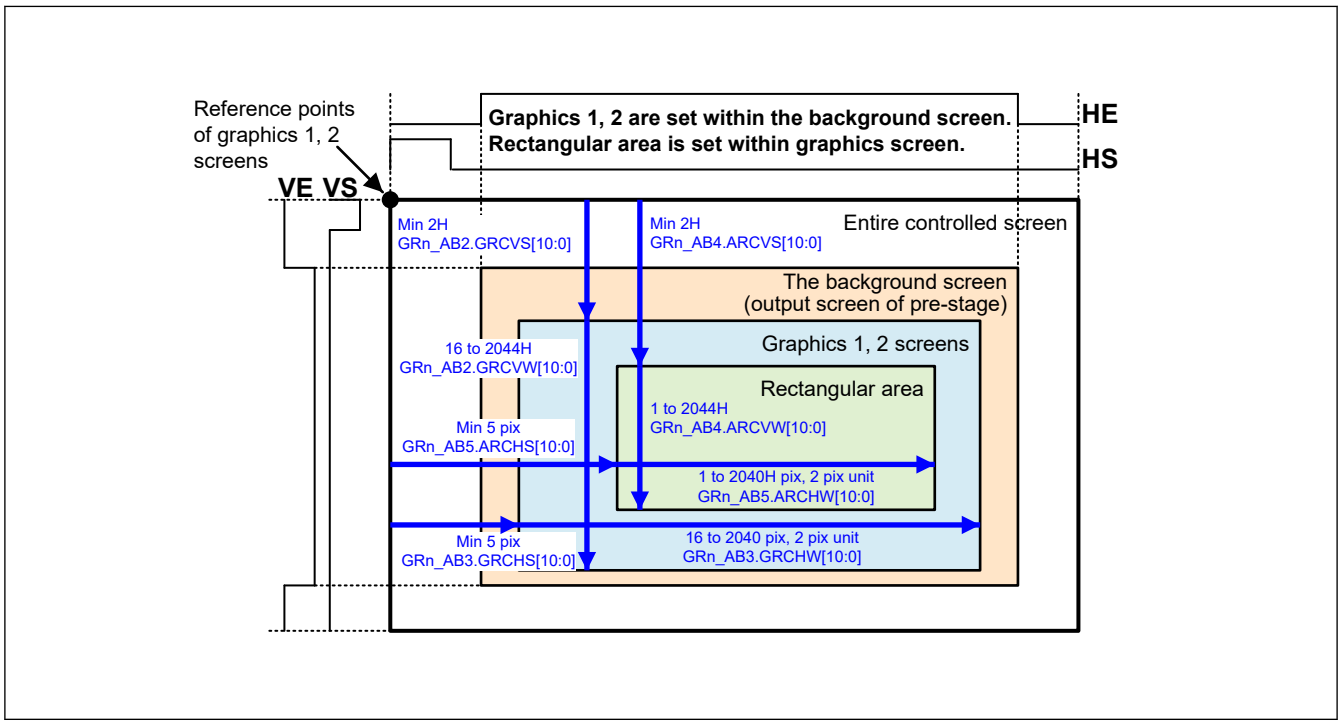


Figure 56.27 Definition of graphics 1 and 2 screens

56.3.3 Underflow and Interrupts

The GLCDC can detect the three types of status conditions described in this section. To detect each status condition, set the associated bit in the Status Detection Control Register (SYSCNT_DTCTEN) to 1.

56.3.3.1 Graphics 2 underflow detection

This function detects underflows in the graphics data interface block for graphics 2. The SYSCNT_STMON.L2UNDF flag sets to 1 if the graphics data cannot be read from the graphics data interface block (if valid data is not stored in the 4-stage ring buffer). The underflow is cleared as an internal state of graphics 2 on the VS (vertical synchronization) signal assertion of the previous stage. However, to clear the relevant bit in the Status Monitor Register of the system control block, the software must set the associated bit in the Status Clear Register to 1.

56.3.3.2 Graphics 1 underflow detection

This function detects underflows in the graphics data interface block for graphics 1. The SYSCNT_STMON.L1UNDF flag is set to 1 if the graphics data cannot be read from the graphics data interface block (if valid data is not stored in the 4-stage ring buffer). The underflow is cleared as an internal state of graphics 1 on the VS (vertical synchronization) signal assertion of the previous stage. However, to clear the relevant bit in the Status Monitor Register of the system control block, the software must set the associated bit in the Status Clear Register to 1.

56.3.3.3 Graphics 2 line detection

This function detects that the number of lines specified in the GR2_CLUTINT.LINE[10:0] bit was processed. The detection is performed on the HS (horizontal synchronization) signal assertion of the previous stage, not when valid pixels start to be processed. Each time the number of detected lines reach the value specified for graphics 2, the SYSCNT_STMON.VPOS flag sets to 1. As with underflow detection, to clear the relevant bit in the Status Monitor Register of the system control block, the software must set the associated bit in the Status Clear Register to 1.

56.3.3.4 Interrupts

The GLCDC provides three interrupt request output signals (GLCDC_L2UNDF, GLCDC_L1UNDF, and GLCDC_VPOS) that correspond to detection of the three status conditions. The GLCDC_L2UNDF, GLCDC_L1UNDF, and GLCDC_VPOS signals are associated with graphics 2 underflow detection, graphics 1 underflow detection, and graphics 2 line detection, respectively. Each of the interrupt request signals is asserted by setting the associated bit in the Interrupt Request Enable Register (SYSCNT_INTEN) to 1. Detecting the status and enabling the associated interrupt request can be controlled separately. Even if detection is enabled, the interrupt request signal is not asserted unless the interrupt request output signal is enabled. In addition, if the Status Monitor Register (SYSCNT_STMON) clears while the interrupt request output signal is asserted (the associated bit in the Status Clear Register (SYSCNT_STCLR) is set to 1), or if the associated bit in the Interrupt Request Enable Register (SYSCNT_INTEN) is cleared to 0, the interrupt request signal is negated. The interrupt request signal generation circuit is configured as shown in Figure 56.28. This circuit is glitch-free except for reset-induced glitches.

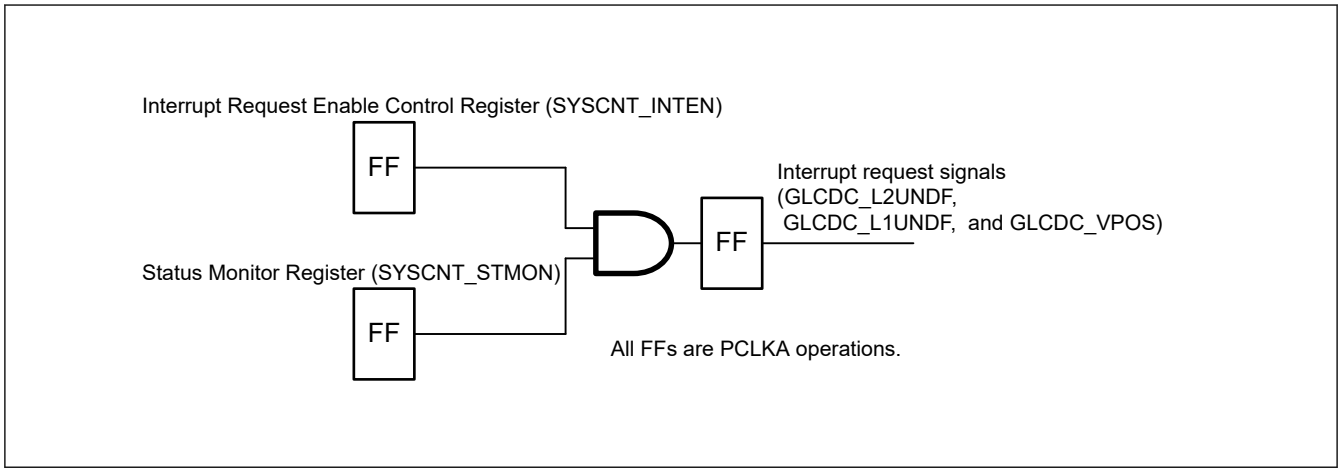


Figure 56.28 Interrupt request signal generation circuit

56.4 Usage notes

56.4.1 Module-stop function

GLCDC operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The GLCDC module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

57. MIPI PHY

57.1 Overview

This MIPI PHY is a MIPI D-PHY module.The D-PHY module supports MIPI Alliance Specification Version 2.1 for D-PHY Specification.

Table 57.1 MIPI DSI specifications

Parameter		Specifications
D-PHY	Number of lanes	Up to 2 Lanes
	Maximum rate	720 Mbps / Lane
MIPI PLL circuit	Input clock source	MOSC
	Input pulse frequency division ratio	Selectable from 1/2/3/4
	Input clock frequency	8 MHz to 48 MHz
	Input clock frequency (After DPHYPLFCR.IDIV[1:0] bits	5 MHz to 24 MHz *1
	Frequency multiplication ratio	Selectable from 20 to 180 (after the decimal point : 0/0.33/0.50/0.66)
	VCO frequency	160 MHz to 1440 MHz
	PLL Output clock	160 MHz to 1440 MHz

Note:
Note 1. . DPHYPLFCR.IDIV[1:0] must be set so that the frequency of the clock divided by DPHYPLFCR.IDIV[1:0] is within the range that is between 8 to 24 MHz. However, under the condition of 160 MHz ≤ PLL output clock (fDPHYPLL) < 320 MHz, the allowable frequency range of the clock divided by IDIV[1:0] is extended to 5 to 24 MHz.

The block diagram and I/O pins see Figure 54.1 and Table 54.1

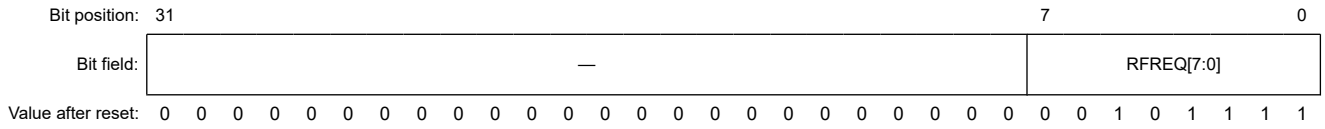
57.2 Register Description

Access the following registers only in units of 32 bits.

57.2.1 DPHYREFCR : D-PHY Reference Clock Setting Register

Base address: MIPI_PHY0 = 0x4034_6C00
MIPI_PHY0_NS = 0x5034_6C00

Offset address: 0x000



Bit	Symbol	Function	R/W
7:0	RFREQ[7:0]	Reference Clock Frequency Setting Set the frequency of the peripheral module clock (PCLKA).*1 PCLKA is used for the reference clock of the counter circuit which controls the internal timing of the D-PHY module. 0x27: 40 MHz*2 0x28: 41 MHz 0x29: 42 MHz ⋮ 0x75: 118 MHz 0x76: 119 MHz 0x77: 120 MHz Others: Settings prohibited.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Remove digits after decimal point.
e.g.) PCLKA: 100.5 MHz → RFREQ[7:0] = 01100011b (100 MHz)
Note 2. Frequency of PCLKA must be 40 MHz or more to use MIPI DSI function. Therefore, when using the MIPI DSI function, the operating power control mode of the system must also be set to high-speed-mode (OPCCR.OPCM[1:0] = 00b).

RFREQ[7:0] bits (Reference Clock Frequency Setting)

Set the RFREQ[7:0] bits to the frequency of the peripheral module clock (PCLKA).*1

PCLKA is used for the reference clock of the counter circuit which controls the internal timing of the D-PHY module.

57.2.2 DPHYPLFCR : D-PHY PLL Frequency Control Register

Base address: MIPI_PHY0 = 0x4034_6C00
MIPI_PHY0_NS = 0x5034_6C00

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	NMUL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	NFMUL[1:0]		—	—	—	—	—	—	IDIV[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	IDIV[1:0]	D-PHY PLL Input Frequency Division Ratio Select 0 0: 1 0 1: 1/2 1 0: 1/3 1 1: 1/4	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	NFMUL[1:0]	D-PHY PLL Frequency Multiplication Factor Select (Fractional Part) 0 0: 0.00 0 1: 0.33 1 0: 0.66 1 1: 0.50	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
23:16	NMUL[7:0]	D-PHY PLL Frequency Multiplication Factor Select (Integer Part) 0x13: 20 0x14: 21 0x15: 22 ⋮ 0x62: 99 0x63: 100 0x64: 101 ⋮ 0xB1: 178 0xB2: 179 0xB3: 180 Others: Setting prohibited	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

IDIV[1:0] bits (D-PHY PLL Input Frequency Division Ratio Select)

The IDIV[1:0] bits control the input division ratio I for D-PHY PLL.*1*2*3

NFMUL[1:0] bits (D-PHY PLL Frequency Multiplication Factor Select (Fractional Part))

The NFMUL[1:0] bits control the fractional part NF of the multiplication factor for D-PHY PLL.*2*3

NMUL[7:0] bits (D-PHY PLL Frequency Multiplication Factor Select (Integer Part))

The NMUL[7:0] bits control the integer part N of the multiplication factor for D-PHY PLL. *2*3

The frequency of the PLL output clock ($f_{DPHYPLL}$) can be calculated by the following fomula.

$f_{DPHYPLL}$ [MHz] = f_{MAIN} [MHz] × I × (NF + N)

Where, f_{MAIN} is the frequency of Main clock oscillator.

The D-PHY High-Speed data transmission rate is determined by the following formula:

Line rate [Mbps] = $f_{DPHYPLL}$ [MHz] / 2

e.g.)

In the case of f_{MAIN} = 24.576 MHz, IDIV[1:0] = 01b, NFMUL[1:0] = 11b, NMUL[7:0] = 0x63

$f_{DPHYPLL}$ = 24.576 [MHz] × 1/2 × (0.50 + 100) = 1235 [MHz]

Line rate [Mbps] = $f_{DPHYPLL}$ / 2 = 1235 MHz / 2 = 617.47 Mbps

Note 1. IDIV[1:0] must be set so that the frequency of the clock divided by IDIV[1:0] is within the range that is between 8 to 24 MHz. However, under the condition of 160 MHz ≤ PLL output clock ($f_{DPHYPLL}$) < 320 MHz, the allowable frequency range of the clock divided by IDIV[1:0] is extended to 5 to 24 MHz.

Note 2. DPHYPLFCR must be set so that the frequency of the PLL output clock ($f_{DPHYPLL}$) is within the range that is between 160 MHz to 1.44 GHz.

Note 3. DPHYPLFCR must be set while D-PHY PLL operation is stopped (DPHYPLOCR. PLLSTP = 1).

57.2.3 DPHYPLOCR : D-PHY PLL Operation Control Register

Base address: MIPI_PHY0 = 0x4034_6C00
MIPI_PHY0_NS = 0x5034_6C00

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLSTP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	PLLSTP	D-PHY PLL Operation Control 0: Operate the PLL 1: Stop the PLL	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

PLLSTP bit (D-PHY PLL Operation Control)

The PLLSTP bit controls the operation permission of the D-PHY PLL.

57.2.4 DPHYESCCR : D-PHY Escape Mode Clock Control Register

Base address: MIPI_PHY0 = 0x4034_6C00
MIPI_PHY0_NS = 0x5034_6C00

Offset address: 0x00C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ESCDIV[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	ESCDIV[4:0]	Escape Mode Transfer Clock Division Ratio 0x00: ×1/1 0x01: ×1/2 0x02: ×1/3 0x03: ×1/4 ⋮ 0x1D: ×1/30 0x1E: ×1/31 0x1F: ×1/32	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

ESCDIV[4:0] bits (Escape Mode Transfer Clock Division Ratio)

The ESCDIV[4:0] bits control the division ratio for Low-Power state clock which is use for Escape Mode Transfer.

The frequency of Low-Power state clock can be calculated by the following formula:

$f_{LPCLK} = f_{DPHYPLL} \times (1/16) \times (\text{division ratio set by ESCDIV[4:0]})^{*1*2*3}$

Note 1. These bits should be set so that the ratio of $T_{LPX\ (MASTER)}$/ $T_{LPX\ (SLAVE)}$ between this device and slave side is in the range of 2/3 to 3/2.

$T_{LPX\ (MASTER)} = 1 / f_{LPCLK}$

$T_{LPX\ (SLAVE)}$ is Transmitted length of any Low-Power state period on peripheral.

Note 2. These bits should be set so that the frequency of escape mode transfer clock (f_{LPX}) in this device is in the range of 2 to 17 MHz.

Note 3. These bits must be set while D-PHY PLL operation is stopped (DPHYPLOCR. PLLSTP = 1).

57.2.5 DPHYPWRCR : D-PHY Power Supplying Control Register

Base address: MIPI_PHY0 = 0x4034_6C00
MIPI_PHY0_NS = 0x5034_6C00

Offset address: 0x010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PWRSEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PWRSEN	D-PHY Power Supplying Control 0: Disable D-PHY LDO operation (stop supplying VDD_DPHY) 1: Enable D-PHY LDO operation (supply VDD_DPHY)	R/W

Bit	Symbol	Function	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

PWRSEN bit (D-PHY Power Supplying Control)

The PWRSEN bit controls the operation permission of the D-PHY LDO.

57.2.6 DPHYSFR : D-PHY Status Flag Register

Base address: MIPI_PHY0 = 0x4034_6C00
MIPI_PHY0_NS = 0x5034_6C00

Offset address: 0x01C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	PLLSF	—	—	—	—	—	—	—	PWRSF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PWRSF	D-PHY LDO Power-on Status Flag 0: D-PHY LDO is stopped or starting up 1: D-PHY LDO startup is completed (VDD_DPHY is stable)	R
7:1	—	These bits are read as 0.	R
8	PLLSF	D-PHY PLL Oscillation Stabilization Flag 0: D-PHY PLL clock is stopped or is not yet stable 1: D-PHY PLL clock is stable	R
31:9	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

PWRSF flag (D-PHY LDO Power-on Status Flag)

The PWRSF flag indicates whether the operation of the D-PHY LDO has stabilized or not.

PLLSF flag (D-PHY PLL Oscillation Stabilization Flag)

The PLLSF flag indicates whether the oscillation of the D-PHY PLL has stabilized or not.

57.2.7 DPHYOCR : D-PHY Operation Control Register

Base address: MIPI_PHY0 = 0x4034_6C00
MIPI_PHY0_NS = 0x5034_6C00

Offset address: 0x020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DPHYEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DPHYEN	D-PHY Operation Control 0: Disable D-PHY operation 1: Enable D-PHY operation	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

DPHYEN bit (D-PHY Operation Control)

The DPHYEN bit controls the operation permission of the D-PHY.

57.2.8 DPHYTIM1 : D-PHY Timing Control Register 1

Base address: MIPI_PHY0 = 0x4034_6C00
MIPI_PHY0_NS = 0x5034_6C00

Offset address: 0x024

Bit position:	31													18													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	TINIT[18:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	0	0	0	0	1

Bit	Symbol	Function	R/W
18:0	TINIT[18:0]	D-PHY T_INIT Parameter Setting Set the D-PHY TINIT parameter.	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

TINIT[18:0] bits (D-PHY T_INIT Parameter Setting)

The TINIT[18:0] bits specify the T_{INIT} parameter, which is the minimum duration of the TINIT state.

The time is calculated by the following formula:

T_{INIT} = (TINIT[18:0] + 1) × (period of the peripheral module clock (PCLKA))

When the frequency of PCLK is 120 MHz, the default value is below.

T_{INIT} = (72001 + 1) × 8.333 [ns] = 600 [µs]

57.2.9 DPHYTIM2 : D-PHY Timing Control Register 2

Base address: MIPI_PHY0 = 0x4034_6C00
MIPI_PHY0_NS = 0x5034_6C00

Offset address: 0x028

Bit position:	31																							7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TCLKPREP[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
7:0	TCLKPREP[7:0]	D-PHY T_CLK_PREPARE Parameter Setting Set the D-PHY $T_{CLK\text{-}PREPARE}$ parameter shown in Figure 57.1.	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
12	—	This bit is read as 1. The write value should be 1.	R/W
16:13	—	These bits are read as 0. The write value should be 0.	R/W
17	—	This bit is read as 1. The write value should be 1.	R/W

Bit	Symbol	Function	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

TCLKPREP[7:0] bits (D-PHY T_CLK_PREPARE Parameter Setting)

The TCLKPREP[7:0] bits specify the $T_{CLK\text{-}PREPARE}$ parameter, which is the duration of the LP-00 state (immediately before entry to the HS-0 state) in the clock lane. See Figure 57.1.

The time is calculated by the following formula:

$T_{CLK\text{-}PREPARE}$ = (TCLKPREP[7:0] + 1) × (period of the peripheral module clock (PCLKA))

When the frequency of PCLK is 120 MHz, the default value is below.

$T_{CLK\text{-}PREPARE}$ = (8 + 1) × 8.333 [ns] = 75 [ns]

57.2.10 DPHYTIM3 : D-PHY Timing Control Register 3

Base address: MIPI_PHY0 = 0x4034_6C00
MIPI_PHY0_NS = 0x5034_6C00

Offset address: 0x02C

Bit position:	31																						7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	THSPREP[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
7:0	THSPREP[7:0]	D-PHY T_THS_PREPARE Parameter Setting Set the D-PHY $T_{HS\text{-}PREPARE}$ parameter shown in Figure 57.1.	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
12	—	This bit is read as 1. The write value should be 1.	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

THSPREP[7:0] bits (D-PHY T_THS_PREPARE Parameter Setting)

The THSPREP[7:0] bits specify the $T_{HS\text{-}PREPARE}$ parameter, which is the duration of the LP-00 state (immediately before entry to the HS-0 state) in the data lane. See Figure 57.1.

The time is calculated by the following formula:

$T_{HS\text{-}PREPARE}$ = (THSPREP[7:0] + 1) × (period of the peripheral module clock (PCLKA))

When the frequency of PCLK is 120 MHz, the default value is below.

$T_{HS\text{-}PREPARE}$ = (8 + 1) × 8.333 [ns] = 75 [ns]

57.2.11 DPHYTIM4 : D-PHY Timing Control Register 4

Base address: MIPI_PHY0 = 0x4034_6C00
MIPI_PHY0_NS = 0x5034_6C00

Offset address: 0x030

Bit position:	31							24	23							16	15							8	7						0
Bit field:	TCLKTRL[7:0]								TCLKPOST[7:0]								TCLKPRE[7:0]								TCLKZERO[7:0]							
Value after reset:	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	TCLKZERO[7:0]	D-PHY T_CLK_ZERO Parameter Setting Set the D-PHY $T_{CLK-ZERO}$ parameter shown in Figure 57.1.	R/W
15:8	TCLKPRE[7:0]	D-PHY T_TCLK_PRE Parameter Setting Set the D-PHY $T_{CLK-PRE}$ parameter shown in Figure 57.1.	R/W
23:16	TCLKPOST[7:0]	D-PHY T_TCLK_POST Parameter Setting Set the D-PHY $T_{CLK-POST}$ parameter shown in Figure 57.1.	R/W
31:24	TCLKTRL[7:0]	D-PHY T_TCLK_TRAIL Parameter Setting Set the D-PHY $T_{CLK-TRAIL}$ parameter shown in Figure 57.1.	R/W

Note: S-TYPE-3, P-TYPE-3

TCLKZERO[7:0] bits (D-PHY T_CLK_ZERO Parameter Setting)

The TCLKZERO[7:0] bits specify the $T_{CLK-ZERO}$ parameter. For details, see Figure 57.1.

The time is calculated by the following formula:

$T_{CLK-ZERO}$ = (TCLKZERO[7:0] + 1) × (period of the peripheral module clock (PCLKA))

When the frequency of PCLK is 120 MHz, the default value is below.

$T_{CLK-ZERO}$ = (31 + 1) × 8.333 [ns] = 267 [ns]

TCLKPRE[7:0] bits (D-PHY T_TCLK_PRE Parameter Setting)

The TCLKPRE[7:0] bits specify the $T_{CLK-PRE}$ parameter. For details, see Figure 57.1.

The time is calculated by the following formula:

$T_{CLK-PRE}$ = (TCLKPRE[7:0] + 1) × (period of the peripheral module clock (PCLKA))

When the frequency of PCLK is 120 MHz, the default value is below.

$T_{CLK-PRE}$ = (3 + 1) × 8.333 [ns] = 33 [ns]

TCLKPOST[7:0] bits (D-PHY T_TCLK_POST Parameter Setting)

The TCLKPOST[7:0] bits specify the $T_{CLK-POST}$ parameter. For details, see Figure 57.1.

The time is calculated by the following formula:

$T_{CLK-POST}$ = (TCLKPOST[7:0] + 1) × (period of the peripheral module clock (PCLKA))

When the frequency of PCLK is 120 MHz, the default value is below.

$T_{CLK-POST}$ = (32 + 1) × 8.333 [ns] = 275 [ns]

TCLKTRL[7:0] bits (D-PHY T_TCLK_TRAIL Parameter Setting)

The TCLKTRL[7:0] bits specify the $T_{CLK-POST}$ parameter. For details, see Figure 57.1.

The time is calculated by the following formula:

$T_{CLK-TRAIL}$ = (TCLKTRL[7:0] + 1) × (period of the peripheral module clock (PCLKA))

When the frequency of PCLK is 120 MHz, the default value is below.

$T_{CLK-TRAIL}$ = (6 + 1) × 8.333 [ns] = 58 [ns]

57.2.12 DPHYTIM5 : D-PHY Timing Control Register 5

Base address: MIPI_PHY0 = 0x4034_6C00
MIPI_PHY0_NS = 0x5034_6C00

Offset address: 0x034

Bit position:	31								23								16	15							8	7							0
Bit field:	—	—	—	—	—	—	—	—	THSEXIT[7:0]									THSTRL[7:0]								THSZERO[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1

Bit	Symbol	Function	R/W
7:0	THSZERO[7:0]	D-PHY T_THS_ZERO Parameter Setting Set the D-PHY $T_{HS-ZERO}$ parameter shown in Figure 57.1.	R/W
15:8	THSTRL[7:0]	D-PHY T_THS_TRAIL Parameter Setting Set the D-PHY $T_{HS-TRAIL}$ parameter shown in Figure 57.1.	R/W
23:16	THSEXIT[7:0]	D-PHY T_THS_EXIT Parameter Setting Set the D-PHY $T_{HS-EXIT}$ parameter shown in Figure 57.1.	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

THSZERO[7:0] bits (D-PHY T_THS_ZERO Parameter Setting)

The THSZERO[7:0] bits specify the $T_{HS-ZERO}$ parameter. For details, see Figure 57.1.

The time is calculated by the following formula:

$T_{HS-ZERO}$ = (THSZERO[7:0] + 1) × (period of the peripheral module clock (PCLKA))

When the frequency of PCLK is 120 MHz, the default value is below.

$T_{HS-ZERO}$ = (15 + 1) × 8.333 [ns] = 133 [ns]

THSTRL[7:0] bits (D-PHY T_THS_TRAIL Parameter Setting)

The THSTRL[7:0] bits specify the $T_{HS-TRAIL}$ parameter. For details, see Figure 57.1.

The time is calculated by the following formula:

$T_{HS-TRAIL}$ = (THSTRL[7:0] + 1) × (period of the peripheral module clock (PCLKA))

When the frequency of PCLK is 120 MHz, the default value is below.

$T_{HS-TRAIL}$ = (8 + 1) × 8.333 [ns] = 75 [ns]

THSEXIT[7:0] bits (D-PHY T_THS_EXIT Parameter Setting)

The THSEXIT[7:0] bits specify the $T_{HS-EXIT}$ parameter. For details, see Figure 57.1.

The time is calculated by the following formula:

$T_{HS-EXIT}$ = (THSEXIT[7:0] + 1) × (period of the peripheral module clock (PCLKA))

When the frequency of PCLK is 120 MHz, the default value is below.

$T_{HS-EXIT}$ = (12 + 1) × 8.333 [ns] = 108 [ns]

57.2.13 DPHYTIM6 : D-PHY Timing Control Register 6

Base address: MIPI_PHY0 = 0x4034_6C00
MIPI_PHY0_NS = 0x5034_6C00

Offset address: 0x038

Bit position:	31																							7						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TLPX[7:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Bit	Symbol	Function	R/W
7:0	TLPX[7:0]	D-PHY T_TLPX Parameter Setting Set the D-PHY T_{LPX} parameter shown in Figure 57.1.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

TLPX[7:0] bits (D-PHY T_TLPX Parameter Setting)

The THSZERO[7:0] bits specify the T_{LPX} parameter. For details, see Figure 57.1.

The time is calculated by the following formula:

T_{LPX} = (TLPX[7:0] + 1) × (period of the peripheral module clock (PCLKA))

When the frequency of PCLK is 120 MHz, the default value is below.

T_{LPX} = (6 + 1) × 8.333 [ns] = 58 [ns]

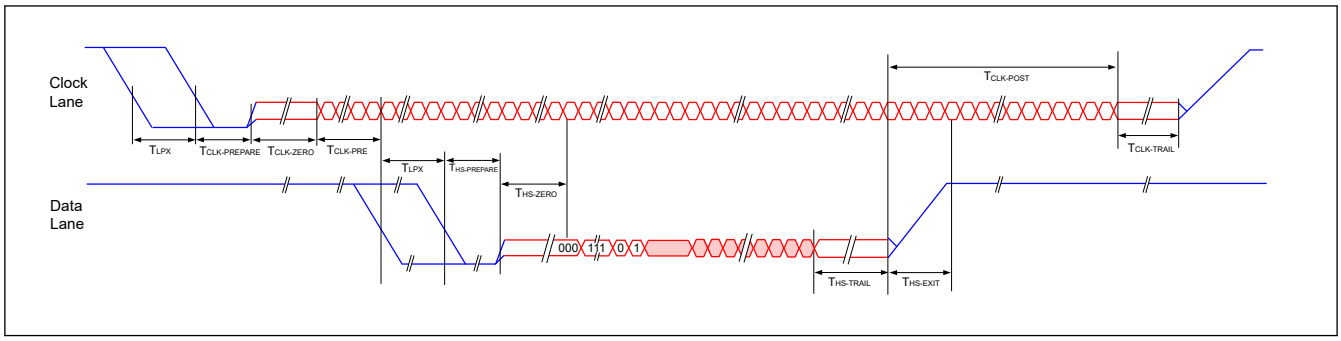


Figure 57.1 High-Speed data transmission in normal mode

Table 57.2 Recommended D-PHY timing setting for PCLKA = 120/100/80/75 [MHz] (1 of 6)

PCLKA 120MHz					
No.	Bit Name	Transmission Rate			
		80Mbps	125Mbps	250Mbps	Over 250Mbps
1	TINIT[18:0]	19'd72001 (0x11941)	19'd72001 (0x11941)	19'd72001 (0x11941)	19'd72001 (0x11941)
2	TCLK_PREPARE[7:0]	8'd8 (0x8)	8'd8 (0x8)	8'd8 (0x8)	8'd8 (0x8)
3	THS_PREPARE[7:0]	8'd13 (0xD)	8'd11 (0xB)	8'd11 (0xB)	8'd9 (0x9)
4	TCLK_ZERO[7:0]	8'd31 (0x1F)	8'd31 (0x1F)	8'd31 (0x1F)	8'd31 (0x1F)
5	TCLK_PRE[7:0]	8'd22 (0x16)	8'd13 (0xD)	8'd12 (0xC)	8'd3 (0x3)
6	TCLK_POST[7:0]	8'd85 (0x55)	8'd85 (0x55)	8'd85 (0x55)	8'd32 (0x20)
7	TCLK_TRAIL[7:0]	8'd10 (0xA)	8'd10 (0xA)	8'd10 (0xA)	8'd6 (0x6)
8	THS_ZERO[7:0]	8'd21 (0x15)	8'd21 (0x15)	8'd21 (0x15)	8'd15 (0xF)
9	THS_TRAIL[7:0]	8'd15 (0xF)	8'd15 (0xF)	8'd15 (0xF)	8'd8 (0x8)
10	THS_EXIT[7:0]	8'd12 (0xC)	8'd12 (0xC)	8'd12 (0xC)	8'd12 (0xC)

Table 57.2 Recommended D-PHY timing setting for PCLKA = 120/100/80/75 [MHz] (2 of 6)

PCLKA 120MHz					
No.	Bit Name	Transmission Rate			
		80Mbps	125Mbps	250Mbps	Over 250Mbps
11	TLPX[7:0]	8'd6 (0x6)	8'd6 (0x6)	8'd6 (0x6)	8'd6 (0x6)

Table 57.2 Recommended D-PHY timing setting for PCLKA = 120/100/80/75 [MHz] (3 of 6)

PCLKA 100MHz					
No.	Bit Name	Transmission Rate			
		80Mbps	125Mbps	250Mbps	Over 250Mbps
1	TINIT[18:0]	19'd60001 (0xEA61)	19'd60001 (0xEA61)	19'd60001 (0xEA61)	19'd60001 (0xEA61)
2	TCLK_PREPARE[7:0]	8'd8 (0x8)	8'd8 (0x8)	8'd8 (0x8)	8'd8 (0x8)
3	THS_PREPARE[7:0]	8'd12 (0xC)	8'd12 (0xC)	8'd8 (0x8)	8'd7 (0x7)
4	TCLK_ZERO[7:0]	8'd28 (0x1C)	8'd28 (0x1C)	8'd28 (0x1C)	8'd28 (0x1C)
5	TCLK_PRE[7:0]	8'd22 (0x16)	8'd13 (0xD)	8'd12 (0xC)	8'd3 (0x3)
6	TCLK_POST[7:0]	8'd76 (0x4C)	8'd62 (0x3E)	8'd32 (0x20)	8'd32 (0x20)
7	TCLK_TRAIL[7:0]	8'd10 (0xA)	8'd10 (0xA)	8'd6 (0x6)	8'd6 (0x6)
8	THS_ZERO[7:0]	8'd21 (0x15)	8'd21 (0x15)	8'd15 (0xF)	8'd15 (0xF)
9	THS_TRAIL[7:0]	8'd15 (0xF)	8'd10 (0xA)	8'd6 (0x8)	8'd6 (0x8)
10	THS_EXIT[7:0]	8'd12 (0xC)	8'd12 (0xC)	8'd12 (0xC)	8'd12 (0xC)
11	TLPX[7:0]	8'd6 (0x6)	8'd6 (0x6)	8'd6 (0x6)	8'd6 (0x6)

Table 57.2 Recommended D-PHY timing setting for PCLKA = 120/100/80/75 [MHz] (4 of 6)

PCLKA 80MHz					
No.	Bit Name	Transmission Rate			
		80Mbps	125Mbps	250Mbps	Over 250Mbps
1	TINIT[18:0]	19'd48001 (0xBB81)	19'd48001 (0xBB81)	19'd48001 (0xBB81)	19'd48001 (0xBB81)
2	TCLK_PREPARE[7:0]	8'd6 (0x6)	8'd6 (0x6)	8'd6 (0x6)	8'd6 (0x6)
3	THS_PREPARE[7:0]	8'd10 (0xA)	8'd8 (0x8)	8'd8 (0x8)	8'd7 (0x7)
4	TCLK_ZERO[7:0]	8'd20 (0x14)	8'd20 (0x14)	8'd20 (0x14)	8'd20 (0x14)
5	TCLK_PRE[7:0]	8'd22 (0x16)	8'd13 (0xD)	8'd12 (0xC)	8'd3 (0x3)
6	TCLK_POST[7:0]	8'd58 (0x3A)	8'd46 (0x2E)	8'd32 (0x20)	8'd32 (0x20)
7	TCLK_TRAIL[7:0]	8'd6 (0x6)	8'd4 (0x4)	8'd3 (0x3)	8'd3 (0x3)
8	THS_ZERO[7:0]	8'd16 (0x10)	8'd13 (0xD)	8'd10 (0xA)	8'd10 (0xA)
9	THS_TRAIL[7:0]	8'd9 (0x9)	8'd7 (0x7)	8'd4 (0x4)	8'd4 (0x4)
10	THS_EXIT[7:0]	8'd10 (0xA)	8'd10 (0xA)	8'd10 (0xA)	8'd10 (0xA)
11	TLPX[7:0]	8'd4 (0x4)	8'd4 (0x4)	8'd4 (0x4)	8'd4 (0x4)

Table 57.2 Recommended D-PHY timing setting for PCLKA = 120/100/80/75 [MHz] (5 of 6)

PCLKA 75MHz					
No.	Bit Name	Transmission Rate			
		80Mbps	125Mbps	250Mbps	Over 250Mbps
1	TINIT[18:0]	19'd45001 (0xAFC9)	19'd45001 (0xAFC9)	19'd45001 (0xAFC9)	19'd45001 (0xAFC9)
2	TCLK_PREPARE[7:0]	8'd6 (0x6)	8'd6 (0x6)	8'd6 (0x6)	8'd6 (0x6)
3	THS_PREPARE[7:0]	8'd10 (0xA)	8'd8 (0x8)	8'd8 (0x8)	8'd7 (0x7)

Table 57.2 Recommended D-PHY timing setting for PCLKA = 120/100/80/75 [MHz] (6 of 6)

PCLKA 75MHz					
No.	Bit Name	Transmission Rate			
		80Mbps	125Mbps	250Mbps	Over 250Mbps
4	TCLK_ZERO[7:0]	8'd20 (0x14)	8'd20 (0x14)	8'd20 (0x14)	8'd20 (0x14)
5	TCLK_PRE[7:0]	8'd22 (0x16)	8'd13 (0xD)	8'd12 (0xC)	8'd3 (0x3)
6	TCLK_POST[7:0]	8'd58 (0x3A)	8'd46 (0x2E)	8'd32 (0x20)	8'd32 (0x20)
7	TCLK_TRAIL[7:0]	8'd6 (0x6)	8'd4 (0x4)	8'd3 (0x3)	8'd3 (0x3)
8	THS_ZERO[7:0]	8'd16 (0x10)	8'd13 (0xD)	8'd10 (0xA)	8'd10 (0xA)
9	THS_TRAIL[7:0]	8'd9 (0x9)	8'd7 (0x7)	8'd3 (0x3)	8'd3 (0x3)
10	THS_EXIT[7:0]	8'd10 (0xA)	8'd10 (0xA)	8'd10 (0xA)	8'd10 (0xA)
11	TLPX[7:0]	8'd4 (0x4)	8'd4 (0x4)	8'd4 (0x4)	8'd4 (0x4)

57.3 Operation

57.3.1 D-PHY Start-up Procedure

To use the MIPI DSI function, D-PHY must first be activated. The D-PHY has a dedicated regulator (D-PHY LDO) and PLL (D-PHY PLL), and these are activated when the D-PHY startup.

The startup procedure for the D-PHY is as follows:

Before executing the following flow, the basic register settings of GLCDC must be completed. Especially, SYSCNT_PANEL_CLK.CLKEN must be set to 1 to supply the panel clock to the DSI Host as the video clock.

1. Confirm that Graphics power domain is supplied power and PCLKA is supplied to the MIPI subsystem.
2. Set the DPHYREFCR.RFREQ[7:0] bits for the frequency of PCLKA.
3. Set the DPHYPWRCR.PWRSEN bit to 1 to start operating the D-PHY LDO for supplying VDD_DPHY.
4. Wait for the start-up of VDD_DPHY while the DPHYSFR.PWRSF bit is 0.
5. Set the DPHYPLFCR register to determine the frequency of D-PHY PLL clock.
6. Set the DPHYESCCR.ESCDIV[4:0] bits to determine the frequency of the D-PHY escape mode clock.
7. Clear the DPHYPLOCR.PLLSTP bit to 0 to enable PLL operation.
8. Wait for the oscillation stabilization while the DPHYSFR.PLLSF bit is 0.
9. Set the following registers for the D-PHY timing parameters.
   (DPHYTIM1, DPHYTIM2, DPHYTIM3, DPHYTIM4, DPHYTIM5, DPHYTIM6)
10. Set the DPHYOCR.DPHYEN bit to 1 to enable D-PHY operation.

57.3.2 D-PHY Stop Procedure

To stop the D-PHY, follow the steps below.

And the D-PHY LDO is impossible to operate in the standby mode. Therefore, before going to standby mode, it is necessary to disable the D-PHY LDO operation according to follows.

1. Clear the DPHYOCR.DPHYEN bit to 0 to disable the D-PHY.
2. Set the DPHYPLOCR.PLLSTP bit to 1 to stop the PLL operation.
3. Clear the DPHYPWRCR.PWRSEN bit to 0 to disable the D-PHY LDO (Stop VDD_DPHY supply).

58. MIPI DSI

58.1 Overview

This MIPI DSI is a MIPI DSI-2 Host module.The DSI-2 Host module has a transmitter function for MIPI Alliance Specification for Display Serial Interface 2 (DSI-2). The DSI-2 Host supports MIPI Alliance Specification for Display Serial Interface 2 (DSI-2) Specification.

Table 58.1 MIPI DSI specifications

Parameter		Specifications
Compliant specification		• MIPI Alliance Specification for Display Serial Interface 2 Version 1.1 (with Errata 01) • MIPI Alliance Specification for D-PHY Version 2.1 (with Errata 01) (80 Mbps to 720 Mbps/Lane and up to 2 Lanes).
Video Mode Operation	Available input video format from GLCDC	• Parallel RGB888 (24 bits), little endian • Parallel RGB666 (18 bits), little endian • Parallel RGB565 (16 bits), little endian
	Available output format	• RGB (16 bits, 18 bits, 24 bits)
	Available video mode packet sequence	• Non-Burst Mode with Sync Pulse • Non-Burst Mode with Sync Event • Burst Mode
	Others	• Selectable Blanking Packet or LP-11 during each of blanking interval of HSA, HBP, and HFP
Command Mode Operation*1	Sequence Operation Channel-0	LP only packet generation and LP packet reception from descriptor list
	Sequence Operation Channel-1	HS or LP packet generation and LP packet reception from descriptor list
DSI Link support functions		• 1 and 2 Lane configurations • Unidirectional High-Speed mode transfer (HS-TX) • Bidirectional LP mode transfer/receipt (LP-TX / LP-RX) (Only Lane 0) • ECC/Checksum generation for Tx packet • ECC/Checksum verification and ECC error correction for Rx packet • Ultra-Low-Power mode (ULPS) • Automated power change to LP mode and return to HS mode • Automated clock stop and resume (non-continuous clock mode) • Assignment for Virtual Channel in video mode • Assignment for individual Virtual Channel for each packet in Command mode • Detection for PHY contention error and timeout error • Generation of scrambled packets • Input of TE signal
Module-stop function		Module-stop state can be set to reduce power consumption.
TrustZone Filter		Security and Privilege attribution can be set.

Note 1. Ch0 and Ch1 are exclusive.Cannot be used simultaneously.

The block diagram and I/O pins see Figure 54.1 and Table 54.1.

58.2 Register Description

Access the following registers only in units of 32 bits.

58.2.1 ISR : Interrupt Status Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	PPI	—	—	—	FERR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	RCV	—	—	—	VM	—	—	—	SQ1	—	—	—	SQ0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SQ0	Sequence Channel-0 Interrupt Flag 0: No interrupt detected 1: Sequence Operation Channel-0 interrupt detected	R
3:1	—	These bits are read as 0.	R
4	SQ1	Sequence Channel-1 Interrupt Flag 0: No interrupt detected 1: Sequence Operation Channel-1 interrupt detected	R
7:5	—	These bits are read as 0.	R
8	VM	Video Mode Interrupt Flag 0: No interrupt detected 1: Video Mode interrupt detected	R
11:9	—	These bits are read as 0.	R
12	RCV	Receive Interrupt Flag 0: No interrupt detected 1: Receive interrupt detected	R
15:13	—	These bits are read as 0.	R
16	FERR	Fatal Error Interrupt Flag 0: No interrupt detected 1: Fatal Error interrupt detected	R
19:17	—	These bits are read as 0.	R
20	PPI	PPI Interrupt Flag 0: No interrupt detected 1: PPI interrupt detected	R
31:21	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

SQ0 bit (Sequence Channel-0 Interrupt Flag)

The SQ0 flag indicates that the Sequence Operation Channel-0 interrupt is detected. The Sequence Operation Channel-0 interrupt is an interrupt shown in the SQCH0SR register.

The SQ0 flag is set to 1 when any one of the flags of the SQCH0SR register elements is set. For details, see section 58.2.54. SQCH0SR : Sequence Channel 0 Status Register.

SQ1 bit (Sequence Channel-1 Interrupt Flag)

The SQ1 flag indicates that the Sequence Operation Channel-1 interrupt is detected. The Sequence Operation Channel-1 interrupt is an interrupt shown in the SQCH1SR register.

The SQ1 flag is set to 1 when any one of the flags of the SQCH1SR register elements is set. For details, see section 58.2.58. SQCH1SR : Sequence Channel 1 Status Register.

VM bit (Video Mode Interrupt Flag)

The VM flag indicates that the Video Mode interrupt is detected. The Video Mode interrupt is an interrupt shown in the VMSR register.

The VM flag is set to 1 when any one of the flags of the VMSR register elements is set. For details, see section 58.2.45. VMSR : Video Mode Status Register.

RCV bit (Receive Interrupt Flag)

The RCV flag indicates that the Receive interrupt is detected. The Receive interrupt is an interrupt shown in the RXSR register.

The RCV flag is set to 1 when any one of the flags of the RXSR register elements is set. For details, see section 58.2.14. RXSR : Receive Status Register.

FERR bit (Fatal Error Interrupt Flag)

The FERR flag indicates that the Fatal Error interrupt is detected. The Fatal Error interrupt is an interrupt shown in the FERRSR register.

The FERR flag is set to 1 when any one of the flags of the FERRSR register elements is set. For details, see section 58.2.35. FERRSR : Fatal Error Status Register.

PPI bit (PPI Interrupt Flag)

The PPI flag indicates that the PPI interrupt is detected. The PPI interrupt is an interrupt shown in the PLSR register.

The PPI flag is set to 1 when any one of the flags of the PLSR register elements is set. For details, see section 58.2.40. PLSR : Physical Lane Status Register.

58.2.2 LINKSR : Link Status Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	LPBUSY	HSBUSY	—	—	—	VRUN	—	—	—	SQ1RUN	—	—	—	SQ0RUN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SQ0RUN	Sequence Channel-0 Running Flag 0: Sequence Operation Channel-0 stopped 1: Sequence Operation Channel-0 in operation	R
3:1	—	These bits are read as 0.	R
4	SQ1RUN	Sequence Channel-1 Running Flag 0: Sequence Operation Channel-1 stopped 1: Sequence Operation Channel-1 in operation	R
7:5	—	These bits are read as 0.	R
8	VRUN	Video Mode Operation Running Flag 0: Video mode stopped 1: Video mode in operation	R
11:9	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
12	HSBUSY	HS Operation Busy Flag 0: HS mode stopped 1: HS mode in operation	R
13	LPBUSY	LP Operation Busy Flag 0: LP mode stopped 1: LP mode in operation	R
15:14	—	These bits are read as 0.	R
20:16	—	The read values are undefined.	R
23:21	—	These bits are read as 0.	R
28:24	—	The read values are undefined.	R
31:29	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

SQ0RUN bit (Sequence Channel-0 Running Flag)

The SQ0RUN flag is a copy of the SQCH0SR.RUNNING flag. The meaning also is same. For details, see section 58.2.54. SQCH0SR : Sequence Channel 0 Status Register.

SQ1RUN bit (Sequence Channel-1 Running Flag)

The SQ1RUN flag is a copy of the SQCH1SR.RUNNING flag. The meaning also is the same. For details, see section 58.2.58. SQCH1SR : Sequence Channel 1 Status Register.

VRUN bit (Video Mode Operation Running Flag)

The VRUN flag is a copy of the VMSR.RUNNING bit. The meaning also is the same. For details, see section 58.2.45. VMSR : Video Mode Status Register.

HSBUSY bit (HS Operation Busy Flag)

The HSBUSY flag indicates that an operation related to HS mode is running.

LPBUSY bit (LP Operation Busy Flag)

The LPBUSY flag indicates that an operation related to LP mode is running.

58.2.3 TXSETR : Transmit Set Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x100

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	DLEN	CLEN	—	—	—	—	—	—	NUMLANE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
1:0	NUMLANE[1:0]	Number of Lane Set the number of lanes for use. 0 0: 1 Lane (Use of Lane-0) 0 1: 2 Lane (Use of Lane-0 and Lane-1) Others Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	CLEN	Clock Lane Enable 0: Disable 1: Enable	R/W
9	DLEN	Data Lane Enable 0: Disable 1: Enable	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	—	This bit is read as 1. The write value should be 1.	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register can only be set to change during the initialization at initial startup or Software Reset process (section 58.3.2. Software Reset).

NUMLANE[1:0] bits (Number of Lane)

NUMLANE[1:0] bits control the number of lanes to be used. Set the number of lanes for use.

CLEN bit (Clock Lane Enable)

The CLEN bit controls a Clock Lane of D-PHY. Set the CLEN bit to 1 to enable the Clock Lane.

DLEN bit (Data Lane Enable)

The DLEN bit controls Data Lanes of D-PHY. When the DLEN bit is set to 1, the Data Lanes specified by the NUMLANE[1:0] bits are enabled.

58.2.4 HSCLKSETR : HS Clock Set Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x104

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HSCLMD	HSCLST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	HSCLST	HS Clock Start 0: Stop HS transmission (keep LP state) 1: Start HS transmission	R/W
1	HSCLMD	HS Clock Running Mode 0: Non-continuous clock mode 1: Continuous clock mode	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

HSCLST bit (HS Clock Start)

The HSCLST bit controls the HS transmission from Clock Lane. When HSCLST = 1, the HS clock is transferred from the Clock Lane according to the setting of the HSCLMD bit.

It is prohibited to set the HSCLST bit to 1 when TXSETR.CLEN = 0.

HSCLMD bit (HS Clock Running Mode)

The HSCLMD bit controls the HS transmission mode from Clock Lane.

When HSCLMD = 0, the HS clock is transferred from the Clock Lane only when the HS transmit is requested (Non-continuous clock mode).

When HSCLMD = 1, the Clock Lane keeps the HS transmission (Continuous clock mode).

It is prohibited to change the setting of the HSCLMD bit when HSCLST = 1.

58.2.5 ULPSSETR : ULPS Set Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x108

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WKUP[7:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Bit	Symbol	Function	R/W
7:0	WKUP[7:0]	ULPS Wakeup Period Set the T_{WAKEUP} period of ULPS.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register can only be set to change during the initialization at initial startup or Software Reset process (section 58.3.2. Software Reset).

WKUP[7:0] bits (ULPS Wakeup Period)

The WKUP[7:0] bits control the T_{WAKEUP} period during which DSI Host drives a Mark-1 state prior to a Stop state in order to initiate an exit from ULPS.

This period is calculated by the following formula:

$T_{WAKEUP}[ms]^{*1} = WKUP[7:0] \times 128 \times (1 / f_{LPCLK}[MHz]^{*2}) / 1000$

Note 1. This period should be more than 1 ms.
Note 2. f_{LPCLK} is the frequency of the Low-Power mode clock (LPCLK).

58.2.6 ULPSCR : ULPS Control Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x10C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	DLEXIT	DLENT	—	—	CLEXIT	CLENT	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
23:0	—	These bits are read as 0. The write value should be 0.	W

Bit	Symbol	Function	R/W
24	CLENT	CL ULPS Enter 0: No operation 1: Transition Clock Lane to ULPS	W
25	CLEXIT	CL ULPS Exit 0: No operation 1: Clock Lane exits from ULPS	W
27:26	—	These bits are read as 0. The write value should be 0.	W
28	DLENT	DL ULPS Enter 0: No operation 1: Transition Data Lanes to ULPS	W
29	DLEXIT	DL ULPS Exit 0: No operation 1: Data Lanes exit from ULPS	W
31:30	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

CLENT bit (CL ULPS Enter)

The CLENT bit controls the transition of the Clock Lane to Ultra-low Power State (ULPS).

Set the CLENT bit to 1 during the period when the HSCLKSETR.HSCLST bit is 0.

It is prohibited to set the CLENT bit to 1 when Clock Lane is in ULPS.

CLEXIT bit (CL ULPS Exit)

The CLEXIT bit controls the exit of the Clock Lane from Ultra-low Power State (ULPS).

It is prohibited to set the CLEXIT bit to 1 when the Clock Lane is not in ULPS.

DLENT bit (DL ULPS Enter)

The DLENT bit controls the transition of the Data Lanes to Ultra-low Power State (ULPS).

It is prohibited to set the DLENT bit to 1 when the Data Lanes is in ULPS.

DLEXIT bit (DL ULPS Exit)

The DLEXIT bit controls the exit of the Data Lanes from Ultra-low Power State (ULPS).

It is prohibited to set the DLEXIT bit to 1 when the Data Lanes is not in ULPS.

58.2.7 RSTCR : Reset Control Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x110

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FTXSTP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SWRST	Software Reset 0: Complete the Software Reset 1: Request a Software Reset	R/W

Bit	Symbol	Function	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
16	FTXSTP	Force Tx Stop Mode 0: Finish the Force Stop state 1: Force Data Lanes into transmit mode and generate Stop state	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

SWRST bit (Software Reset)

The SWRST bit is used to request a software reset. See section 58.3.2. Software Reset for the software reset procedure.

FTXSTP bit (Force Tx Stop Mode)

When FTXSTP = 1, the Data Lanes immediately transitions into transmit mode and is forced into the Stop state.

It is prohibited to set the FTXSTP bit to 1 when SWRST = 0. The FTXSTP bit is set during the software reset procedure. See section 58.3.2. Software Reset for the software reset procedure.

58.2.8 RSTSR : Reset Status Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x114

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DL0DIR	—	—	—	—	—	DL1STP	DL0STP	—	—	—	RSTV	RSTAXI	RSTAPB	RSTLP	RSTHS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RSTHS	HS Software Reset Status 0: Not in software reset procedure 1: Running software reset procedure for HS mode	R
1	RSTLP	LP Software Reset Status 0: Not in software reset procedure 1: Running software reset procedure for LP mode	R
2	RSTAPB	APB Software Reset Status 0: Not in software reset procedure 1: Running software reset procedure for APB bus	R
3	RSTAXI	AXI Software Reset Status 0: Not in software reset procedure 1: Running software reset procedure for AXI bus	R
4	RSTV	Video Software Reset Status 0: Not in software reset procedure 1: Running software reset procedure for Video mode	R
7:5	—	These bits are read as 0.	R
8	DL0STP	Data Lane-0 Stop Status 0: Not Stop state 1: Stop state	R
9	DL1STP	Data Lane-1 Stop Status 0: Not Stop state 1: Stop state	R

Bit	Symbol	Function	R/W
14:10	—	These bits are read as 0.	R
15	DL0DIR	Data Lane-0 Direction 0: TX mode 1: RX mode	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

RSTHS bit (HS Software Reset Status)

The RSTHS bit indicates the state of the HS mode initialization process by software reset, which is caused by setting the RSTCR.SWRST bit to 1.

RSTLP bit (LP Software Reset Status)

The RSTLP bit indicates the state of the LP mode initialization process by software reset, which is caused by setting the RSTCR.SWRST bit to 1.

RSTAPB bit (APB Software Reset Status)

The RSTAPB bit indicates the state of the APB initialization process by software reset, which is caused by setting the RSTCR.SWRST bit to 1.

RSTAXI bit (AXI Software Reset Status)

The RSTAXI bit indicates the state of the AXI initialization process by software reset, which is caused by setting the RSTCR.SWRST bit to 1.

RSTV bit (Video Software Reset Status)

The RSTV bit indicates the state of the Video initialization process by software reset, which is caused by setting the RSTCR.SWRST bit to 1.

DL0STP bit (Data Lane-0 Stop Status)

The DL0STP bit indicates the Stop state of the Data Lane-0. The DL0STP bit is a copy of the PLSR.DL0STP bit. The meaning also is same. See section 58.2.40. PLSR : Physical Lane Status Register for details on the PLSR.DL0STP bit.

DL1STP bit (Data Lane-1 Stop Status)

The DL1STP bit indicates the Stop state of the Data Lane-1. The DL1STP bit is a copy of the PLSR.DL1STP bit. The meaning also is same. See section 58.2.40. PLSR : Physical Lane Status Register for details on the PLSR.DL1STP bit.

DL0DIR bit (Data Lane-0 Direction)

The DL0DIR bit indicates the direction of the Data Lane-0. The DL0DIR bit is a copy of the PLSR.DL0DIR bit. The meaning also is same. See section 58.2.40. PLSR : Physical Lane Status Register for details on the PLSR.DL0DIR bit.

58.2.9 DSISETR : DSI Set Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x120

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EOTPEN	EXTEMD	SCREN	—	—	—	—	—	VC3CRCEN	VC2CRCEN	VC1CRCEN	VC0CRCEN	—	—	—	ECCEN
Value after reset:	1	0	0	0	0	0	0	0	1	1	1	1	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MRPSZ[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
15:0	MRPSZ[15:0]	Maximum Return Packet Size Set the maximum return packet size.	R/W
16	ECCEN	ECC Check Enable 0: Disable 1: Enable	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	VC0CRCEN	VC-0 CRC Check Enable 0: Disable 1: Enable	R/W
21	VC1CRCEN	VC-1 CRC Check Enable 0: Disable 1: Enable	R/W
22	VC2CRCEN	VC-2 CRC Check Enable 0: Disable 1: Enable	R/W
23	VC3CRCEN	VC-3 CRC Check Enable 0: Disable 1: Enable	R/W
28:24	—	These bits are read as 0. The write value should be 0.	R/W
29	SCREN	Data Scramble Enable 0: Disable 1: Enable	R/W
30	EXTEMD	External Tearing Effect Detection Sense Select 0: Rising edge 1: Falling edge	R/W
31	EOTPEN	HS Transfer EoTp Enable 0: Disable 1: Enable	R/W

Note: S-TYPE-3, P-TYPE-3

This register can only be set to change during the initialization at initial startup or Software Reset process (section 58.3.2. Software Reset).

MRPSZ[15:0] bits (Maximum Return Packet Size)

The MRPSZ[15:0] bits specify the maximum packet size to be received in LP-RX mode.

If the WC of the returned long packet exceeds the value of the MRPSZ[15:0] bits, Return Packet Size Error is occurred and the RXSR.RSIZEERR flag is set to 1. In this case, no data will be stored in the memory area.

It is prohibited to set the MMRPSZ[15:0] bits to 0.

ECCEN bit (ECC Check Enable)

The ECCEN bit controls the ECC check permission. To enable the ECC check, set the ECCEN bit to 1.

VC0CRCEN bit (VC-0 CRC Check Enable)

The VC0CRCEN bit controls the CRC check permission for Virtual Channel-0 (VC-0). To enable the CRC check for VC-0, set the VC0CRCEN bit to 1.

VC1CRCEN bit (VC-1 CRC Check Enable)

The VC1CRCEN bit controls the CRC check permission for Virtual Channel-1 (VC-1). To enable the CRC check for VC-1, set the VC1CRCEN bit to 1.

VC2CRCEN bit (VC-2 CRC Check Enable)

The VC2CRCEN bit controls the CRC check permission for Virtual Channel-2 (VC-2). To enable the CRC check for VC-2, set the VC2CRCEN bit to 1.

VC3CRCEN bit (VC-3 CRC Check Enable)

The VC3CRCEN bit controls the CRC check permission for Virtual Channel-3 (VC-3). To enable the CRC check for VC-3, set the VC3CRCEN bit to 1.

SCREN bit (Data Scramble Enable)

The SCREN bit controls the data scrambling permission. If the peripheral device does not have a scrambling function, do not set the SCREN bit to 1.

EXTEMD bit (External Tearing Effect Detection Sense Select)

The EXTEMD bit controls the detection edge of the External Tearing Effect (DSI_TE pin) input.

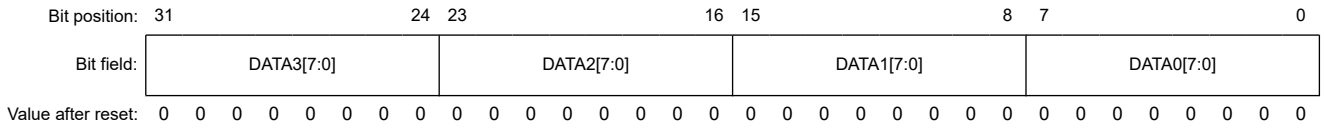
EOTPEN bit (HS Transfer EoTp Enable)

The EOTPEN bit controls the permission of the EoTp transfer in HS-TX mode. To enable to transfer EoTp, set the EOTPEN bit to 1. EoTp is always disabled in LP-TX mode.

58.2.10 TXPPD0R : Transmit Packet Payload Data 0 Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x160



Bit	Symbol	Function	R/W
7:0	DATA0[7:0]	Payload Data 0 Set the Long Packet Payload Data 0 to be transferred in Command mode.*1	R/W
15:8	DATA1[7:0]	Payload Data 1 Set the Long Packet Payload Data 1 to be transferred in Command mode.*1	R/W
23:16	DATA2[7:0]	Payload Data 2 Set the Long Packet Payload Data 2 to be transferred in Command mode.*1	R/W
31:24	DATA3[7:0]	Payload Data 3 Set the Long Packet Payload Data 3 to be transferred in Command mode.*1	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. These bits are valid when the SQCHnDSCmAR.FMT bit is set to 1 and the SQCHnDSCmBR.DTSEL[1:0] bits are set to 00b.

DATA0[7:0] bits (Payload Data 0)

TheDATA0[7:0] bits are used to store the Long Packet payload (Data 0) to be transferred in Command mode using Sequence operation.

DATA1[7:0] bits (Payload Data 1)

TheDATA1[7:0] bits are used to store the Long Packet payload (Data 1) to be transferred in Command mode using Sequence operation.

DATA2[7:0] bits (Payload Data 2)

TheDATA2[7:0] bits are used to store the Long Packet payload (Data 2) to be transferred in Command mode using Sequence operation

DATA3[7:0] bits (Payload Data 3)

TheDATA3[7:0] bits are used to store the Long Packet payload (Data 3) to be transferred in Command mode using Sequence operation.

58.2.11 TXPPD1R : Transmit Packet Payload Data 1 Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x164

Bit position:	31							24	23							16	15							8	7							0
Bit field:	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DATA4[7:0]	Payload Data 4 Set the Long Packet Payload Data 4 to be transferred in Command mode.*1	R/W
15:8	DATA5[7:0]	Payload Data 5 Set the Long Packet Payload Data 5 to be transferred in Command mode.*1	R/W
23:16	DATA6[7:0]	Payload Data 6 Set the Long Packet Payload Data 6 to be transferred in Command mode.*1	R/W
31:24	DATA7[7:0]	Payload Data 7 Set the Long Packet Payload Data 7 to be transferred in Command mode.*1	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. These bits are valid when the SQCHnDSCmAR.FMT bit is set to 1 and the SQCHnDSCmBR.DTSEL[1:0] bits are set to 00b.

DATA4[7:0] bits (Payload Data 4)

The DATA4[7:0] bits are used to store the Long Packet payload (Data 4) to be transferred in Command mode using Sequence operation.

DATA5[7:0] bits (Payload Data 5)

TheDATA5[7:0] bits are used to store the Long Packet payload (Data 5) to be transferred in Command mode using Sequence operation.

DATA6[7:0] bits (Payload Data 6)

The DATA6[7:0] bits are used to store the Long Packet payload (Data 6) to be transferred in Command mode using Sequence operation.

DATA7[7:0] bits (Payload Data 7)

The DATA7[7:0] bits are used to store the Long Packet payload (Data 7) to be transferred in Command mode using Sequence operation.

58.2.12 TXPPD2R : Transmit Packet Payload Data 2 Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x168

Bit position:	31								23							16	15							8	7							0
Bit field:	DATA11[7:0]								DATA10[7:0]								DATA9[7:0]								DATA8[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DATA8[7:0]	Payload Data 8 Set the Long Packet Payload Data 8 to be transferred in Command mode.*1	R/W
15:8	DATA9[7:0]	Payload Data 9 Set the Long Packet Payload Data 9 to be transferred in Command mode.*1	R/W

Bit	Symbol	Function	R/W
23:16	DATA10[7:0]	Payload Data 10 Set the Long Packet Payload Data 10 to be transferred in Command mode.*1	R/W
31:24	DATA11[7:0]	Payload Data 11 Set the Long Packet Payload Data 11 to be transferred in Command mode.*1	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. These bits are valid when the SQCHnDSCmAR.FMT bit is set to 1 and the SQCHnDSCmBR.DTSEL[1:0] bits are set to 00b.

DATA8[7:0] bits (Payload Data 8)

The DATA8[7:0] bits are used to store the Long Packet payload (Data 8) to be transferred in Command mode using Sequence operation.

DATA9[7:0] bits (Payload Data 9)

The DATA9[7:0] bits are used to store the Long Packet payload (Data 9) to be transferred in Command mode using Sequence operation.

DATA10[7:0] bits (Payload Data 10)

The DATA10[7:0] bits are used to store the Long Packet payload (Data 10) to be transferred in Command mode using Sequence operation.

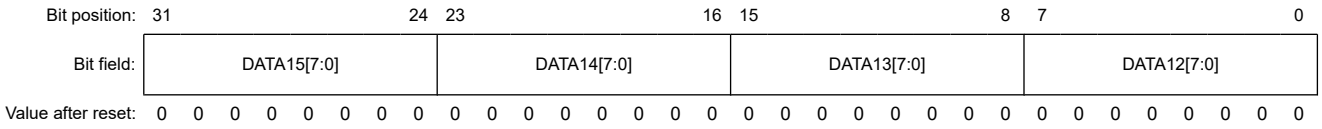
DATA11[7:0] bits (Payload Data 11)

The DATA11[7:0] bits are used to store the Long Packet payload (Data 11) to be transferred in Command mode using Sequence operation.

58.2.13 TXPPD3R : Transmit Packet Payload Data 3 Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x16C



Bit	Symbol	Function	R/W
7:0	DATA12[7:0]	Payload Data 12 Set the Long Packet Payload Data 12 to be transferred in Command mode.*1	R/W
15:8	DATA13[7:0]	Payload Data 13 Set the Long Packet Payload Data 13 to be transferred in Command mode.*1	R/W
23:16	DATA14[7:0]	Payload Data 14 Set the Long Packet Payload Data 14 to be transferred in Command mode.*1	R/W
31:24	DATA15[7:0]	Payload Data 15 Set the Long Packet Payload Data 15 to be transferred in Command mode.*1	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. These bits are valid when the SQCHnDSCmAR.FMT bit is set to 1 and the SQCHnDSCmBR.DTSEL[1:0] bits are set to 00b.

DATA12[7:0] bits (Payload Data 12)

The DATA12[7:0] bits are used to store the Long Packet payload (Data 12) to be transferred in Command mode using Sequence operation.

DATA13[7:0] bits (Payload Data 13)

The DATA13[7:0] bits are used to store the Long Packet payload (Data 13) to be transferred in Command mode using Sequence operation.

DATA14[7:0] bits (Payload Data 14)

The DATA14[7:0] bits are used to store the Long Packet payload (Data 14) to be transferred in Command mode using Sequence operation.

DATA15[7:0] bits (Payload Data 15)

The DATA15[7:0] bits are used to store the Long Packet payload (Data 15) to be transferred in Command mode using Sequence operation.

58.2.14 RXSR : Receive Status Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x200

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	RXAKE	—	ECCERRS	—	RSIZEERR	NORESERR	PRTOERR	RXOVFERR	IBERR	CRCERR	WCERR	—	UNEXERR	ECCERRM	MLFERR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EXTEDET	RXACK	—	—	—	RXEOTP	—	RXRESP	—	—	—	—	—	TATO	LRXHTO	BTAREND
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BTAREND	BTA Request End Interrupt Flag 0: Not detected 1: BTA Completion detected	R
1	LRXHTO	LP-RX Host Processor Timeout Interrupt Flag 0: Not detected 1: LP-RX Host Processor Timeout (LRX-H_TO) detected	R
2	TATO	Turnaround Acknowledge Timeout Interrupt Flag 0: Not detected 1: Turnaround Acknowledge Timeout (TA_TO) detected	R
7:3	—	These bits are read as 0.	R
8	RXRESP	Response Packet Receive Interrupt Flag 0: No response received 1: Response packet received	R
9	—	This bit is read as 0.	R
10	RXEOTP	EoTp Receive Interrupt Flag 0: No received 1: EoTp received	R
13:11	—	These bits are read as 0.	R
14	RXACK	ACK Trigger Receive Interrupt Flag 0: No trigger received 1: ACK trigger received	R
15	EXTEDET	External Tearing Effect Detect Interrupt Flag 0: Not detected 1: External Tearing Effect detected	R
16	MLFERR	Malform Error Interrupt Flag 0: No error received 1: A packet of less than 4 bytes received	R
17	ECCERRM	Multi Bit ECC Error Interrupt Flag 0: No error detected 1: A multi-bit ECC error detected	R

Bit	Symbol	Function	R/W
18	UNEXERR	Unexpected Packet Error Interrupt Flag 0: No error 1: Unexpected packet received	R
19	—	This bit is read as 0.	R
20	WCERR	Word Count Error Interrupt Flag 0: No error detected 1: The length of the received packet is shorter than the WC indicated in the packet header.	R
21	CRCERR	CRC Error Interrupt Flag 0: No error detected 1: CRC error detected	R
22	IBERR	Internal Bus Error Interrupt Flag 0: No error detected 1: Internal AXI bus write failed	R
23	RXOVFERR	Receive Buffer Overflow Error Interrupt Flag 0: No error detected 1: A buffer overflow error detected on receiving long packets	R
24	PRTOERR	Peripheral Response Timeout Error Interrupt Flag 0: No error occurred 1: Peripheral response timeout occurred	R
25	NORESERR	No Response Error Interrupt Flag 0: No error occurred 1: No triggers or packets returned during BTA period	R
26	RSIZEERR	Return Packet Size Error Interrupt Flag 0: No error detected 1: Oversize error in a received long packet detected	R
27	—	This bit is read as 0.	R
28	ECCERRS	Single Bit ECC Error Interrupt Flag 0: No error detected 1: A single-bit ECC error detected	R
29	—	This bit is read as 0.	R
30	RXAKE	Acknowledge and Error Report Receive Interrupt Flag 0: No received 1: An Acknowledge and Error Report packet received	R
31	—	This bit is read as 0.	R

Note: S-TYPE-3, P-TYPE-3

BTAREND bit (BTA Request End Interrupt Flag)

The BTAREND flag indicates that the BTA request specified by the SQCHnDSCmAR.BTA[1:0] bits is completed.

For example, if the SQCHnDSCmAR.BTA[1:0] bits are set to 10b, the BTAREND flag is set to 1 after the reception of the response packet to read request following the BTA is completed.

The BTAREND flag is not cleared automatically, so set the RXSCR.BTAREND bit to 1 to clear the BTAREND flag.

LRXHTO bit (LP-RX Host Processor Timeout Interrupt Flag)

The LRXHTO flag indicates that a LP-RX Host Processor Timeout Interrupt Flag (LRX-H_TO) is detected.

The LRXHTO flag is not cleared automatically, so set the RXSCR.LRXHTO bit to 1 to clear the LRXHTO flag.

TATO bit (Turnaround Acknowledge Timeout Interrupt Flag)

The TATO flag indicates that a Turnaround Acknowledge Timeout (TA_TO) is detected.

The TATO flag is not cleared automatically, so set the RXSCR.TATO bit to 1 to clear the TATO flag.

RXRESP bit (Response Packet Receive Interrupt Flag)

The RXRESP flag indicates that a response packet is received.

The RXRESP flag is not cleared automatically, so set the RXSCR.RXRESP bit to 1 to clear the RXRESP flag.

RXEOTP bit (EoTp Receive Interrupt Flag)

The RXEOTP flag indicates that an EoTp is received.

The RXEOTP flag is not cleared automatically, so set the RXSCR.RXEOTP bit to 1 to clear the RXEOTP flag.

RXACK bit (ACK Trigger Receive Interrupt Flag)

The RXACK flag indicates that an ACK trigger is received.

The RXACK flag is not cleared automatically, so set the RXSCR.RXACK bit to 1 to clear the RXACK flag.

EXTEDET bit (External Tearing Effect Detect Interrupt Flag)

The EXTEDET flag indicates that External Tearing Effect from DSI_TE pin is detected.

The EXTEDET flag is not cleared automatically, so set the RXSCR.EXTEDET bit to 1 to clear the EXTEDET flag.

MLFERR bit (Malform Error Interrupt Flag)

The MLFERR flag indicates that a packet of less than 4 bytes is received.

The MLFERR flag is not cleared automatically, so set the RXSCR.MLFERR bit to 1 to clear the MLFERR flag.

ECCERRM bit (Multi Bit ECC Error Interrupt Flag)

The ECCERRM flag indicates that a multi-bit ECC error is detected in a received packet.

The ECCERRM flag is not cleared automatically, so set the RXSCR.ECCERRM bit to 1 to clear the ECCERRM flag.

UNEXERR bit (Unexpected Packet Error Interrupt Flag)

The UNEXERR flag indicates that an unexpected Data Type (DT) or unexpected response is received.

The UNEXERR flag is not cleared automatically, so set the RXSCR.UNEXERR bit to 1 to clear the UNEXERR flag.

WCERR bit (Word Count Error Interrupt Flag)

The WCERR flag indicates that the length of the received packet payload is shorter than the Word Count (WC) indicated in the packet header.

The WCERR flag is not cleared automatically, so set the RXSCR.WCERR bit to 1 to clear the WCERR flag.

CRCERR bit (CRC Error Interrupt Flag)

The CRCERR flag indicates that a CRC error is detected in a received packet.

The CRCERR flag is not cleared automatically, so set the RXSCR.CRCERR bit to 1 to clear the CRCERR flag.

IBERR bit (Internal Bus Error Interrupt Flag)

The IBERR flag is set to 1 when the internal AXI bus write failed.

The IBERR flag is not cleared automatically, so set the RXSCR.IBERR bit to 1 to clear the IBERR flag.

RXOVFERR bit (Receive Buffer Overflow Error Interrupt Flag)

The RXOVFERR flag is set to 1 when the receive buffer of Long Packet overflows.

The RXOVFERR flag is not cleared automatically, so set the RXSCR.RXOVFERR bit to 1 to clear the RXOVFERR flag.

PRTOERR bit (Peripheral Response Timeout Error Interrupt Flag)

The PRTOERR flag is set to 1 when the timeout for peripheral response expires after giving bus possession to the peripheral and entering in LP-RX mode. The timeout period is specified by the following registers depending on the type of Bus Turn-Around (BTA).

- PRESPTOBTASETR or
- PRESPTOLPSETR.LPRTO[15:0] or
- PRESPTOLPSETR.LPWTO[15:0] or
- PRESPTOHSSETR.HSRTO[15:0] or

- PRESPTOHSSETR.HSWTO[15:0]

The PRTOERR flag is not cleared automatically, so set the RXSCR.PRTOERR bit to 1 to clear the PRTOERR flag.

NORESERR bit (No Response Error Interrupt Flag)

The NORESERR flag is set to 1 when no trigger or packet is returned during BTA period.

The NORESERR flag is not cleared automatically, so set the RXSCR.NORESERR bit to 1 to clear the NORESERR flag.

RSIZEERR bit (Return Packet Size Error Interrupt Flag)

The RSIZEERR flag is set to 1 when the WC value of the received long packet is larger than the size set by the DSISETR.MRPSZ[15:0] bits.

The RSIZEERR flag is not cleared automatically, so set the RXSCR.RSIZEERR bit to 1 to clear the RSIZEERR flag.

ECCERRS bit (Single Bit ECC Error Interrupt Flag)

The ECCERRS flag indicates that a single-bit ECC error is detected and corrected in a received packet.

The ECCERRS flag is not cleared automatically, so set the RXSCR.ECCERRS bit to 1 to clear the ECCERRS flag.

RXAKE bit (Acknowledge and Error Report Receive Interrupt Flag)

The RXAKE flag indicates that the Acknowledge and Error Report packet is received from the peripheral.

The RXAKE flag is not cleared automatically, so set the RXSCR.RXAKE bit to 1 to clear the RXAKE flag.

58.2.15 RXSCR : Receive Status Clear Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x204

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	RXAKE	—	ECCERRS	—	RSIZEERR	NORESERR	PRTOERR	RXOVFERR	IBERR	CRCERR	WCERR	—	UNEXERR	ECCERRM	MLFERR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EXTEDET	RXACK	—	—	—	RXEOTP	—	RXRESP	—	—	—	—	—	TATO	LRXHTO	BTAREND
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BTAREND	BTA Request End Interrupt Flag Clear 0: No operation 1: Clear the RXSR.BTAREND flag	W
1	LRXHTO	LP-RX Host Processor Timeout Interrupt Flag Clear 0: No operation 1: Clear the RXSR.LRXHTO flag	W
2	TATO	Turnaround Acknowledge Timeout Interrupt Flag Clear 0: No operation 1: Clear the RXSR.TATO flag	W
7:3	—	These bits are read as 0. The write value should be 0.	W
8	RXRESP	Response Packet Receive Interrupt Flag Clear 0: No operation 1: Clear the RXSR.RXRESP flag	W
9	—	This bit is read as 0. The write value should be 0.	W
10	RXEOTP	EoTp Receive Interrupt Flag Clear 0: No operation 1: Clear the RXSR.RXEOTP flag	W

Bit	Symbol	Function	R/W
13:11	—	These bits are read as 0. The write value should be 0.	W
14	RXACK	ACK Trigger Receive Interrupt Flag Clear 0: No operation 1: Clear the RXSR.RXACK flag	W
15	EXTEDET	External Tearing Effect Detect Interrupt Flag Clear 0: No operation 1: Clear the RXSR.EXTEDET flag	W
16	MLFERR	Malform Error Interrupt Flag Clear 0: No operation 1: Clear the RXSR.MLFERR flag	W
17	ECCERRM	Multi Bit ECC Error Interrupt Flag Clear 0: No operation 1: Clear the RXSR.ECCERRM flag	W
18	UNEXERR	Unexpected Packet Error Interrupt Flag Clear 0: No operation 1: Clear the RXSR.UNEXERR flag	W
19	—	This bit is read as 0. The write value should be 0.	W
20	WCERR	Word Count Error Interrupt Flag Clear 0: No operation 1: Clear the RXSR.WCERR flag	W
21	CRCERR	CRC Error Interrupt Flag Clear 0: No operation 1: Clear the RXSR.CRCERR flag	W
22	IBERR	Internal Bus Error Interrupt Flag Clear 0: No operation 1: Clear the RXSR.IBERR flag	W
23	RXOVFERR	Receive Buffer Overflow Error Interrupt Flag Clear 0: No operation 1: Clear the RXSR.RXOVFERR flag	W
24	PRTOERR	Peripheral Response Timeout Error Interrupt Flag Clear 0: No operation 1: Clear the RXSR.PRTOERR flag	W
25	NORESERR	No Response Error Interrupt Flag Clear 0: No operation 1: Clear the RXSR.NORESERR flag	W
26	RSIZEERR	Return Packet Size Error Interrupt Flag Clear 0: No operation 1: Clear the RXSR.RSIZEERR flag	W
27	—	This bit is read as 0. The write value should be 0.	W
28	ECCERRS	Single Bit ECC Error Interrupt Flag Clear 0: No operation 1: Clear the RXSR.ECCERRS flag	W
29	—	This bit is read as 0. The write value should be 0.	W
30	RXAKE	Acknowledge and Error Report Receive Interrupt Flag Clear 0: No operation 1: Clear the RXSR.RXAKE flag	W
31	—	This bit is read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

BTAREND bit (BTA Request End Interrupt Flag Clear)

Set the BTAREND bit to 1 to clear the RXSR.BTAREND flag.

LRXHTO bit (LP-RX Host Processor Timeout Interrupt Flag Clear)

Set the LRXHTO bit to 1 to clear the RXSR.LRXHTO flag.

TATO bit (Turnaround Acknowledge Timeout Interrupt Flag Clear)

Set the TATO bit to 1 to clear the RXSR.TATO flag.

RXRESP bit (Response Packet Receive Interrupt Flag Clear)

Set the RXRESP bit to 1 to clear the RXSR.RXRESP flag.

RXEOTP bit (EoTp Receive Interrupt Flag Clear)

Set the RXEOTP bit to 1 to clear the RXSR.RXEOTP flag.

RXACK bit (ACK Trigger Receive Interrupt Flag Clear)

Set the RXACK bit to 1 to clear the RXSR.RXACK flag.

EXTEDET bit (External Tearing Effect Detect Interrupt Flag Clear)

Set the EXTEDET bit to 1 to clear the RXSR.EXTEDET flag.

MLFERR bit (Malform Error Interrupt Flag Clear)

Set the MLFERR bit to 1 to clear the RXSR.MLFERR flag.

ECCERRM bit (Multi Bit ECC Error Interrupt Flag Clear)

Set the ECCERRM bit to 1 to clear the RXSR.ECCERRM flag.

UNEXERR bit (Unexpected Packet Error Interrupt Flag Clear)

Set the UNEXERR bit to 1 to clear the RXSR.UNEXERR flag.

WCERR bit (Word Count Error Interrupt Flag Clear)

Set the WCERR bit to 1 to clear the RXSR.WCERR flag.

CRCERR bit (CRC Error Interrupt Flag Clear)

Set the CRCERR bit to 1 to clear the RXSR.CRCERR flag.

IBERR bit (Internal Bus Error Interrupt Flag Clear)

Set the IBERR bit to 1 to clear the RXSR.IBERR flag.

RXOVFERR bit (Receive Buffer Overflow Error Interrupt Flag Clear)

Set the RXOVFERR bit to 1 to clear the RXSR.RXOVFERR flag.

PRTOERR bit (Peripheral Response Timeout Error Interrupt Flag Clear)

Set the PRTOERR bit to 1 to clear the RXSR.PRTOERR flag.

NORESERR bit (No Response Error Interrupt Flag Clear)

Set the NORESERR bit to 1 to clear the RXSR.NORESERR flag.

RSIZEERR bit (Return Packet Size Error Interrupt Flag Clear)

Set the RSIZEERR bit to 1 to clear the RXSR.RSIZEERR flag.

ECCERRS bit (Single Bit ECC Error Interrupt Flag Clear)

Set the ECCERRS bit to 1 to clear the RXSR.ECCERRS flag.

RXAKE bit (Acknowledge and Error Report Receive Interrupt Flag Clear)

Set the RXAKE bit to 1 to clear the RXSR.RXAKE flag.

58.2.16 RXIER : Receive Interrupt Enable Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x208

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	RXAKE	—	ECCERRS	—	RSIZERR	NORESERR	PRTOERR	RXOVFERR	IBERR	CRCERR	WCERR	—	UNEXERR	ECCERRM	MLFERR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EXTEDET	RXACK	—	—	—	RXEOTP	—	RXRESP	—	—	—	—	—	TATO	LRXHTO	BTAREND
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BTAREND	BTA Request End Interrupt Enable 0: Disable 1: Enable	R/W
1	LRXHTO	LP-RX Host Processor Timeout Interrupt Enable 0: Disable 1: Enable	R/W
2	TATO	Turnaround Acknowledge Timeout Interrupt Enable 0: Disable 1: Enable	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	RXRESP	Response Packet Receive Interrupt Enable 0: Disable 1: Enable	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	RXEOTP	EoTp Receive Interrupt Enable 0: Disable 1: Enable	R/W
13:11	—	These bits are read as 0. The write value should be 0.	R/W
14	RXACK	ACK Trigger Receive Interrupt Enable 0: Disable 1: Enable	R/W
15	EXTEDET	External Tearing Effect Detect Interrupt Enable 0: Disable 1: Enable	R/W
16	MLFERR	Malform Error Interrupt Enable 0: Disable 1: Enable	R/W
17	ECCERRM	Multi Bit ECC Error Interrupt Enable 0: Disable 1: Enable	R/W
18	UNEXERR	Unexpected Packet Error Interrupt Enable 0: Disable 1: Enable	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
20	WCERR	Word Count Error Interrupt Enable 0: Disable 1: Enable	R/W

Bit	Symbol	Function	R/W
21	CRCERR	CRC Error Interrupt Enable 0: Disable 1: Enable	R/W
22	IBERR	Internal Bus Error Interrupt Enable 0: Disable 1: Enable	R/W
23	RXOVFERR	Receive Buffer Overflow Error Interrupt Enable 0: Disable 1: Enable	R/W
24	PRTOERR	Peripheral Response Timeout Error Interrupt Enable 0: Disable 1: Enable	R/W
25	NORESERR	No Response Error Interrupt Enable 0: Disable 1: Enable	R/W
26	RSIZEERR	Return Packet Size Error Interrupt Enable 0: Disable 1: Enable	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
28	ECCERRS	Single Bit ECC Error Interrupt Enable 0: Disable 1: Enable	R/W
29	—	This bit is read as 0. The write value should be 0.	R/W
30	RXAKE	Acknowledge and Error Report Receive Interrupt Enable 0: Disable 1: Enable	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

BTAREND bit (BTA Request End Interrupt Enable)

The BTAREND bit controls BTA Request End Interrupt permission. When a BTA Request End Interrupt occurs, the RXSR.BTAREND flag is set to 1.

To enable BTA Request End Interrupt, set the BTAREND bit to 1.

LRXHTO bit (LP-RX Host Processor Timeout Interrupt Enable)

The LRXHTO bit controls LP-RX Host Processor Timeout Interrupt permission. When an LP-RX Host Processor Timeout Interrupt occurs, the RXSR.LRXHTO flag is set to 1.

To enable LP-RX Host Processor Timeout Interrupt, set the LRXHTO bit to 1.

TATO bit (Turnaround Acknowledge Timeout Interrupt Enable)

The TATO bit controls Turnaround Acknowledge Timeout Interrupt permission. When a Turnaround Acknowledge Timeout Interrupt occurs, the RXSR.TATO flag is set to 1.

To enable Turnaround Acknowledge Timeout Interrupt, set the TATO bit to 1.

RXRESP bit (Response Packet Receive Interrupt Enable)

The RXRESP bit controls Response Packet Receive Interrupt permission. When a Response Packet Receive Interrupt occurs, the RXSR.RXRESP flag is set to 1.

To enable Response Packet Receive Interrupt, set the RXRESP bit to 1.

RXEOTP bit (EoTp Receive Interrupt Enable)

The RXEOTP bit controls EoTp Receive Interrupt permission. When an EoTp Receive Interrupt occurs, the RXSR.RXEOTP flag is set to 1.

To enable EoTp Receive Interrupt, set the RXEOTP bit to 1.

RXACK bit (ACK Trigger Receive Interrupt Enable)

The RXACK bit controls ACK Trigger Receive Interrupt permission. When an ACK Trigger Receive Interrupt occurs, the RXSR.RXACK flag is set to 1.

To enable ACK Trigger Receive Interrupt, set the RXACK bit to 1.

EXTEDET bit (External Tearing Effect Detect Interrupt Enable)

The EXTEDET bit controls External Tearing Effect Detect Interrupt permission. When an External Tearing Effect Detect Interrupt occurs, the RXSR.EXTEDET flag is set to 1.

To enable External Tearing Effect Detect Interrupt, set the EXTEDET bit to 1.

MLFERR bit (Malform Error Interrupt Enable)

The MLFERR bit controls Malform Error Interrupt permission. When a Malform Error Interrupt occurs, the RXSR.MLFERR flag is set to 1.

To enable Malform Error Interrupt, set the MLFERR bit to 1.

ECCERRM bit (Multi Bit ECC Error Interrupt Enable)

The ECCERRM bit controls Multi Bit ECC Error Interrupt permission. When a Multi Bit ECC Error Interrupt occurs, the RXSR.ECCERRM flag is set to 1.

To enable Multi Bit ECC Error Interrupt, set the ECCERRM bit to 1.

UNEXERR bit (Unexpected Packet Error Interrupt Enable)

The UNEXERR bit controls Unexpected Packet Error Interrupt permission. When an Unexpected Packet Error Interrupt occurs, the RXSR.UNEXERR flag is set to 1.

To enable Unexpected Packet Error Interrupt, set the UNEXERR bit to 1.

WCERR bit (Word Count Error Interrupt Enable)

The WCERR bit controls Word Count Error Interrupt permission. When a Word Count Error Interrupt occurs, the RXSR.WCERR flag is set to 1.

To enable Word Count Error Interrupt, set the WCERR bit to 1.

CRCERR bit (CRC Error Interrupt Enable)

The CRCERR bit controls CRC Error Interrupt permission. When a CRC Error Interrupt occurs, the RXSR.CRCERR flag is set to 1.

To enable CRC Error Interrupt, set the CRCERR bit to 1.

IBERR bit (Internal Bus Error Interrupt Enable)

The IBERR bit controls Internal Bus Error Interrupt permission. When an Internal Bus Error Interrupt occurs, the RXSR.IBERR flag is set to 1.

To enable Internal Bus Error Interrupt, set the IBERR bit to 1.

RXOVFERR bit (Receive Buffer Overflow Error Interrupt Enable)

The RXOVFERR bit controls Receive Buffer Overflow Error Interrupt permission. When a Receive Buffer Overflow Error Interrupt occurs, the RXSR.RXOVFERR flag is set to 1.

To enable Receive Buffer Overflow Error Interrupt, set the RXOVFERR bit to 1.

PRTOERR bit (Peripheral Response Timeout Error Interrupt Enable)

The PRTOERR bit controls Peripheral Response Timeout Error Interrupt permission. When a Peripheral Response Timeout Error Interrupt occurs, the RXSR.PRTOERR flag is set to 1.

To enable Peripheral Response Timeout Error Interrupt, set the PRTOERR bit to 1.

NORESERR bit (No Response Error Interrupt Enable)

The NORESERR bit controls No Response Error Interrupt permission. When an No Response Error Interrupt occurs, the RXSR.NORESERR flag is set to 1.

To enable No Response Error Interrupt, set the NORESERR bit to 1.

RSIZEERR bit (Return Packet Size Error Interrupt Enable)

The RSIZEERR bit controls Return Packet Size Error Interrupt permission. When a Return Packet Size Error Interrupt occurs, the RXSR.RSIZEERR flag is set to 1.

To enable Return Packet Size Error Interrupt, set the RSIZEERR bit to 1.

ECCERRS bit (Single Bit ECC Error Interrupt Enable)

The ECCERRS bit controls Single Bit ECC Error Interrupt permission. When a Single Bit ECC Error Interrupt occurs, the RXSR.ECCERRS flag is set to 1.

To enable Single Bit ECC Error Interrupt, set the ECCERRS bit to 1.

RXAKE bit (Acknowledge and Error Report Receive Interrupt Enable)

The RXAKE bit controls Acknowledge and Error Report Receive Interrupt permission. When an Acknowledge and Error Report Receive Interrupt occurs, the RXSR.RXAKE flag is set to 1.

To enable Acknowledge and Error Report Receive Interrupt, set the RXAKE bit to 1.

58.2.17 PRESPTOBTASETR : Peripheral Response Timeout BTA Set Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x210

Bit position:	31																												0
Bit field:	PRTBTA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PRTBTA[31:0]	Peripheral Response Timeout Count Set the Peripheral Response Timeout value for Bus Turn-Around (BTA). If the setting value is 0, timeout is not detected.	R/W

Note: S-TYPE-3, P-TYPE-3

This register can only be set to change during the initialization at initial startup or Software Reset process (section 58.3.2. Software Reset).

The PRTBTA register defines the limit for waiting for a peripheral response after Bus Turn-Around (BTA) executed by settings SQCHnDSCmAR.BTA[1:0] = 11b.

Set the peripheral response timeout value from the time of entering LP-RX mode until the peripheral responds to a BTA.

The timeout value is calculated by the following formula:

Time [µs] = PRTBTA[31:0] × (1 / f_{LPCLK} [MHz]*1)

When a timeout is detected, the RXSR.PRTOERR bit is set to 1.

Note 1. f_{LPCLK} is frequency of Low-Power mode clock (LPCLK).

58.2.18 PRESPTOLPSETR : Peripheral Response Timeout LP Set Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x214

Bit position:	31															16	15													0
Bit field:	LPRTO[15:0]																LPWTO[15:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	LPWTO[15:0]	LPDT WRITE Request Timeout Set the Peripheral Response Timeout value for LPDT WRITE Request. If the setting value is 0, timeout is not detected.	R/W
31:16	LPRTO[15:0]	LPDT READ Request Timeout Set the Peripheral Response Timeout value for LPDT READ Request. If the setting value is 0, timeout is not detected.	R/W

Note: S-TYPE-3, P-TYPE-3

This register can only be set to change during the initialization at initial startup or Software Reset process (section 58.3.2. Software Reset).

LPWTO[15:0] bits (LPDT WRITE Request Timeout)

The LPWTO[15:0] bits define the limit for waiting for a peripheral response after Bus Turn-Around (BTA) executed by settings SQCHnDSCmAR.BTA[1:0] = 01b and SQCHnDSCmAR.SPD = 1.

Set the peripheral response timeout value from the time of entering LP-RX mode until the peripheral responds to a write request.

The timeout value is calculated by the following formula:

Time [µs] = LPWTO[15:0] × (1 / f_{LPCLK} [MHz]*1)

When a timeout is detected, the RXSR.PRTOERR bit is set to 1.

LPRTO[15:0] bits (LPDT READ Request Timeout)

The LPRTO[15:0] bits define the limit for waiting for a peripheral response after Bus Turn-Around (BTA) executed by settings SQCHnDSCmAR.BTA[1:0] = 10b and SQCHnDSCmAR.SPD = 1.

Set the peripheral response timeout value from the time of entering LP-RX mode until the peripheral responds to a read request.

The timeout value is calculated by the following formula:

Time [µs] = LPRTO[15:0] × (1 / f_{LPCLK} [MHz]*1)

When a timeout is detected, the RXSR.PRTOERR bit is set to 1.

Note 1. f_{LPCLK} is frequency of Low-Power mode clock (LPCLK).

58.2.19 PRESPTOHSSETR : Peripheral Response Timeout HS Set Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x218

Bit position:	31															16	15													0
Bit field:	HSRTO[15:0]																HSWTO[15:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	HSWTO[15:0]	HS WRITE Request Timeout Set the Peripheral Response Timeout value for HS WRITE Request. If the setting value is 0, timeout is not detected.	R/W
31:16	HSRTO[15:0]	HS READ Request Timeout Set the Peripheral Response Timeout value for HS READ Request. If the setting value is 0, timeout is not detected.	R/W

Note: S-TYPE-3, P-TYPE-3

This register can only be set to change during the initialization at initial startup or Software Reset process (section 58.3.2. Software Reset).

HSWTO[15:0] bits (HS WRITE Request Timeout)

The HSWTO[15:0] bits define the limit for waiting for a peripheral response after Bus Turn-Around (BTA) executed by settings SQCHnDSCmAR.BTA[1:0] = 01b and SQCHnDSCmAR.SPD = 0.

Set the peripheral response timeout value from the time of entering LP-RX mode until the peripheral responds to a write request.

The timeout value is calculated by the following formula:

Time [µs] = HSWTO[15:0] × (1 / f_{LPCLK} [MHz]*1)

When a timeout is detected, the RXSR.PRTOERR bit is set to 1.

HSRTO[15:0] bits (HS READ Request Timeout)

The HSRTO[15:0] bits define the limit for waiting for a peripheral response after Bus Turn-Around (BTA) executed by settings SQCHnDSCmAR.BTA[1:0] = 10b and SQCHnDSCmAR.SPD = 0.

Set the peripheral response timeout value from the time of entering LP-RX mode until the peripheral responds to a read request.

The timeout value is calculated by the following formula:

Time [µs] = HSRTO[15:0] × (1 / f_{LPCLK} [MHz]*1)

When a timeout is detected, the RXSR.PRTOERR bit is set to 1.

Note 1. f_{LPCLK} is frequency of Low-Power mode clock (LPCLK).

58.2.20 AKEPLATIR : Acknowledge and Error Report Packet Parameter Latest Info Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x220

Bit position:	31												19		16	15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	VC[3:0]			EREP[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	EREP[15:0]	Error Report Received Error Report bits 15-0	R
19:16	VC[3:0]	Virtual Channel ID An Acknowledge and Error Report was received from the following Virtual Channel ID. 0x0: No Error Report received 0x1: From VC-0 0x2: From VC-1 0x4: From VC-2 0x8: From VC-3 Others: Reserved	R
31:20	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

EREP[15:0] bits (Error Report)

When an Acknowledge and Error Report packet is received from the peripheral device, the Error Report is stored in the EREP[15:0] bits.

VC[3:0] bits (Virtual Channel ID)

The VC[3:0] bits are indicated from which virtual channel the Error Report stored in the EREP[15:0] bits was received.

58.2.21 AKEPACMSR : Acknowledge and Error Report Packet Parameter Accumulate Status Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x224

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	AVC3	AVC2	AVC1	AVC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AEREP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	AEREP[15:0]	Accumulated Error Report Accumulated Error Report bits 15-0	R
16	AVC0	Virtual Channel-0 Accumulated Information 0: No Error Report received from VC-0 1: Received an Acknowledge and Error Report from VC-0	R
17	AVC1	Virtual Channel-1 Accumulated Information 0: No Error Report received from VC-1 1: Received an Acknowledge and Error Report from VC-1	R
18	AVC2	Virtual Channel-2 Accumulated Information 0: No Error Report received from VC-2 1: Received an Acknowledge and Error Report from VC-2	R
19	AVC3	Virtual Channel-3 Accumulated Information 0: No Error Report received from VC-3 1: Received an Acknowledge and Error Report from VC-3	R
31:20	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

AEREP[15:0] bits (Accumulated Error Report)

The AKEPLATIR.EREP[15:0] bits contain the most recently received Error Report, while the AEREP[15:0] bits are the accumulated value of past Error Reports.

AVC0 bit (Virtual Channel-0 Accumulated Information)

The AKEPLATIR.VC[3:0] bits contain the virtual channel ID of the most recently received Error Report, while the AVC0 bit is the accumulated value received from Virtual Channel-0.

If the AVC0 bit is set to 1, it means that an Acknowledge and Error Report from Virtual Channel-0 has been received one or more times.

AVC1 bit (Virtual Channel-1 Accumulated Information)

The AKEPLATIR.VC[3:0] bits contain the virtual channel ID of the most recently received Error Report, while the AVC1 bit is the accumulated value received from Virtual Channel-1.

If the AVC1 bit is set to 1, it means that an Acknowledge and Error Report from Virtual Channel-1 has been received one or more times.

AVC2 bit (Virtual Channel-2 Accumulated Information)

The AKEPLATIR.VC[3:0] bits contain the virtual channel ID of the most recently received Error Report, while the AVC2 bit is the accumulated value received from Virtual Channel-2.

If the AVC2 bit is set to 1, it means that an Acknowledge and Error Report from Virtual Channel-2 has been received one or more times.

AVC3 bit (Virtual Channel-3 Accumulated Information)

The AKEPLATIR.VC[3:0] bits contain the virtual channel ID of the most recently received Error Report, while the AVC3 bit is the accumulated value received from Virtual Channel-3.

If the AVC3 bit is set to 1, it means that an Acknowledge and Error Report from Virtual Channel-3 has been received one or more times.

58.2.22 AKEPSCR : Acknowledge and Error Report Packet Parameter Status Clear Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x228

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	AVC3	AVC2	AVC1	AVC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AEREP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	AEREP[15:0]	Accumulated Error Report Clear 0: No operation 1: Clear the AKEPACMSR.AEREP[15:0] bits	W
16	AVC0	Virtual Channel-0 Accumulated Information Clear 0: No operation 1: Clear the AKEPACMSR.AVC0 bit	W
17	AVC1	Virtual Channel-1 Accumulated Information Clear 0: No operation 1: Clear the AKEPACMSR.AVC1 bit	W
18	AVC2	Virtual Channel-2 Accumulated Information Clear 0: No operation 1: Clear the AKEPACMSR.AVC2 bit	W
19	AVC3	Virtual Channel-3 Accumulated Information Clear 0: No operation 1: Clear the AKEPACMSR.AVC3 bit	W
31:20	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

AEREP[15:0] bits (Accumulated Error Report Clear)

Set the AEREP[15:0] bits to 1 to clear the AKEPACMSR.AEREP[15:0] bits.

Each of the AEREP[15:0] bits correspond to each of the AKEPACMSR.AEREP[15:0] bits to be cleared.

AVC0 bit (Virtual Channel-0 Accumulated Information Clear)

Set the AVC0 bit to 1 to clear the AKEPACMSR.AVC0 bit.

AVC1 bit (Virtual Channel-1 Accumulated Information Clear)

Set the AVC1 bit to 1 to clear the AKEPACMSR.AVC1 bit.

AVC2 bit (Virtual Channel-2 Accumulated Information Clear)

Set the AVC2 bit to 1 to clear the AKEPACMSR.AVC2 bit.

AVC3 bit (Virtual Channel-3 Accumulated Information Clear)

Set the AVC3 bit to 1 to clear the AKEPACMSR.AVC3 bit.

58.2.23 RXRSSR : Receive Result Saved Status Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x230

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	SLT3VLD	SLT2VLD	SLT1VLD	SLT0VLD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SLT0VLD	Slot-0 Valid Flag 0: None received 1: Response packet is received and stored in the RXRSS0R register	R
1	SLT1VLD	Slot-1 Valid Flag 0: None received 1: Response packet is received and stored in the RXRSS1R register	R
2	SLT2VLD	Slot-2 Valid Flag 0: None received 1: Response packet is received and stored in the RXRSS2R register	R
3	SLT3VLD	Slot-3 Valid Flag 0: None received 1: Response packet is received and stored in the RXRSS3R register	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

SLT0VLD bit (Slot-0 Valid Flag)

The SLT0VLD flag indicates that the response packet is received and stored in the RXRSS0R register.

The SLT0VLD flag is set to 1 if a response packet is received when the SQCHnDSCmCR.ACTCODE[7:0] bits are set to 0x00b.

The SLT0VLD flag is not cleared automatically, so set the RXRSSCR.SLT0VLD bit to 1 to clear the SLT0VLD flag.

SLT1VLD bit (Slot-1 Valid Flag)

The SLT1VLD flag indicates that the response packet is received and stored in the RXRSS1R register.

The SLT1VLD flag is set to 1 if a response packet is received when the SQCHnDSCmCR.ACTCODE[7:0] bits are set to 0x01.

The SLT1VLD flag is not cleared automatically, so set the RXRSSCR.SLT1VLD bit to 1 to clear the SLT1VLD flag.

SLT2VLD bit (Slot-2 Valid Flag)

The SLT2VLD flag indicates that the response packet is received and stored in the RXRSS2R register.

The SLT2VLD flag is set to 1 if a response packet is received when the SQCHnDSCmCR.ACTCODE[7:0] bits are set to 0x02.

The SLT2VLD flag is not cleared automatically, so set the RXRSSCR.SLT2VLD bit to 1 to clear the SLT2VLD flag.

SLT3VLD bit (Slot-3 Valid Flag)

The SLT3VLD flag indicates that the response packet is received and stored in the RXRSS3R register.

The SLT3VLD flag is set to 1 if a response packet is received when the SQCHnDSCmCR.ACTCODE[7:0] bits are set to 0x03.

The SLT3VLD flag is not cleared automatically, so set the RXRSSCR.SLT3VLD bit to 1 to clear the SLT3VLD flag.

58.2.24 RXRSSCR : Receive Result Saved Status Clear Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x234

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	SLT3VLD	SLT2VLD	SLT1VLD	SLT0VLD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SLT0VLD	Slot-0 Valid Flag Clear 0: No operation 1: Clear the RXRSSR.SLT0VLD flag	W
1	SLT1VLD	Slot-1 Valid Flag Clear 0: No operation 1: Clear the RXRSSR.SLT1VLD flag	W
2	SLT2VLD	Slot-2 Valid Flag Clear 0: No operation 1: Clear the RXRSSR.SLT2VLD flag	W
3	SLT3VLD	Slot-3 Valid Flag Clear 0: No operation 1: Clear the RXRSSR.SLT3VLD flag	W
31:4	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

SLT0VLD bit (Slot-0 Valid Flag Clear)

Set the SLT0VLD bit to 1 to clear the RXRSSR.SLT0VLD flag.

SLT1VLD bit (Slot-1 Valid Flag Clear)

Set the SLT1VLD bit to 1 to clear the RXRSSR.SLT1VLD flag.

SLT2VLD bit (Slot-2 Valid Flag Clear)

Set the SLT2VLD bit to 1 to clear the RXRSSR.SLT2VLD flag.

SLT3VLD bit (Slot-3 Valid Flag Clear)

Set the SLT3VLD bit to 1 to clear the RXRSSR.SLT3VLD flag.

58.2.25 RXRINFOOWSR : Receive Result Info Overwrite Status Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x238

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	SL3OW	SL2OW	SL1OW	SL0OW
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SL0OW	Slot-0 Information Overwrite Flag 0: No overwritten 1: Slot-0 information overwritten	R
1	SL1OW	Slot-1 Information Overwrite Flag 0: No overwritten 1: Slot-1 information overwritten	R
2	SL2OW	Slot-2 Information Overwrite Flag 0: No overwritten 1: Slot-2 information overwritten	R
3	SL3OW	Slot-3 Information Overwrite Flag 0: No overwritten 1: Slot-3 information overwritten	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

SL0OW bit (Slot-0 Information Overwrite Flag)

The SL0OW flag is copy of the RXRSS0R.INFOOW flag. The meaning also is same. For details, see section 58.2.27. RXRSSxR : Receive Result Save Slot-x Register (x = 0 to 3).

SL1OW bit (Slot-1 Information Overwrite Flag)

The SL1OW flag is copy of the RXRSS1R.INFOOW flag. The meaning also is same. For details, see section 58.2.27. RXRSSxR : Receive Result Save Slot-x Register (x = 0 to 3).

SL2OW bit (Slot-2 Information Overwrite Flag)

The SL2OW flag is copy of the RXRSS2R.INFOOW flag. The meaning also is same. For details, see section 58.2.27. RXRSSxR : Receive Result Save Slot-x Register (x = 0 to 3).

SL3OW bit (Slot-3 Information Overwrite Flag)

The SL3OW flag is copy of the RXRSS3R.INFOOW flag. The meaning also is same. For details, see section 58.2.27. RXRSSxR : Receive Result Save Slot-x Register (x = 0 to 3).

58.2.26 RXRINFOOWSCR : Receive Result Info Overwrite Status Clear Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x23C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	SL3OW	SL2OW	SL1OW	SL0OW
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SL0OW	Slot-0 Information Overwrite Flag Clear 0: No operation 1: Clear the RXRINFOOWSR.SL0OW flag	W
1	SL1OW	Slot-1 Information Overwrite Flag Clear 0: No operation 1: Clear the RXRINFOOWSR.SL1OW flag	W
2	SL2OW	Slot-2 Information Overwrite Flag Clear 0: No operation 1: Clear the RXRINFOOWSR.SL2OW flag	W
3	SL3OW	Slot-3 Information Overwrite Flag Clear 0: No operation 1: Clear the RXRINFOOWSR.SL3OW flag	W
31:4	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

SL0OW bit (Slot-0 Information Overwrite Flag Clear)

Set the SL0OW bit to 1 to clear the RXRINFOOWSR.SL0OW flag (the RXRSS0R.INFOOW bit).

SL1OW bit (Slot-1 Information Overwrite Flag Clear)

Set the SL1OW bit to 1 to clear the RXRINFOOWSR.SL1OW flag (the RXRSS1R.INFOOW bit).

SL2OW bit (Slot-2 Information Overwrite Flag Clear)

Set the SL2OW bit to 1 to clear the RXRINFOOWSR.SL2OW flag (the RXRSS2R.INFOOW bit).

SL3OW bit (Slot-3 Information Overwrite Flag Clear)

Set the SL3OW bit to 1 to clear the RXRINFOOWSR.SL3OW flag (the RXRSS3R.INFOOW bit).

58.2.27 RXRSSxR : Receive Result Save Slot-x Register (x = 0 to 3)

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x240 + 0x4 × x

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	INFOOW	RXAKE	RXCERR	RXPFAIL	RXFAIL	RXFERR	RXSUC	FMT	VC[1:0]		DT[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DATA1[7:0]								DATA0[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DATA0[7:0]	Data 0 Data0 of received packet header	R
15:8	DATA1[7:0]	Data 1 Data1 of received packet header	R
21:16	DT[5:0]	Data Type Data Type	R
23:22	VC[1:0]	Virtual Channel Virtual Channel ID	R
24	FMT	Packet Format 0: Short packet 1: Long packet	R
25	RXSUC	Receive Success 0: No received 1: Response packet or ACK trigger received	R
26	RXFERR	Fatal Error 0: No fatal error 1: Fatal timeout occurred during BTA	R
27	RXFAIL	Receive Fail 0: No error 1: Expected receive not done	R
28	RXPFAIL	Receive Packet Data Fail 0: No error 1: Payload data not saved correctly	R
29	RXCERR	Receive Correctable Error 0: No error 1: Correctable error detected	R
30	RXAKE	Receive Acknowledge and Error Report Packet 0: No received 1: An Acknowledge and Error Report packet received	R
31	INFOOW	Information Overwrite 0: No update 1: This register information (RXRSSxR[30:0]) was overwritten	R

Note: S-TYPE-3, P-TYPE-3

DATA0[7:0] bits (Data 0)

Data0 of the received packet header are stored in the DATA0[7:0] bits.

When the received packet is Long Packet, the lower 8 bits of the word count are stored in the DATA0[7:0] bits.

The DATA0[7:0] bits are valid when the RXSUC bit is 1 and the DT[5:0] bits are not 0x00.

DATA1[7:0] bits (Data 1)

Data1 of the received packet header are stored in the DATA1[7:0] bits.

When the received packet is Long Packet, the upper 8 bits of the word count are stored in the DATA1[7:0] bits.

The DATA1[7:0] bits are valid when the RXSUC bit is 1 and the DT[5:0] bits are not 0x00.

DT[5:0] bits (Data Type)

The DT[5:0] bits indicate the Data Type of the received packet header.

When an ACK trigger is received, the DT[5:0] bits are set to 0x00.

The DT[5:0] bits are valid when the RXSUC bit is 1.

VC[1:0] bits (Virtual Channel)

The VC[1:0] bits indicate the Virtual Channel ID of the received packet header.

The VC[1:0] bits are valid when the RXSUC bit is 1 and the DT[5:0] bits are not 0x00.

FMT bit (Packet Format)

The FMT bit indicates the packet format of the received packet header.

The FMT bit is valid when the RXSUC bit is 1 and the DT[5:0] bits are not 0x00. It is also valid when the RXPFAIL bit is set to 1

RXSUC bit (Receive Success)

The RXSUC bit indicates that a response packet or ACK trigger is received.

The RXSR.RXRESP flag or the RXSR.RXACK flag is also set to 1.

RXFERR bit (Fatal Error)

The RXFERR bit indicates that a fatal timeout occurred during BTA.

The FERRSR.TATO flag or the FERRSR.LRXHTO flag is also set to 1.

RXFAIL bit (Receive Fail)

The RXFAIL bit indicates that an expected receive did not done.

One or more of the following flags are also set to 1.

- RXSR.PRTOERR
- RXSR.ECCERRM
- RXSR.MLFERR
- RXSR.NORESERR

RXPFAIL bit (Receive Packet Data Fail)

The RXPFAIL bit indicates that the packet header is saved correctly but payload data was not saved correctly.

One or more of the following flags are also set to 1.

- RXSR.CRCERR
- RXSR.WCERR
- RXSR.RSIZEERR
- RXSR.UNEXERR
- RXSR.RXOVFERR
- RXSR.IBERR

When the RXPFAIL bit and the RXSR.RXRESP bit are set to 1, it means that the packet received during BTA is redundant.

RXCERR bit (Receive Correctable Error)

The RXCERR bit indicates that the received packet has correctable error.

The RXSR.ECCERRS flag is also set to 1.

RXAKE bit (Receive Acknowledge and Error Report Packet)

The RXAKE bit indicates that an Acknowledge and Error Report packet is received.

The RXSR.RXAKE flag is also set to 1.

INFOOW bit (Information Overwrite)

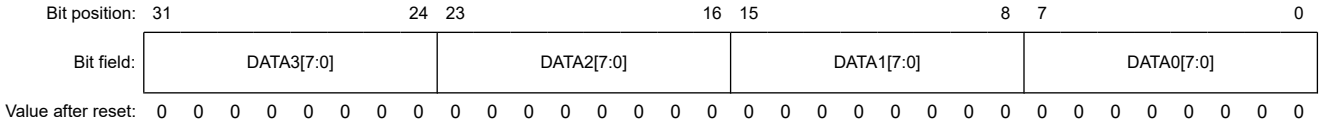
The INFOOW bit indicates that this register information (RXRSSxR[30:0]) are updated when RXRSSR.SLTxVLD = 1.

Set the RXRINFOOWSCR.SLxOW bit to 1 to clear the INFOOW bit.

58.2.28 RXPPD0R : Receive Packet Payload Data 0 Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x2C0



Bit	Symbol	Function	R/W
7:0	DATA0[7:0]	Payload Data 0 Long Packet Payload Data 0 received in Command mode*1	R
15:8	DATA1[7:0]	Payload Data 1 Long Packet Payload Data 1 received in Command mode*1	R
23:16	DATA2[7:0]	Payload Data 2 Long Packet Payload Data 2 received in Command mode*1	R
31:24	DATA3[7:0]	Payload Data 3 Long Packet Payload Data 3 received in Command mode*1	R

Note: S-TYPE-3, P-TYPE-3
Note 1. These bits are valid when SQCHnDSCmBR.DTSEL[1:0] = 00b.

DATA0[7:0] bits (Payload Data 0)

The DATA0[7:0] bits are used to store the Long Packet payload (Data 0) to be received in Command mode using Sequence operation.

DATA1[7:0] bits (Payload Data 1)

The DATA1[7:0] bits are used to store the Long Packet payload (Data 1) to be received in Command mode using Sequence operation.

DATA2[7:0] bits (Payload Data 2)

The DATA2[7:0] bits are used to store the Long Packet payload (Data 2) to be received in Command mode using Sequence operation.

DATA3[7:0] bits (Payload Data 3)

The DATA3[7:0] bits are used to store the Long Packet payload (Data 3) to be received in Command mode using Sequence operation.

58.2.29 RXPPD1R : Receive Packet Payload Data 1 Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x2C4

Bit position:	31							24	23							16	15							8	7							0
Bit field:	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DATA4[7:0]	Payload Data 4 Long Packet Payload Data 4 received in Command mode*1	R
15:8	DATA5[7:0]	Payload Data 5 Long Packet Payload Data 5 received in Command mode*1	R
23:16	DATA6[7:0]	Payload Data 6 Long Packet Payload Data 6 received in Command mode*1	R
31:24	DATA7[7:0]	Payload Data 7 Long Packet Payload Data 7 received in Command mode*1	R

Note: S-TYPE-3, P-TYPE-3
Note 1. These bits are valid when SQCHnDSCmBR.DTSEL[1:0] = 00b.

DATA4[7:0] bits (Payload Data 4)

The DATA4[7:0] bits are used to store the Long Packet payload (Data 4) to be received in Command mode using Sequence operation.

DATA5[7:0] bits (Payload Data 5)

The DATA5[7:0] bits are used to store the Long Packet payload (Data 5) to be received in Command mode using Sequence operation.

DATA6[7:0] bits (Payload Data 6)

The DATA6[7:0] bits are used to store the Long Packet payload (Data 6) to be received in Command mode using Sequence operation.

DATA7[7:0] bits (Payload Data 7)

The DATA7[7:0] bits are used to store the Long Packet payload (Data 7) to be received in Command mode using Sequence operation.

58.2.30 RXPPD2R : Receive Packet Payload Data 2 Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x2C8

Bit position:	31							24	23							16	15							8	7							0
Bit field:	DATA11[7:0]								DATA10[7:0]								DATA9[7:0]								DATA8[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DATA8[7:0]	Payload Data 8 Long Packet Payload Data 8 received in Command mode*1	R
15:8	DATA9[7:0]	Payload Data 9 Long Packet Payload Data 9 received in Command mode*1	R

Bit	Symbol	Function	R/W
23:16	DATA10[7:0]	Payload Data 10 Long Packet Payload Data 10 received in Command mode*1	R
31:24	DATA11[7:0]	Payload Data 11 Long Packet Payload Data 11 received in Command mode*1	R

Note: S-TYPE-3, P-TYPE-3
Note 1. These bits are valid when the SQCHnDSCmBR.DTSEL[1:0] bits are set to 00b.

DATA8[7:0] bits (Payload Data 8)

The DATA8[7:0] bits are used to store the Long Packet payload (Data 8) to be received in Command mode using Sequence operation.

DATA9[7:0] bits (Payload Data 9)

The DATA9[7:0] bits are used to store the Long Packet payload (Data 9) to be received in Command mode using Sequence operation.

DATA10[7:0] bits (Payload Data 10)

The DATA10[7:0] bits are used to store the Long Packet payload (Data 10) to be received in Command mode using Sequence operation.

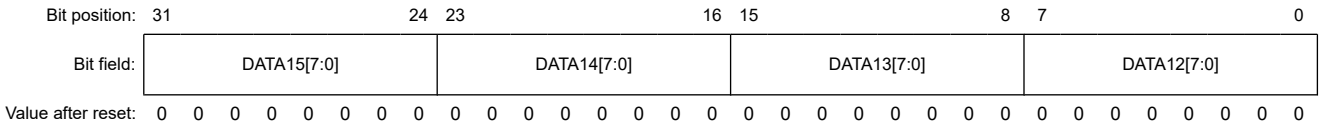
DATA11[7:0] bits (Payload Data 11)

The DATA11[7:0] bits are used to store the Long Packet payload (Data 11) to be received in Command mode using Sequence operation.

58.2.31 RXPPD3R : Receive Packet Payload Data 3 Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x2CC



Bit	Symbol	Function	R/W
7:0	DATA12[7:0]	Payload Data 12 Long Packet Payload Data 12 received in Command mode*1	R
15:8	DATA13[7:0]	Payload Data 13 Long Packet Payload Data 13 received in Command mode*1	R
23:16	DATA14[7:0]	Payload Data 14 Long Packet Payload Data 14 received in Command mode*1	R
31:24	DATA15[7:0]	Payload Data 15 Long Packet Payload Data 15 received in Command mode*1	R

Note: S-TYPE-3, P-TYPE-3
Note 1. These bits are valid when SQCHnDSCmBR.DTSEL[1:0] = 00b.

DATA12[7:0] bits (Payload Data 12)

The DATA12[7:0] bits are used to store the Long Packet payload (Data 12) to be received in Command mode using Sequence operation.

DATA13[7:0] bits (Payload Data 13)

The DATA13[7:0] bits are used to store the Long Packet payload (Data 13) to be received in Command mode using Sequence operation.

DATA14[7:0] bits (Payload Data 14)

The DATA14[7:0] bits are used to store the Long Packet payload (Data 14) to be received in Command mode using Sequence operation.

DATA15[7:0] bits (Payload Data 15)

The DATA15[7:0] bits are used to store the Long Packet payload (Data 15) to be received in Command mode using Sequence operation.

58.2.32 HSTXTOSETR : HS TX Timeout Set Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x2E0

Bit position:	31																												0
Bit field:	HTXTO[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	HTXTO[31:0]	HS TX Timeout Count Set HS TX Timeout (HTX_TO) value.*1	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. If the setting value is 0, timeout is not detected.

This register can only be set to change during the initialization at initial startup or Software Reset process (section 58.3.2. Software Reset).

The HSTXTO register defines the limit for waiting for DSI Host own length of HS Transmission.

The timeout value is calculated by the following formula:

Time [µs] = HSTXTOSETR[31:0] × 32 × (period of High-Speed serial UI [ns]) / 1000

When a timeout is detected, the FERRSR.HTXTO bit is set to 1.

58.2.33 LRXHTOSETR : LRX-H Timeout Set Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x2E4

Bit position:	31																												0
Bit field:	LRXHTO[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	LRXHTO[31:0]	LP-RX Host Processor Timeout Set LP-RX Timeout (LRX-H_TO) value. If the setting value is 0, timeout is not detected.	R/W

Note: S-TYPE-3, P-TYPE-3

This register can only be set to change during the initialization at initial startup or Software Reset process (section 58.3.2. Software Reset).

The LRXHTO register defines the limit for waiting for transmission from the peripheral. If a LP TX-Peripheral timeout (LTX-P_TO) occurs on the peripheral side, then this LP-RX timeout will occur on DSI Host.

The timeout value is calculated by the following formula:

Time [µs] = LRXHTOSETR[31:0] × (1 / f_{LPCLK} [MHz]*1)

When a timeout is detected, the FERRSR.LRXHTO bit is set to 1.

Note 1. f_{LPCLK} is frequency of Low-Power mode clock (LPCLK).

58.2.34 TATOSETR : TA Timeout Set Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x2E8

Bit position:	31																												0
Bit field:	TATO[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TATO[31:0]	Turnaround Acknowledge Timeout Set Turnaround Acknowledge Timeout (TA_TO) value. If the setting value is 0, timeout is not detected.	R/W

Note: S-TYPE-3, P-TYPE-3

This register can only be set to change during the initialization at initial startup or Software Reset process (section 58.3.2. Software Reset).

The TATO register defines the limit for waiting for the data lane transition sequence processing after Bus Turn-Around (BTA) is issued.

The timeout value is calculated by the following formula:

Time [µs] = TATOSETR[31:0] × (1 / f_{LPCLK} [MHz]*1)

When a timeout is detected, the FERRSR.TATO bit is set to 1.

Note 1. f_{LPCLK} is frequency of Low-Power mode clock (LPCLK).

58.2.35 FERRSR : Fatal Error Status Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x300

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	CLP1S	CLP0S	—	—	—	—	—	—	CLP1	CLP0	CTRL	SYNC ESC	ESCE NT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	TATO	LRXH TO	HTXT O
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	HTXTO	HS TX Timeout Interrupt Flag 0: No detected 1: HS TX Timeout (HTX_TO) detected	R
1	LRXHTO	LP-RX Host Processor Timeout Interrupt Flag 0: No detected 1: LP-RX Host Processor Timeout (LRX-H_TO) detected	R
2	TATO	Turnaround Acknowledge Timeout Interrupt Flag 0: No detected 1: Turnaround Acknowledge Timeout (TA_TO) detected	R
15:3	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
16	ESCENT	Escape mode Entry Error Interrupt Flag 0: No error detected 1: Escape mode Entry Error (ErrEsc) detected	R
17	SYNCESC	LPDT Sync Error Interrupt Flag 0: No error detected 1: LPDT Synchronization Error (ErrSyncEsc) detected	R
18	CTRL	Control Error Interrupt Flag 0: No error detected 1: Control Error (ErrControl) detected	R
19	CLP0	LP0 Contention Error Interrupt Flag 0: No error detected 1: LP0 Contention Error (ErrContentionLP0) detected	R
20	CLP1	LP1 Contention Error Interrupt Flag 0: No error detected 1: LP1 Contention Error (ErrContentionLP1) detected	R
26:21	—	These bits are read as 0.	R
27	CLP0S	LP0 Contention Error Status 0: Normal state 1: LP0 Contention Error (ErrContentionLP0) state	R
28	CLP1S	LP1 Contention Error Status 0: Normal state 1: LP1 Contention Error (ErrContentionLP1) state	R
31:29	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

HTXTO flag (HS TX Timeout Interrupt Flag)

The HTXTO flag indicates that a HS TX Timeout (HTX_TO) is detected.

The HTXTO flag is set to 1 when the HS TX Timeout Timer expires. This timer monitors the DSI Host's own length of HS transmission.

The HTXTO flag is not cleared automatically, so set the FERRSCR.HTXTO bit to 1 to clear the HTXTO flag.

LRXHTO flag (LP-RX Host Processor Timeout Interrupt Flag)

The LRXHTO flag indicates that a LP-RX Host Processor Timeout (LRX-H_TO) is detected.

The LRXHTO flag is set to 1 when the Host Processor LP-RX Timer expires.

The LRXHTO flag is not cleared automatically, so set the FERRSCR.LRXHTO bit to 1 to clear the LRXHTO flag.

TATO flag (Turnaround Acknowledge Timeout Interrupt Flag)

The TATO flag indicates that a Turnaround Acknowledge Timeout (TA_TO) is detected.

The TATO flag is set to 1 when the timer TA_TO expires.

The TATO flag is not cleared automatically, so set the FERRSCR.TATO bit to 1 to clear the TATO flag.

ESCENT flag (Escape mode Entry Error Interrupt Flag)

The ESCENT flag indicates that an Escape mode Entry Error (ErrEsc) is detected.

The ESCENT flag is not cleared automatically, so set the FERRSCR.ESCENT bit to 1 to clear the ESCENT flag.

SYNCESC flag (LPDT Sync Error Interrupt Flag)

The SYNCESC flag indicates that a Low-Power Data Transmission Synchronization Error (ErrSyncEsc) is detected.

The SYNCESC flag is not cleared automatically, so set the FERRSCR.SYNCESC bit to 1 to clear the SYNCESC flag.

CTRL flag (Control Error Interrupt Flag)

The CTRL flag indicates that a Control Error (ErrControl) is detected.

The CTRL flag is not cleared automatically, so set the FERRSCR.CTRL bit to 1 to clear the CTRL flag.

CLP0 flag (LP0 Contention Error Interrupt Flag)

The CLP0 flag indicates that a LP0 Contention Error (ErrContentionLP0) is detected.

Once the CLP0S bit is detected to be 1, the CLP0 flag is set to 1.

The CLP0 flag is not cleared automatically, so set the FERRSCR.CLP0 bit to 1 to clear the CLP0 flag.

CLP1 flag (LP1 Contention Error Interrupt Flag)

The CLP1 flag indicates that a LP1 Contention Error (ErrContentionLP1) is detected.

Once the CLP1S bit is detected to be 1, the CLP1 flag is set to 1.

The CLP1 flag is not cleared automatically, so set the FERRSCR.CLP1 bit to 1 to clear the CLP1 flag.

CLP0S bit (LP0 Contention Error Status)

The CLP0S bit indicates the latest LP0 Contention Error state of the D-PHY.

The CLP0S bit is set to 1 when the D-PHY Lane module detects a contention situation on a line while trying to drive the line low.

CLP1S bit (LP1 Contention Error Status)

The CLP1S bit indicates the latest LP1 Contention Error state of the D-PHY.

The CLP1S bit is set to 1 when the D-PHY Lane module detects a contention situation on a line while trying to drive the line high.

58.2.36 FERRSCR : Fatal Error Status Clear Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x304

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	CLP1	CLP0	CTRL	SYNCESC	ESCENT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	TATO	LRXHTO	HTXTO
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	HTXTO	HS TX Timeout Interrupt Flag Clear 0: No operation 1: Clear the FERRSR.HTXTO flag	W
1	LRXHTO	LP-RX Host Processor Timeout Interrupt Flag Clear 0: No operation 1: Clear the FERRSR.LRXHTO flag	W
2	TATO	Turnaround Acknowledge Timeout Interrupt Flag Clear 0: No operation 1: Clear the FERRSR.TATO flag	W
15:3	—	These bits are read as 0. The write value should be 0.	W
16	ESCENT	Escape mode Entry Error Interrupt Flag Clear 0: No operation 1: Clear the FERRSR.ESCENT flag	W
17	SYNCESC	LPDT Sync Error Interrupt Flag Clear 0: No operation 1: Clear the FERRSR.SYNCESC flag	W

Bit	Symbol	Function	R/W
18	CTRL	Control Error Interrupt Flag Clear 0: No operation 1: Clear the FERRSR.CTRL flag	W
19	CLP0	LP0 Contention Error Interrupt Flag Clear 0: No operation 1: Clear the FERRSR.CLP0 flag	W
20	CLP1	LP1 Contention Error Interrupt Flag Clear 0: No operation 1: Clear the FERRSR.CLP1 flag	W
31:21	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

HTXTO bit (HS TX Timeout Interrupt Flag Clear)

Set the HTXTO bit to 1 to clear the FERRSR.HTXTO flag.

LRXHTO bit (LP-RX Host Processor Timeout Interrupt Flag Clear)

Set the LRXHTO bit to 1 to clear the FERRSR.LRXHTO flag.

TATO bit (Turnaround Acknowledge Timeout Interrupt Flag Clear)

Set the TATO bit to 1 to clear the FERRSR.TATO flag.

ESCENT bit (Escape mode Entry Error Interrupt Flag Clear)

Set the ESCENT bit to 1 to clear the FERRSR.ESCENT flag.

SYNCESC bit (LPDT Sync Error Interrupt Flag Clear)

Set the SYNCESC bit to 1 to clear the FERRSR.SYNCESC flag.

CTRL bit (Control Error Interrupt Flag Clear)

Set the CTRL bit to 1 to clear the FERRSR.CTRL flag.

CLP0 bit (LP0 Contention Error Interrupt Flag Clear)

Set the CLP0 bit to 1 to clear the FERRSR.CLP0 flag.

CLP1 bit (LP1 Contention Error Interrupt Flag Clear)

Set the CLP1 bit to 1 to clear the FERRSR.CLP1 flag.

58.2.37 FERRIER : Fatal Error Interrupt Enable Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x308

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	CLP1	CLP0	CTRL	SYNCESC	ESCENT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	TATO	LRXHTO	HTXTO
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	HTXTO	HS TX Timeout Interrupt Enable 0: Disable 1: Enable	R/W
1	LRXHTO	LP-RX Host Processor Timeout Interrupt Enable 0: Disable 1: Enable	R/W
2	TATO	Turnaround Acknowledge Timeout Interrupt Enable 0: Disable 1: Enable	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
16	ESCENT	Escape mode Entry Error Interrupt Enable 0: Disable 1: Enable	R/W
17	SYNCESC	LPDT Sync Error Interrupt Enable 0: Disable 1: Enable	R/W
18	CTRL	Control Error Interrupt Enable 0: Disable 1: Enable	R/W
19	CLP0	LP0 Contention Error Interrupt Enable 0: Disable 1: Enable	R/W
20	CLP1	LP1 Contention Error Interrupt Enable 0: Disable 1: Enable	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

HTXTO bit (HS TX Timeout Interrupt Enable)

The HTXTO bit controls the HS TX Timeout (HTX_TO) Interrupt permission. When a HS TX Timeout Interrupt occurs, the FERRSR.HTXTO flag is set to 1.

To enable the HS TX Timeout Interrupt, set the HTXTO bit to 1.

LRXHTO bit (LP-RX Host Processor Timeout Interrupt Enable)

The LRXHTO bit controls the LP-RX Host Processor Timeout (LRX-H_TO) Interrupt permission. When a LP-RX Host Processor Timeout Interrupt occurs, the FERRSR.LRXHTO flag is set to 1.

To enable the LP-RX Host Processor Timeout Interrupt, set the LRXHTO bit to 1.

TATO bit (Turnaround Acknowledge Timeout Interrupt Enable)

The TATO bit controls the Turnaround Acknowledge Timeout (TA_TO) Interrupt permission. When a Turnaround Acknowledge Timeout Interrupt occurs, the FERRSR.TATO flag is set to 1.

To enable the Turnaround Acknowledge Timeout Interrupt, set the TATO bit to 1.

ESCENT bit (Escape mode Entry Error Interrupt Enable)

The ESCENT bit controls the Escape mode Entry Error (ErrEsc) Interrupt permission. When an Escape mode Entry Error Interrupt occurs, the FERRSR.ESCENT flag is set to 1.

To enable the Escape mode Entry Error Interrupt, set the ESCENT bit to 1.

SYNCESC bit (LPDT Sync Error Interrupt Enable)

The SYNCESC bit controls the Low-Power Data Transmission Synchronization Error (ErrSyncEsc) Interrupt permission. When a Low-Power Data Transmission Synchronization Error Interrupt occurs, the FERRSR.SYNCESC flag is set to 1.

To enable the Low-Power Data Transmission Synchronization Error Interrupt, set the SYNCESC bit to 1.

CTRL bit (Control Error Interrupt Enable)

The CTRL bit controls the Control Error (ErrControl) Interrupt permission. When a Control Error Interrupt occurs, the FERRSR.CTRL flag is set to 1.

To enable the Control Error Interrupt, set the CTRL bit to 1.

CLP0 bit (LP0 Contention Error Interrupt Enable)

The CLP0 bit controls the LP0 Contention Error (ErrContentionLP0) Interrupt permission. When an LP0 Contention Error Interrupt occurs, the FERRSR.CLP0 flag is set to 1.

To enable the LP0 Contention Error Interrupt, set the CLP0 bit to 1.

CLP1 bit (LP1 Contention Error Interrupt Enable)

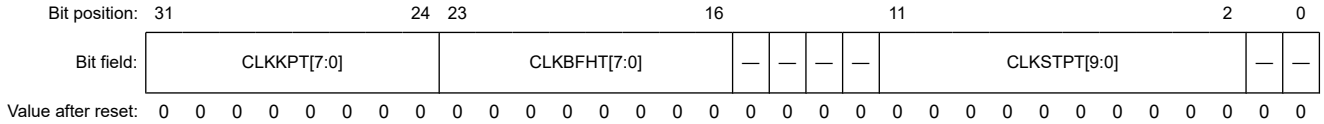
The CLP1 bit controls the LP1 Contention Error (ErrContentionLP1) Interrupt permission. When an LP1 Contention Error Interrupt occurs, the FERRSR.CLP1 flag is set to 1.

To enable the LP1 Contention Error Interrupt, set the CLP1 bit to 1.

58.2.38 CLSTPTSETR : Clock Lane Stop Time Set Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x314



Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
11:2	CLKSTPT[9:0]	Clock Stop Time Set the CLKSTPT period shown in Figure 58.1.*1	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
23:16	CLKBFHT[7:0]	Clock Beforehand Time Set the CLKBFHT period shown in Figure 58.1.*2*3	R/W
31:24	CLKKPT[7:0]	Clock Keep Time Set the CLKKPT period shown in Figure 58.1.*2*3	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Set this bit to a value greater than or equal to the value of the calculated period rounded up plus 3.
Note 2. Set the appropriate value (non-zero) before setting the HSCLKSETR.HSCLST bit to 1.
Note 3. Set these bits to a value greater than the desired time.

This register can only be set to change during the initialization at initial startup or Software Reset process (section 58.3.2. Software Reset).

CLKSTPT[9:0] bits (Clock Stop Time)

The CLKSTPT[9:0] bits define the minimum period of time between the transition to LP mode and the return to HS mode on the Clock Lane.

This setting is used to stop the Clock Lane when no HS transfer is taking place in the non-continuous clock mode (HSCLKSETR.HSCLMD = 0).

If there is an interval of more than the period specified by the CLKSTPT[9:0] bits between HS transfers on the Clock Lane, an LP mode period is inserted on the Clock Lane.

When the HSCLKSETR.HSCLMD bit is set to 1 (Continuous clock mode), the setting of the CLKSTPT[9:0] bits is meaningless.

This period corresponds to the time specified in the MIPI D-PHY Specification ($T_{HS-TRAIL}$ + $T_{CLK-POST}$ + $T_{CLK-TRAIL}$

+ $T_{HS-EXIT}$ + T_{LPX} × 2 + $T_{LPX(For\ Clock\ Lane\ HS\ Entry)}$ + $T_{CLK-PREPARE}$ + $T_{CLK-ZERO}$ + $T_{CLK-PRE}$ + $T_{LPX(For\ Data\ Lane\ HS\ Entry)}$ + $T_{HS-PREPARE}$ + $T_{HS-ZERO}$ + $T_{HS-SYNC}$). See Figure 58.1.

The period is derived from the following formula:

CLKSTPT period [µs] = CLKSTPT[9:0] × (period of HS serial UI) [µs] × 32

e.g.)

In the case of CLKSTPT[9:0] = 0x3E8, HS rate = 720 Mbps

CLKSTPT period [µs] = 1000 × (1 / 720 Mbps) × 32 = 44.4 [µs]

For information about each parameter, refer to the D-PHY specification and Figure 3-1. Of these parameters, it is necessary to set the time set in CLKKPT for ($T_{HS-TRAIL}$+ $T_{CLK-POST}$) and that set in CLKBFHT for ($T_{LPX\ (For\ Clock\ Lane\ HS\ Entry)}$ + $T_{CLK-PREPARE}$ + $T_{CLK-ZERO}$ + $T_{CLK-PRE}$). Set the rounded-up value plus at least 3. Setting this field to 0 is prohibited.

$T_{LPX(For\ Clock\ Lane\ HS\ Entry)}$ means the period of LP-01 in the HS entry sequence of Clock Lane. $T_{LPX(For\ Data\ Lane\ HS\ Entry)}$ means the period of LP-01 in the HS entry sequence of Data Lane. If these periods are not the same as the standard T_{LPX} value of the host device (in the range of 67% to 150% of the T_{LPX} value of the peripheral device), use the actual periods for this calculation.

CLKBFHT[7:0] bits (Clock Beforehand Time)

The CLKBFHT[7:0] bits specify the period during which the Clock Lane returns to HS mode prior to the start of HS packet transfer.

If HS transfers are scheduled in LP mode, the Clock Lane will start HS transfer in advance according to this setting. See Figure 58.1.

The period is derived from the following formula:

CLKBFHT period [µs] = CLKBFHT[7:0] × (period of HS serial UI) [µs] × 32

e.g.)

In the case of CLKBFHT[7:0] = 0x64, HS rate = 720 Mbps

CLKBFHT period [µs] = 100 × (1 / 720 Mbps) × 32 = 4.4 [µs]

This time corresponds to the D-PHY specification ($T_{LPX\ (For\ Clock\ Lane\ HS\ Entry)}$ + $T_{CLK-PREPARE}$+ $T_{CLK-ZERO}$ + $T_{CLK-PRE}$). The value must be greater than the one calculated. Set an appropriate value (other than 0) before HSCLKSETR.HSCLST is set to 1.

$T_{LPX\ (For\ Clock\ Lane\ HS\ Entry)}$ means the period of LP-01 in the HS entry sequence of Clock Lane. If this period is not same as normal T_{LPX} value of host device (in the range of 67% to 150% of the T_{LPX} value of peripheral device), use the actual period for this calculation.

CLKKPT[7:0] bits (Clock Keep Time)

The CLKKPT[7:0] bits specify the period that the Clock Lane keeps the HS mode active after the last HS packet transfer. See Figure 58.1.

The period is derived from the following formula:

CLKKPT period [µs] = CLKKPT[7:0] × (period of HS serial UI) [µs] × 32

e.g.)

In the case of CLKKPT[7:0] = 0x64, HS rate = 720 Mbps

CLKKPT period [µs] = 100 × (1 / 720 Mbps) × 32 = 4.4 [µs]

This time corresponds to the D-PHY specification's ($T_{HS-TRAIL}$ + $T_{CLK-POST}$).

The value must be greater than the one calculated. Set an appropriate value (other than 0) before HSCLKSETR.HSCLST is set to "1".

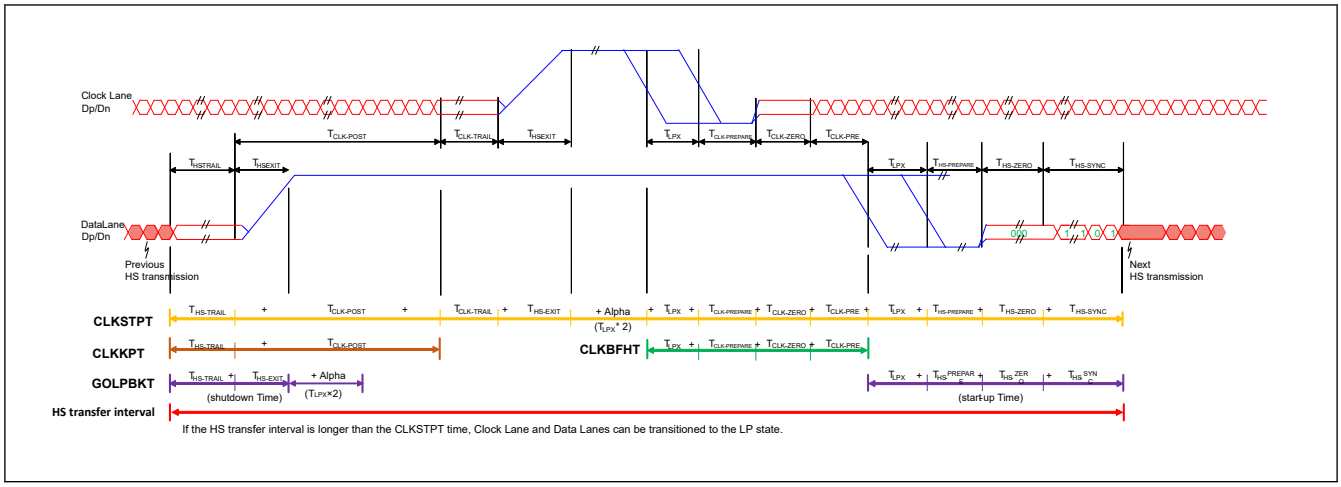


Figure 58.1 Minimum duration of LP mode on the Clock Lane

58.2.39 LPTRNSTSETR : LP Transition Time Set Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x318

Bit position:	31																					9								0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GOLPBKT[9:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	GOLPBKT[9:0]	Go LP and Back Time Set the GOLPBKT period shown in Figure 58.2. Set this bit to a value greater than or equal to the value of the calculated period rounded up plus 3.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register can only be set to change during the initialization at initial startup or Software Reset process (section 58.3.2. Software Reset).

GOLPBKT[9:0] bits (Go LP and Back Time)

The GOLPBKT[9:0] bits indicate the minimum period of time between the transition to LP mode and the return to HS mode on the Data Lane.

This setting is used for the ability of the DSI Host to automatically transition the Data Lane from HS mode to LP mode during the HSA, HBP, and HFP periods. See section 58.3.6.5. Selection of Blanking Packets or LP-11 for HSA, HBP, and HFP.

If there is an interval of more than the period specified by the GOLPBKT[9:0] bits between HS transfers, an LP mode period is inserted on the Data Lane.

This period corresponds to the time specified in the MIPI D-PHY Specification ($T_{HS\text{-}TRAIL} + T_{HS\text{-}EXIT} + T_{LPX} \times 2 + T_{LPX}$ (For Data Lane HS Entry) $+ T_{HS\text{-}PREPARE} + T_{HS\text{-}ZERO} + T_{HS\text{-}SYNC}$). See Figure 58.2.

The period is derived from the following formula:

GOLPBKT period [µs] = GOLPBKT[9:0] × (period of HS serial 8UI) [µs]*1

e.g.)

In the case of GOLPBKT[9:0] = 0x3E8, HS rate = 720 Mbps

GOLPBKT period [µs] = 1000 × (1 / 720 Mbps) × 8 = 11.1 [µs]

The unit of this setting is 8 UIs, so if the result of this formula contains a fraction of less than 8 UIs, round it up. Set the rounded-up value plus at least 3. $T_{LPX\ (For\ Data\ Lane\ HS\ Entry)}$ means the period of LP-01 in the HS entry sequence of Data Lane. If this period is not the same as the standard T_{LPX} value of the host device (in the range of 67% to 150% of the TLPX value of the peripheral device), use the actual period for this calculation.

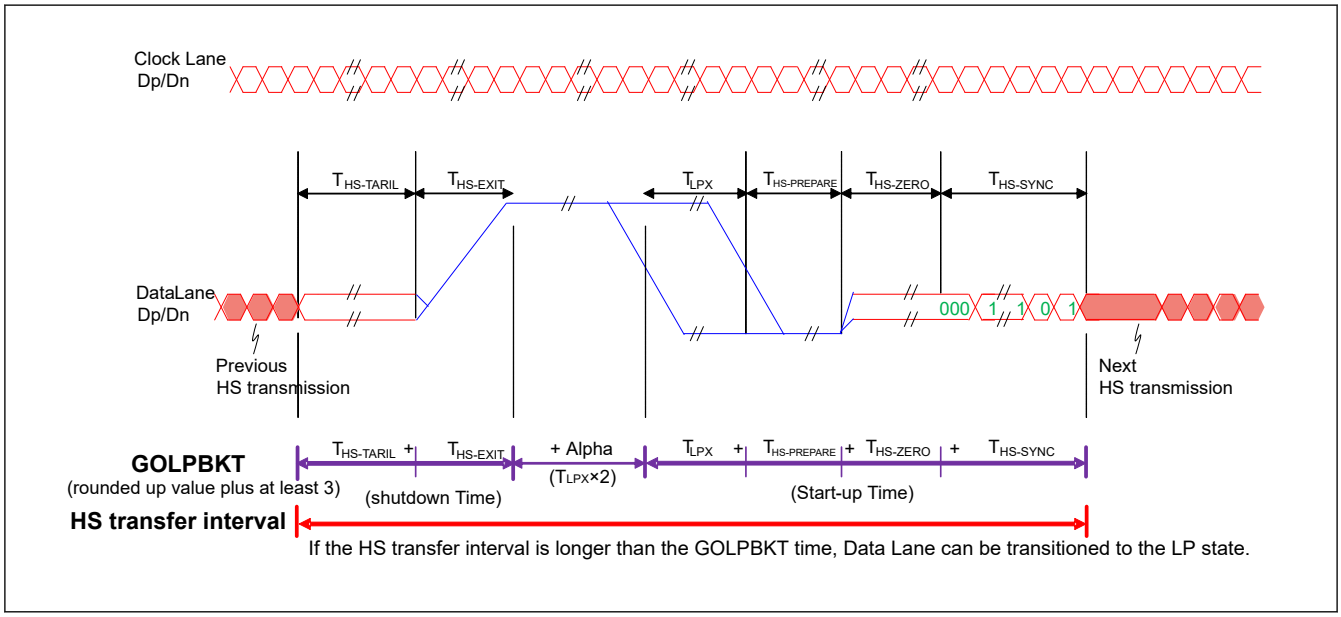


Figure 58.2 Minimum LP mode period of Data Lane

Note 1. The unit for this setting is 8UI, so if the result of this equation contains more than half of 8UI, round it up.

58.2.40 PLSR : Physical Lane Status Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x320

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	DLULPEXT	DLULPENT	CLHS2LP	CLLP2HS	CLULPEXT	CLULPENT	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DL0DIR	—	DL0TX2RX	DL0RX2TX	—	—	DL1STP	DL0STP	—	—	DL1UAN	DL0UAN	DL0RUE	DL0RLE	CLSTP	CLUAN
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	1

Bit	Symbol	Function	R/W
0	CLUAN	Clock Lane UlpsActiveNot Status Clock Lane state 0: In ULP state 1: Not in ULP state	R
1	CLSTP	Clock Lane Stop Status Clock Lane state 0: Not in Stop state 1: In Stop state	R
2	DL0RLE	Data Lane-0 RxLpdtEsc Status Data Lane-0 state 0: Not in Escape Low-Power Data Receive mode 1: In Escape Low-Power Data Receive mode	R

Bit	Symbol	Function	R/W
3	DL0RUE	Data Lane-0 RxUlpsEsc Status Data Lane-0 state 0: Not in Escape Ultra-Low Power (Receive) mode 1: In Escape Ultra-Low Power (Receive) mode	R
4	DL0UAN	Data Lane-0 UlpsActiveNot Status Data Lane-0 state 0: In ULP state 1: Not in ULP state	R
5	DL1UAN	Data Lane-1 UlpsActiveNot Status Data Lane-1 state 0: In ULP state 1: Not in ULP state	R
7:6	—	These bits are read as 1.	R
8	DL0STP	Data Lane-0 Stop Status Data Lane-0 state 0: Not in Stop state 1: In Stop state	R
9	DL1STP	Data Lane-1 Stop Status Data Lane-1 state 0: Not in Stop state 1: In Stop state	R
11:10	—	These bits are read as 0.	R
12	DL0RX2TX	Data Lane-0 RX to TX Transition Interrupt Flag Data Lane-0 direction transition from receiver to transmitter 0: Not detected 1: Detected	R
13	DL0TX2RX	Data Lane-0 TX to RX Transition Interrupt Flag Data Lane-0 direction transition from transmitter to receiver 0: Not detected 1: Detected	R
14	—	This bit is read as 0.	R
15	DL0DIR	Data Lane-0 Direction Data Lane-0 direction 0: Transmitter 1: Receiver	R
23:16	—	These bits are read as 0.	R
24	CLULPENT	Clock Lane ULPS Enter Interrupt Flag Clock Lane transition to ULPS 0: Not detected 1: Detected	R
25	CLULPEXT	Clock Lane ULPS Exit Interrupt Flag Clock Lane transition from ULPS 0: Not detected 1: Detected	R
26	CLLP2HS	Clock Lane LP to HS Transition Interrupt Flag Clock Lane mode transition from LP to HS 0: Not detected 1: Detected	R
27	CLHS2LP	Clock Lane HS to LP Transition Interrupt Flag Clock Lane mode transition from HS to LP 0: Not detected 1: Detected	R
28	DLULPENT	Data Lane ULPS Enter Interrupt Flag Data Lane transition to ULPS 0: Not detected 1: Detected	R

Bit	Symbol	Function	R/W
29	DLULPEXT	Data Lane ULPS Exit Interrupt Flag Data Lane transition from ULPS 0: Not detected 1: Detected	R
31:30	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

CLUAN bit (Clock Lane UlpsActiveNot Status)

The CLUAN bit indicates that the D-PHY Clock Lane is not in ULP State.*1

CLSTP bit (Clock Lane Stop Status)

The CLSTP bit indicates that the D-PHY Clock Lane module is currently in Stop state. *1

The CLSTP bit is used to indirectly determine if the PHY line levels are in the LP-11 state.

DL0RLE bit (Data Lane-0 RxLpdtEsc Status)

The DL0RLE bit indicates that the D-PHY Data Lane-0 module is in Low-Power data receive mode.*1

DL0RUE bit (Data Lane-0 RxUlpsEsc Status)

The DL0RUE bit indicates that the D-PHY Data Lane-0 module has entered the Ultra-Low Power State.*1

DL0UAN bit (Data Lane-0 UlpsActiveNot Status)

The DL0UAN bit indicates that the D-PHY Data Lane-0 is not in ULP State.*1

DL1UAN bit (Data Lane-1 UlpsActiveNot Status)

The DL1UAN bit indicates that the D-PHY Data Lane-1 is not in ULP State.*1

DL0STP bit (Data Lane-0 Stop Status)

The DL0STP bit indicates that the D-PHY Data Lane-0 module is currently in Stop state.*1

The DL0STP bit is used to indirectly determine if the PHY line levels are in the LP-11 state.

DL1STP bit (Data Lane-1 Stop Status)

The DL1STP bit indicates that the D-PHY Data Lane-1 module is currently in Stop state.*1

The DL1STP bit is used to indirectly determine if the PHY line levels are in the LP-11 state.

DL0RX2TX flag (Data Lane-0 RX to TX Transition Interrupt Flag)

The DL0RX2TX flag indicates that the D-PHY Data Lane-0 has transitioned from the receiver to the transmitter.

The DL0RX2TX flag is not cleared automatically, so set the PLSCR.DL0RX2TX bit to 1 to clear the DL0RX2TX flag.

DL0TX2RX flag (Data Lane-0 TX to RX Transition Interrupt Flag)

The DL0TX2RX flag indicates that the D-PHY Data Lane-0 has transitioned from the transmitter to the receiver.

The DL0TX2RX flag is not cleared automatically, so set the PLSCR.DL0TX2RX bit to 1 to clear the DL0TX2RX flag.

DL0DIR bit (Data Lane-0 Direction)

The DL0DIR bit indicates the status of the lane direction, whether the D-PHY Data Lane-0 is currently a transmitter or a receiver.*1

CLULPENT flag (Clock Lane ULPS Enter Interrupt Flag)

The CLULPENT flag indicates that the Clock Lane has entered ULPS.

The CLULPENT flag is not cleared automatically, so set the PLSCR.CLULPENT bit to 1 to clear the CLULPENT flag.

CLULPEXT flag (Clock Lane ULPS Exit Interrupt Flag)

The CLULPEXT flag indicates that the Clock Lane has exited from ULPS.

The CLULPEXT flag is not cleared automatically, so set the PLSCR.CLULPEXT bit to 1 to clear the CLULPEXT flag.

CLLP2HS flag (Clock Lane LP to HS Transition Interrupt Flag)

The CLLP2HS flag indicates that the Clock Lane has transitioned from LP mode to HS mode.

Ignore the CLLP2HS flag when the HSCLKSETR.HSCLMD bit is set to 0 (Non-continuous clock mode).

The CLLP2HS flag is not cleared automatically, so set the PLSCR.CLLP2HS bit to 1 to clear the CLLP2HS flag.

CLHS2LP flag (Clock Lane HS to LP Transition Interrupt Flag)

The CLHS2LP flag indicates that the Clock Lane has transitioned from HS mode to LP mode.

Ignore the CLHS2LP flag when the HSCLKSETR.HSCLMD bit is set to 0 (Non-continuous clock mode).

The CLHS2LP flag is not cleared automatically, so set the PLSCR.CLHS2LP bit to 1 to clear the CLHS2LP flag.

DLULPENT flag (Data Lane ULPS Enter Interrupt Flag)

The DLULPENT flag indicates that the Data Lanes specified by the TXSETR.NUMLANE[1:0] bits have transitioned from Stop state to ULPS by ULPS Enter.

The DLULPENT flag is not cleared automatically, so set the PLSCR.DLULPENT bit to 1 to clear the DLULPENT flag.

DLULPEXT flag (Data Lane ULPS Exit Interrupt Flag)

The DLULPEXT flag indicates that the Data Lanes specified by the TXSETR.NUMLANE[1:0] bits have transitioned from ULPS to Stop state by ULPS Exit.

The DLULPEXT flag is not cleared automatically, so set the PLSCR.DLULPEXT bit to 1 to clear the DLULPEXT flag.

Note 1. These bits are set after synchronization with LPCLK. Therefore, it takes time for the status to be refrected.

58.2.41 PLSCR : Physical Lane Status Clear Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x324

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	DLULPEXT	DLULPENT	CLHS2LP	CLLP2HS	CLULPEXT	CLULPENT	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	DL0TX2RX	DL0RX2TX	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	—	These bits are read as 0. The write value should be 0.	W
12	DL0RX2TX	Data Lane-0 RX to TX Transition Interrupt Flag Clear 0: No operation 1: Clear the PLSR.DL0RX2TX flag	W
13	DL0TX2RX	Data Lane-0 TX to RX Transition Interrupt Flag Clear 0: No operation 1: Clear the PLSR.DL0TX2RX flag	W
23:14	—	These bits are read as 0. The write value should be 0.	W
24	CLULPENT	Clock Lane ULPS Enter Interrupt Flag Clear 0: No operation 1: Clear the PLSR.CLULPENT flag	W
25	CLULPEXT	Clock Lane ULPS Exit Interrupt Flag Clear 0: No operation 1: Clear the PLSR.CLULPEXT flag	W

Bit	Symbol	Function	R/W
26	CLLP2HS	Clock Lane LP to HS Transition Interrupt Flag Clear 0: No operation 1: Clear the PLSR.CLLP2HS flag	W
27	CLHS2LP	Clock Lane HS to LP Transition Interrupt Flag Clear 0: No operation 1: Clear the PLSR.CLHS2LP flag	W
28	DLULPENT	Data Lane ULPS Enter Interrupt Flag Clear 0: No operation 1: Clear the PLSR.DLULPENT flag	W
29	DLULPEXT	Data Lane ULPS Exit Interrupt Flag Clear 0: No operation 1: Clear the PLSR.DLULPEXT flag	W
31:30	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

DL0RX2TX bit (Data Lane-0 RX to TX Transition Interrupt Flag Clear)

Set the DL0RX2TX bit to 1 to clear the PLSR.DL0RX2TX flag

DL0TX2RX bit (Data Lane-0 TX to RX Transition Interrupt Flag Clear)

Set the DL0TX2RX bit to 1 to clear the PLSR.DL0TX2RX flag.

CLULPENT bit (Clock Lane ULPS Enter Interrupt Flag Clear)

Set the CLULPENT bit to 1 to clear the PLSR.CLULPENT flag.

CLULPEXT bit (Clock Lane ULPS Exit Interrupt Flag Clear)

Set the CLULPEXT bit to 1 to clear the PLSR.CLULPEXT flag.

CLLP2HS bit (Clock Lane LP to HS Transition Interrupt Flag Clear)

Set the CLLP2HS bit to 1 to clear the PLSR.CLLP2HS flag.

CLHS2LP bit (Clock Lane HS to LP Transition Interrupt Flag Clear)

Set the CLHS2LP bit to 1 to clear the PLSR.CLHS2LP flag.

DLULPENT bit (Data Lane ULPS Enter Interrupt Flag Clear)

Set the DLULPENT bit to 1 to clear the PLSR.DLULPENT flag.

DLULPEXT bit (Data Lane ULPS Exit Interrupt Flag Clear)

Set the DLULPEXT bit to 1 to clear the PLSR.DLULPEXT flag.

58.2.42 PLIER : Physical Lane Interrupt Enable Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x328

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	DLULPEXT	DLULPENT	CLHS2LP	CLLP2HS	CLULPEXT	CLULPENT	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	DL0TX2RX	DL0RX2TX	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	—	These bits are read as 0. The write value should be 0.	R/W
12	DL0RX2TX	Data Lane-0 RX to TX Transition Interrupt Enable 0: Disable 1: Enable	R/W
13	DL0TX2RX	Data Lane-0 TX to RX Transition Interrupt Enable 0: Disable 1: Enable	R/W
23:14	—	These bits are read as 0. The write value should be 0.	R/W
24	CLULPENT	Clock Lane ULPS Enter Interrupt Enable 0: Disable 1: Enable	R/W
25	CLULPEXT	Clock Lane ULPS Exit Interrupt Enable 0: Disable 1: Enable	R/W
26	CLLP2HS	Clock Lane LP to HS Transition Interrupt Enable 0: Disable 1: Enable	R/W
27	CLHS2LP	Clock Lane HS to LP Transition Interrupt Enable 0: Disable 1: Enable	R/W
28	DLULPENT	Data Lane ULPS Enter Interrupt Enable 0: Disable 1: Enable	R/W
29	DLULPEXT	Data Lane ULPS Exit Interrupt Enable 0: Disable 1: Enable	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

DL0RX2TX bit (Data Lane-0 RX to TX Transition Interrupt Enable)

The DL0RX2TX bit controls the Data Lane-0 RX to TX Transition Interrupt permission.

When a Data Lane-0 RX to TX Transition Interrupt occurs, the PLSR.DL0RX2TX flag is set to 1.

To enable the Data Lane-0 RX to TX Transition Interrupt, set the DL0RX2TX bit to 1.

DL0TX2RX bit (Data Lane-0 TX to RX Transition Interrupt Enable)

The DL0TX2RX bit controls the Data Lane-0 TX to RX Transition Interrupt permission.

When a Data Lane-0 TX to RX Transition Interrupt occurs, the PLSR.DL0TX2RX flag is set to 1.

To enable the Data Lane-0 TX to RX Transition Interrupt, set the DL0TX2RX bit to 1.

CLULPENT bit (Clock Lane ULPS Enter Interrupt Enable)

The CLULPENT bit controls the Clock Lane ULPS Enter Interrupt permission.

When a Clock Lane ULPS Enter Interrupt occurs, the PLSR.CLULPENT flag is set to 1.

To enable the Clock Lane ULPS Enter Interrupt, set the CLULPENT bit to 1.

CLULPEXT bit (Clock Lane ULPS Exit Interrupt Enable)

The CLULPEXT bit controls the Clock Lane ULPS Exit Interrupt permission.

When a Clock Lane ULPS Exit Interrupt occurs, the PLSR.CLULPEXT flag is set to 1.

To enable the Clock Lane ULPS Exit Interrupt, set the CLULPEXT bit to 1.

CLLP2HS bit (Clock Lane LP to HS Transition Interrupt Enable)

The CLLP2HS bit controls the Clock Lane LP to HS Transition Interrupt permission.

When a Clock Lane LP to HS Transition Interrupt occurs, the PLSR.CLLP2HS flag is set to 1.

To enable the Clock Lane LP to HS Transition Interrupt, set the CLLP2HS bit to 1.

CLHS2LP bit (Clock Lane HS to LP Transition Interrupt Enable)

The CLHS2LP bit controls the Clock Lane HS to LP Transition Interrupt permission.

When a Clock Lane HS to LP Transition Interrupt occurs, the PLSR.CLHS2LP flag is set to 1.

To enable the Clock Lane HS to LP Transition Interrupt, set the CLHS2LP bit to 1.

DLULPENT bit (Data Lane ULPS Enter Interrupt Enable)

The DLULPENT bit controls the Data Lane ULPS Enter Interrupt permission.

When a Data Lane ULPS Enter Interrupt occurs, the PLSR.DLULPENT flag is set to 1.

To enable the Data Lane ULPS Enter Interrupt, set the DLULPENT bit to 1

DLULPEXT bit (Data Lane ULPS Exit Interrupt Enable)

The DLULPEXT bit controls the Data Lane ULPS Exit Interrupt permission.

When a Data Lane ULPS Exit Interrupt occurs, the PLSR.DLULPEXT flag is set to 1.

To enable the Data Lane ULPS Exit Interrupt, set the DLULPEXT bit to 1.

58.2.43 VMSET0R : Video Mode Set 0 Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x400

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	HFPNOLP	HBPNOLP	HSANOLP	—	—	—	—	—	—	VSTOP	VSTART
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VSTART	Video Mode Operation Start 0: No operation 1: Start Video mode operation	W
1	VSTOP	Video Mode Operation Stop 0: No operation 1: Stop Video mode operation	W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	HSANOLP	HSA period No LP 0: Not suppressing the transition to LP during HSA period 1: Suppress the transition to LP during HSA period and keep HS	R/W
9	HBPNOLP	HBP period No LP 0: Not suppressing the transition to LP during HBP period 1: Suppress the transition to LP during HBP period and keep HS	R/W
10	HFPNOLP	HFP period No LP 0: Not suppressing the transition to LP during HFP period 1: Suppress the transition to LP during HFP period and keep HS	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

VSTART bit (Video Mode Operation Start)

The VSTART bit is start trigger for Video Mode operation.

Setting 1 to both the VSTART bit and the VSTOP bit is prohibited.

When the VMSR.RUNNING bit is set to 1, setting the VSTART bit to 1 is prohibited.

VSTOP bit (Video Mode Operation Stop)

The VSTOP bit is stop trigger for Video Mode operation.

The video mode operation is stopped at the end of video-frame. After the video mode operation is stopped, stop the GLCDC.

Setting 1 to both the VSTOP bit and the VSTART bit is prohibited.

When the VMSR.RUNNING bit is cleared to 0, setting the VSTOP bit to 1 is prohibited.

HSANOLP bit (HSA period No LP)

The HSANOLP bit controls whether to transition to LP during the HSA period.

When the HSANOLP bit is set to 1, the link does not transition to LP during HSA period and keep HS mode.

HS mode is kept by sending blanking packet.

HBPNOLP bit (HBP period No LP)

The HBPNOLP bit controls whether to transition to LP during the HBP period.

When the HBPNOLP bit is set to 1, the link does not transition to LP during HBP period and keep HS mode.

HS mode is kept by sending blanking packet.

HFPNOLP bit (HFP period No LP)

The HFPNOLP bit controls whether to transition to LP during the HFP period.

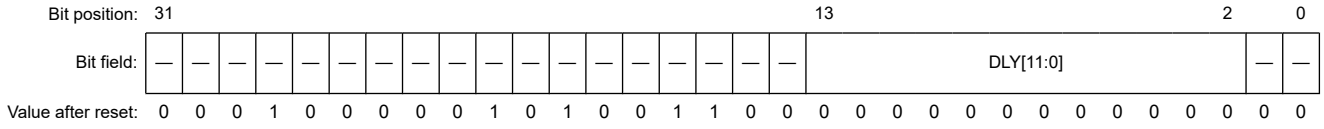
When the HFPNOLP bit is set to 1, the link does not transition to LP during HFP period and keep HS mode.

HS mode is kept by sending blanking packet.

58.2.44 VMSET1R : Video Mode Set 1 Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x404



Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
13:2	DLY[11:0]	Delay Value Set the delay value inside DSI Host until the transfer begins.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
17:16	—	These bits are read as 1. The write value should be 1.	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
20	—	This bit is read as 1. The write value should be 1.	R/W
21	—	This bit is read as 0. The write value should be 0.	R/W
22	—	This bit is read as 1. The write value should be 1.	R/W
23	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
29:24	—	The read values are undefined. The write value should be 0.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

DLY[11:0] bits (Delay Value)

The DLY[11:0] bits are for the control of the delay time inside DSI Host to avoid the underflow or the overflow of the video stream from GLCDC.

The delay time is derived from the following formula:

Delay time [µs] = DLY[11:0] × (period of HS serial UI) [µs] × 32

e.g.)

In the case of DLY[11:0] = 0x3E8, HS line rate = 720 Mbps

Delay time [µs] = 1000 × (1 / 720 Mbps) × 32 = 44.4 [µs]

For more details, see section 58.3.6.7. Delay Adjustment for Video Mode Operation.

It is prohibited to set the VMSET1R.DLY[11:0] bits to 0x000.

If the VMSET0R.HFPNOLP bit is set to 1, set the DLY[11:0] bits to more than HFP period.

If the VMSET0R.HBPNOLP bit is set to 1, set the DLY[11:0] bits to more than HBP period.

If the VMSET0R.HSANOLP bit is set to 1, set the DLY[11:0] bits to more than HSA period.

58.2.45 VMSR : Video Mode Status Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x410

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	VBUFOVF	VBUFUDF	—	TIMERR	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	VIRDY	RUNNING	STOP	START
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	START	Video Mode Operation Start Interrupt Flag 0: Video mode operation has not started 1: Video mode operation has started	R
1	STOP	Video Mode Operation Stop Interrupt Flag 0: Video mode operation has not stopped 1: Video mode operation has stopped	R
2	RUNNING	Video Mode Operation Running Status 0: Video mode operation is stopped 1: Video mode operation is running	R
3	VIRDY	Video Mode Operation Ready Interrupt Flag 0: Video mode operation is not ready 1: Video mode operation is ready	R
19:4	—	These bits are read as 0.	R
20	TIMERR	Timing Error Interrupt Flag 0: Timing error is not occurred 1: Timing error is occurred during the video mode operation	R

Bit	Symbol	Function	R/W
21	—	This bit is read as 0.	R
22	VBUFUDF	Video Buffer Underflow Error Interrupt Flag 0: Data underflow is not occurred 1: Data underflow is occurred in the video buffer	R
23	VBUFOVF	Video Buffer Overflow Error Interrupt Flag 0: Data overflow is not occurred 1: Data overflow is occurred in the video buffer	R
31:24	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

START flag (Video Mode Operation Start Interrupt Flag)

The START flag indicates that the video mode operation has started.

The START flag is not cleared automatically, so set the VMSCR.START bit to 1 to clear the START flag.

STOP flag (Video Mode Operation Stop Interrupt Flag)

The STOP flag indicates that the video mode operation has stopped.

The STOP flag is not cleared automatically, so set the VMSCR.STOP bit to 1 to clear the STOP flag.

RUNNING bit (Video Mode Operation Running Status)

The RUNNING bit indicates the operating state of the video mode.

Changing the settings of VMSET0R, VMSET1R, VMPPSETR, VMVSSETR, VMVPSETR, VMHSSETR and VMHPSETR is prohibited while the RUNNING bit is 1.

VIRDY flag (Video Mode Operation Ready Interrupt Flag)

The VIRDY flag indicates that the video mode operation is ready to start working.

The VIRDY flag is not cleared automatically, so set the VMSCR.VIRDY bit to 1 to clear the VIRDY flag.

TIMERR flag (Timing Error Interrupt Flag)

The TIMERR flag indicates that the timing error is occurred during the video mode operation.

The TIMERR flag is not cleared automatically, so set the VMSCR.TIMERR bit to 1 to clear the TIMERR flag.

VBUFUDF flag (Video Buffer Underflow Error Interrupt Flag)

The VBUFUDF flag indicates that the data underflow is occurred in the video buffer.

The VBUFUDF flag is not cleared automatically, so set the VMSCR.VBUFUDF bit to 1 to clear the VBUFUDF flag.

VBUFOVF flag (Video Buffer Overflow Error Interrupt Flag)

The VBUFOVF flag indicates that the data overflow is occurred in the video buffer.

The VBUFOVF flag is not cleared automatically, so set the VMSCR.VBUFOVF bit to 1 to clear the VBUFOVF flag.

58.2.46 VMSCR : Video Mode Status Clear Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x414

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	VBUFOVF	VBUFUDF	—	TIMERR	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	VIRDY	—	STOP	START
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	START	Video Mode Operation Start Interrupt Flag Clear 0: No operation 1: Clear the VMSR.START flag	W
1	STOP	Video Mode Operation Stop Interrupt Flag Clear 0: No operation 1: Clear the VMSR.STOP flag	W
2	—	This bit is read as 0. The write value should be 0.	W
3	VIRDY	Video Mode Operation Ready Interrupt Flag Clear 0: No operation 1: Clear the VMSR.VIRDY flag	W
19:4	—	These bits are read as 0. The write value should be 0.	W
20	TIMERR	Timing Error Interrupt Flag Clear 0: No operation 1: Clear the VMSR.TIMERR flag	W
21	—	This bit is read as 0. The write value should be 0.	W
22	VBUFUDF	Video Buffer Underflow Error Interrupt Flag Clear 0: No operation 1: Clear the VMSR.VBUFUDF flag	W
23	VBUFOVF	Video Buffer Overflow Error Interrupt Flag Clear 0: No operation 1: Clear the VMSR.VBUFOVF flag	W
31:24	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

START bit (Video Mode Operation Start Interrupt Flag Clear)

Set the START bit to 1 to clear the VMSR.START flag.

STOP bit (Video Mode Operation Stop Interrupt Flag Clear)

Set the STOP bit to 1 to clear the VMSR.STOP flag.

VIRDY bit (Video Mode Operation Ready Interrupt Flag Clear)

Set the VIRDY bit to 1 to clear the VMSR.VIRDY flag.

TIMERR bit (Timing Error Interrupt Flag Clear)

Set the TIMERR bit to 1 to clear the VMSR.TIMERR flag.

VBUFUDF bit (Video Buffer Underflow Error Interrupt Flag Clear)

Set the VBUFUDF bit to 1 to clear the VMSR.VBUFUDF flag.

VBUFOVF bit (Video Buffer Overflow Error Interrupt Flag Clear)

Set the VBUFOVF bit to 1 to clear the VMSR.VBUFOVF flag.

58.2.47 VMIER : Video Mode Interrupt Enable Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x418

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	VBUFOVF	VBUFUDF	—	TIMERR	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	VIRDY	—	STOP	START
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	START	Video Mode Operation Start Interrupt Enable 0: Disable 1: Enable	R/W
1	STOP	Video Mode Operation Stop Interrupt Enable 0: Disable 1: Enable	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	VIRDY	Video Mode Operation Ready Interrupt Enable 0: Disable 1: Enable	R/W
19:4	—	These bits are read as 0. The write value should be 0.	R/W
20	TIMERR	Timing Error Interrupt Enable 0: Disable 1: Enable	R/W
21	—	This bit is read as 0. The write value should be 0.	R/W
22	VBUFUDF	Video Buffer Underflow Error Interrupt Enable 0: Disable 1: Enable	R/W
23	VBUFOVF	Video Buffer Overflow Error Interrupt Enable 0: Disable 1: Enable	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

START bit (Video Mode Operation Start Interrupt Enable)

The START bit controls the Video Mode Operation Start Interrupt permission.

When a Video Mode Operation Start Interrupt occurs, the VMSR.START flag is set to 1.

To enable the Video Mode Operation Start Interrupt, set the START bit to 1.

STOP bit (Video Mode Operation Stop Interrupt Enable)

The STOP bit controls the Video Mode Operation Stop Interrupt permission.

When a Video Mode Operation Stop Interrupt occurs, the VMSR.STOP flag is set to 1.

To enable the Video Mode Operation Stop Interrupt, set the STOP bit to 1.

VIRDY bit (Video Mode Operation Ready Interrupt Enable)

The VIRDY bit controls the Video Mode Operation Ready Interrupt permission.

When a Video Mode Operation Ready Interrupt occurs, the VMSR.VIRDY flag is set to 1.

To enable the Video Mode Operation Ready Interrupt, set the VIRDY bit to 1.

TIMERR bit (Timing Error Interrupt Enable)

The TIMERR bit controls the Timing Error Interrupt permission.

When a Timing Error Interrupt occurs, the VMSR.TIMERR flag is set to 1.

To enable the Timing Error Interrupt, set the TIMERR bit to 1.

VBUFUDF bit (Video Buffer Underflow Error Interrupt Enable)

The VBUFUDF bit controls the Video Buffer Underflow Error Interrupt permission.

When a Video Buffer Underflow Error Interrupt occurs, the VMSR.VBUFUDF flag is set to 1.

To enable the Video Buffer Underflow Error Interrupt, set the VBUFUDF bit to 1.

VBUFOVF bit (Video Buffer Overflow Error Interrupt Enable)

The VBUFOVF bit controls the Video Buffer Overflow Error Interrupt permission.

When a Video Buffer Overflow Error Interrupt occurs, the VMSR.VBUFOVF flag is set to 1.

To enable the Video Buffer Overflow Error Interrupt, set the VBUFOVF bit to 1.

58.2.48 VMPPSETR : Video Mode Pixel Packet Set Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x420

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	VC[1:0]		DT[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TXESYNC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
14:0	—	These bits are read as 0. The write value should be 0.	R/W
15	TXESYNC	Transmit End of Sync Pulse 0: HSE and VSE are not transmitted 1: HSE and VSE are transmitted	R/W
21:16	DT[5:0]	Video Mode Data Type Set the data type for the pixel stream packet header 0x0E: Packed Pixel Stream, 16 bit RGB 0x1E: Packed Pixel Stream, 18 bit RGB 0x3E: Packed Pixel Stream, 24 bit RGB Others: Setting prohibited	R/W
23:22	VC[1:0]	Video Mode Virtual Channel Set the virtual channel identifier of 00b to 11b.	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

TXESYNC bit (Transmit End of Sync Pulse)

0 setting is used for *Burst Mode* sequence or *Non-Burst Mode with Sync Events* sequence.

1 setting is used for *Non-Burst Mode with Sync Pulse* sequence.

DT[5:0] bits (Video Mode Data Type)

Set the same format as the input format from GLCDC.

VC[1:0] bits (Video Mode Virtual Channel)

Set a number between 00b to 11b as the virtual channel identifier.

58.2.49 VMVSSETR : Video Mode Vertical Size Set Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x428

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	VACT[14:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	VSPOL	—	—	—	VSA[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	VSA[11:0]	VSA Lines Set the number of VSA (Vertical Sync Active) lines.	R/W
14:12	—	These bits are read as 0. The write value should be 0.	R/W
15	VSPOL	VSYNC Polarity 0: High Active 1: Low Active	R/W
30:16	VACT[14:0]	Vertical Active Lines Set the number of VACT (Vertical Active) lines.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

VSA[11:0] bits (VSA Lines)

The VSA[11:0] bits control the number of VSA (Vertical Sync Active) lines.

VSPOL bit (VSYNC Polarity)

The VSPOL bit controls the polarity of VSYNC signal.

VACT[14:0] bits (Vertical Active Lines)

The VACT[14:0] bits control the number of VACT (Vertical Active) lines.

58.2.50 VMVPSETR : Video Mode Vertical Porch Set Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x42C

Bit position:	31	28	16	12	0
Bit field:	— — —	VFP[12:0]	— — —	VBP[12:0]	
Value after reset:	0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
12:0	VBP[12:0]	VBP Lines Set the number of VBP (Vertical Back Porch) lines.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
28:16	VFP[12:0]	VFP Lines Set the number of VFP (Vertical Front Porch) lines.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

VBP[12:0] bits (VBP Lines)

The VBP[12:0] bits control the number of VBP (Vertical Back Porch) lines.

VFP[12:0] bits (VFP Lines)

The VFP[12:0] bits control the number of VFP (Vertical Front Porch) lines.

58.2.51 VMHSSETR : Video Mode Horizontal Size Set Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x430

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	HACT[14:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	HSPOL	—	—	—	HSA[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	HSA[11:0]	HSA Pixels Set the number of HSA (Horizontal Sync Active) pixels.	R/W
14:12	—	These bits are read as 0. The write value should be 0.	R/W
15	HSPOL	HSYNC Polarity 0: High Active 1: Low Active	R/W
30:16	HACT[14:0]	HACT Pixels Set the number of HACT (Horizontal Active) pixels.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

HSA[11:0] bits (HSA Pixels)

The HSA[11:0] bits control the number of HSA (Horizontal Sync Active) pixels.

HSPOL bit (HSYNC Polarity)

The HSPOL bit controls the polarity of HSYNC signal.

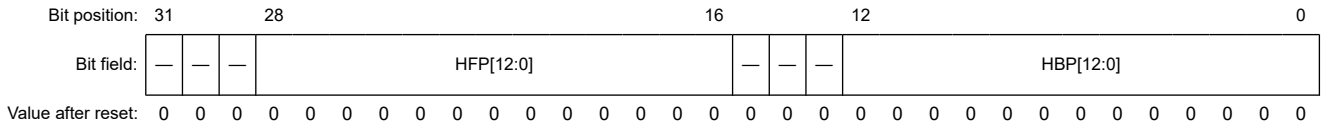
HACT[14:0] bits (HACT Pixels)

The HACT[14:0] bits control the number of HACT (Horizontal Active) pixels.

58.2.52 VMHPSETR : Video Mode Horizontal Porch Set Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x434



Bit	Symbol	Function	R/W
12:0	HBP[12:0]	HBP Pixels Set the number of HBP (Horizontal Back Porch) pixels.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
28:16	HFP[12:0]	HFP Pixels Set the number of HFP (Horizontal Front Porch) pixels.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

HBP[12:0] bits (HBP Pixels)

The HBP[12:0] bits control the number of HBP (Horizontal Back Porch) pixels.

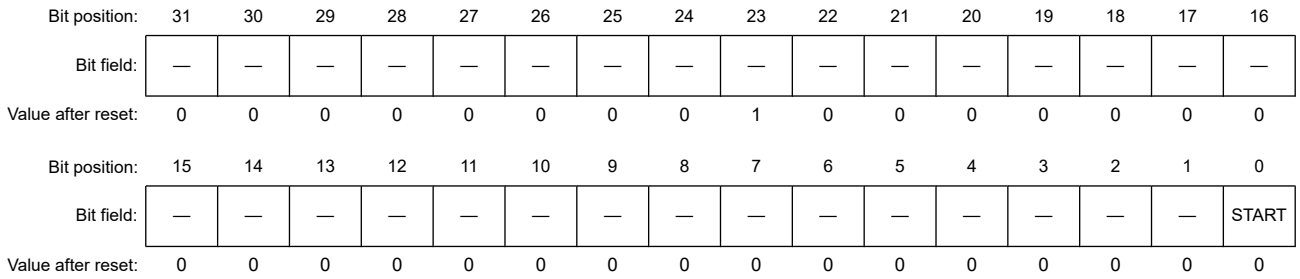
HFP[12:0] bits (HFP Pixels)

The HFP[12:0] bits control the number of HFP (Horizontal Front Porch) pixels.

58.2.53 SQCH0SET0R : Sequence Channel 0 Set 0 Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x5C0



Bit	Symbol	Function	R/W
0	START	Sequence Operation Start 0: No operation 1: Start the sequence operation	W
22:1	—	These bits are read as 0. The write value should be 0.	W
23	—	This bit is read as 1. The write value should be 1.	W
31:24	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

START bit (Sequence Operation Start)

The START bit is the trigger to start the sequence operation. The sequence operations are executed in sequence from Descriptor-0.

Setting the START bit to 1 when the SQCH0SR.RUNNING flag or the SQCH1SR.RUNNING flag is set to 1 is prohibited.

58.2.54 SQCH0SR : Sequence Channel 0 Status Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x5D0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	RXAKE	RXCORERR	RXPFAIL	RXFAIL	RXFERR	—	TXIBERR	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADESFIN	—	—	—	AACTFIN	—	RUNNING	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0.	R
2	RUNNING	Sequence Operation Running Status 0: Sequence operation is stopped 1: Sequence operation is running	R
3	—	This bit is read as 0.	R
4	AACTFIN	All Actions Finish Interrupt Flag 0: All actions of a descriptor have not finished 1: All actions of a descriptor finished operations	R
7:5	—	These bits are read as 0.	R
8	ADESFIN	All-Descriptors Finish Interrupt Flag 0: All descriptors have not finished 1: All descriptors finished operations	R
23:9	—	These bits are read as 0.	R
24	TXIBERR	Tx Internal Bus Error Interrupt Flag 0: No error 1: Internal AXI bus read error occurred	R
25	—	This bit is read as 0.	R
26	RXFERR	Receive Fatal Error Interrupt Flag 0: No error 1: Fatal timeout occurred during BTA	R
27	RXFAIL	Receive Fail Interrupt Flag 0: No error 1: Expected receive did not take place	R
28	RXPFAIL	Receive Packet Data Fail Interrupt Flag 0: No error 1: Received packet data is not stored correctly	R
29	RXCORERR	Receive Correctable Error Interrupt Flag 0: No error 1: Received packets have correctable error	R
30	RXAKE	Receive Acknowledge and Error Report Packet Interrupt Flag 0: No error 1: Acknowledge and Error Report Packet is received	R
31	—	This bit is read as 0.	R

Note: S-TYPE-3, P-TYPE-3

RUNNING bit (Sequence Operation Running Status)

The RUNNING bit indicates the state of the sequence operation.

Changing the settings of the descriptor is prohibited while the RUNNING bit is 1.

AACTFIN flag (All Actions Finish Interrupt Flag)

The AACTFIN flag is set if this descriptor's all actions are finished with SQCHnDSCmCR.FINACT =1.

The AACTFIN flag is not cleared automatically, so set the SQCH0SCR.AACTFIN bit to 1 to clear the AACTFIN flag.

ADESFIN flag (All-Descriptors Finish Interrupt Flag)

The ADESFIN flag is set when all descriptors finish operations.

When Descriptor-m whose the SQCHnDSCmAR.NXACT[1:0] bits are set to 00b finishes the operation, the ADESFIN flag is set to 1.

The ADESFIN flag is not cleared automatically, so set the SQCH0SCR.ADESFIN bit to 1 to clear the ADESFIN flag.

TXIBERR flag (Tx Internal Bus Error Interrupt Flag)

The TXIBERR flag is set to 1 when the internal AXI bus read is failed.

The TXIBERR flag is not cleared automatically, so set the SQCH0SCR.TXIBERR bit to 1 to clear the TXIBERR flag.

RXFERR flag (Receive Fatal Error Interrupt Flag)

The RXFERR flag is set to 1 when the fatal timeout occurred during BTA.

The FERRSR.TATO flag or the FERRSR.LRXHTO flag is also set to 1.

The RXFERR flag is not cleared automatically, so set the SQCH0SCR.RXFERR bit to 1 to clear the RXFERR flag.

RXFAIL flag (Receive Fail Interrupt Flag)

The RXFAIL flag is set to 1 when the expected receive did not done.

The RXSR.PRTOERR flag, the RXSR.ECCERRM flag, the RXSR.MLFERR flag, or the RXSR.NORESERR flag is also set to 1.

The RXFAIL flag is not cleared automatically, so set the SQCH0SCR.RXFAIL bit to 1 to clear the RXFAIL flag.

RXPFAIL flag (Receive Packet Data Fail Interrupt Flag)

The RXPFAIL flag is set to 1 when the received packet header is stored correctly, but the packet data is not stored correctly.

The RXSR.CRCERR flag, the RXSR.WCERR flag, the RXSR.UNEXERR flag, the RXSR.RSIZEERR flag,

the RXSR.RXOVFERR flag, or the RXSR.IBERR flag is also set to 1.

The RXPFAIL flag is not cleared automatically, so set the SQCH0SCR.RXPFAIL bit to 1 to clear the RXPFAIL flag.

RXCORERR flag (Receive Correctable Error Interrupt Flag)

The RXCORERR flag is set to 1 when the received packet has correctable errors.

The RXSR.ECCERRS flag is also set to 1.

The RXCORERR flag is not cleared automatically, so set the SQCH0SCR.RXCORERR bit to 1 to clear the RXCORERR flag.

RXAKE flag (Receive Acknowledge and Error Report Packet Interrupt Flag)

The RXAKE flag is set to 1 when the Acknowledge and Error Report Packet is received.

The RXSR.RXAKE flag is also set to 1.

The RXAKE flag is not cleared automatically, so set the SQCH0SCR.RXAKE bit to 1 to clear the RXAKE flag.

58.2.55 SQCH0SCR : Sequence Channel 0 Status Clear Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x5D4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	RXAKE	RXCORERR	RXPFAIL	RXFAIL	RXFERR	—	TXIBERR	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADESFIN	—	—	—	AACTFIN	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	W
4	AACTFIN	All Actions Finish Interrupt Flag Clear 0: No operation 1: Clear the SQCH0SR.AACTFIN flag	W
7:5	—	These bits are read as 0. The write value should be 0.	W
8	ADESFIN	All-Descriptors Finish Interrupt Flag Clear 0: No operation 1: Clear the SQCH0SR.ADESFIN flag	W
23:9	—	These bits are read as 0. The write value should be 0.	W
24	TXIBERR	Tx Internal Bus Error Interrupt Flag Clear 0: No operation 1: Clear the SQCH0SR.TXIBERR flag	W
25	—	This bit is read as 0. The write value should be 0.	W
26	RXFERR	Receive Fatal Error Interrupt Flag Clear 0: No operation 1: Clear the SQCH0SR.RXFERR flag	W
27	RXFAIL	Receive Fail Interrupt Flag Clear 0: No operation 1: Clear the SQCH0SR.RXFAIL flag	W
28	RXPFAIL	Receive Packet Data Fail Interrupt Flag Clear 0: No operation 1: Clear the SQCH0SR.RXPFAIL flag	W
29	RXCORERR	Receive Correctable Error Interrupt Flag Clear 0: No operation 1: Clear the SQCH0SR.RXCORERR flag	W
30	RXAKE	Receive Acknowledge and Error Report Packet Interrupt Flag Clear 0: No operation 1: Clear the SQCH0SR.RXAKE flag	W
31	—	This bit is read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

AACTFIN bit (All Actions Finish Interrupt Flag Clear)

Set the AACTFIN bit to 1 to clear the SQCH0SR.AACTFIN flag.

ADESFIN bit (All-Descriptors Finish Interrupt Flag Clear)

Set the ADESFIN bit to 1 to clear the SQCH0SR.ADESFIN flag.

TXIBERR bit (Tx Internal Bus Error Interrupt Flag Clear)

Set the TXIBERR bit to 1 to clear the SQCH0SR.TXIBERR flag.

RXFERR bit (Receive Fatal Error Interrupt Flag Clear)

Set the RXFERR bit to 1 to clear the SQCH0SR.RXFERR flag.

RXFAIL bit (Receive Fail Interrupt Flag Clear)

Set the RXFAIL bit to 1 to clear the SQCH0SR.RXFAIL flag.

RXPFAIL bit (Receive Packet Data Fail Interrupt Flag Clear)

Set the RXPFAIL bit to 1 to clear the SQCH0SR.RXPFAIL flag.

RXCORERR bit (Receive Correctable Error Interrupt Flag Clear)

Set the RXCORERR bit to 1 to clear the SQCH0SR.RXCORERR flag.

RXAKE bit (Receive Acknowledge and Error Report Packet Interrupt Flag Clear)

Set the RXAKE bit to 1 to clear the SQCH0SR.RXAKE flag.

58.2.56 SQCH0IER : Sequence Channel 0 Interrupt Enable Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x5D8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	RXAKE	RXCORERR	RXPFAIL	RXFAIL	RXFERR	—	TXIBERR	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADESFIN	—	—	—	AACTFIN	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	AACTFIN	All Actions Finish Interrupt Enable 0: Disable 1: Enable	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	ADESFIN	All-Descriptors Finish Interrupt Enable 0: Disable 1: Enable	R/W
23:9	—	These bits are read as 0. The write value should be 0.	R/W
24	TXIBERR	Tx Internal Bus Error Interrupt Enable 0: Disable 1: Enable	R/W
25	—	This bit is read as 0. The write value should be 0.	R/W
26	RXFERR	Receive Fatal Error Interrupt Enable 0: Disable 1: Enable	R/W
27	RXFAIL	Receive Fail Interrupt Enable 0: Disable 1: Enable	R/W
28	RXPFAIL	Receive Packet Data Fail Interrupt Enable 0: Disable 1: Enable	R/W

Bit	Symbol	Function	R/W
29	RXCORERR	Receive Correctable Error Interrupt Enable 0: Disable 1: Enable	R/W
30	RXAKE	Receive Acknowledge and Error Report Packet Interrupt Enable 0: Disable 1: Enable	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

AACTFIN bit (All Actions Finish Interrupt Enable)

The AACTFIN bit controls the All Actions Finish Interrupt permission.

When an All Actions Finish Interrupt occurs, the SQCH0SR.AACTFIN flag is set to 1.

To enable the All Actions Finish Interrupt, set the AACTFIN bit to 1.

ADESFIN bit (All-Descriptors Finish Interrupt Enable)

The ADESFIN bit controls the All-Descriptors Finish Interrupt permission.

When an All-Descriptors Finish Interrupt occurs, the SQCH0SR.ADESFIN flag is set to 1.

To enable the All-Descriptors Finish Interrupt, set the ADESFIN bit to 1.

TXIBERR bit (Tx Internal Bus Error Interrupt Enable)

The TXIBERR bit controls the Tx Internal Bus Error Interrupt permission.

When a Tx Internal Bus Error Interrupt occurs, the SQCH0SR.TXIBERR flag is set to 1.

To enable the Tx Internal Bus Error Interrupt, set the TXIBERR bit to 1.

RXFERR bit (Receive Fatal Error Interrupt Enable)

The RXFERR bit controls the Receive Fatal Error Interrupt permission.

When a Receive Fatal Error Interrupt occurs, the SQCH0SR.RXFERR flag is set to 1.

To enable the Receive Fatal Error Interrupt, set the RXFERR bit to 1.

RXFAIL bit (Receive Fail Interrupt Enable)

The RXFAIL bit controls the Receive Fail Interrupt permission.

When a Receive Fail Interrupt occurs, the SQCH0SR.RXFAIL flag is set to 1.

To enable the Receive Fail Interrupt, set the RXFAIL bit to 1.

RXPFAIL bit (Receive Packet Data Fail Interrupt Enable)

The RXPFAIL bit controls the Receive Packet Data Fail Interrupt permission.

When a Receive Packet Data Fail Interrupt occurs, the SQCH0SR.RXPFAIL flag is set to 1.

To enable the Receive Packet Data Fail Interrupt, set the RXPFAIL bit to 1.

RXCORERR bit (Receive Correctable Error Interrupt Enable)

The RXCORERR bit controls the Receive Correctable Error Interrupt permission.

When a Receive Correctable Error Interrupt occurs, the SQCH0SR.RXCORERR flag is set to 1.

To enable the Receive Correctable Error Interrupt, set the RXCORERR bit to 1.

RXAKE bit (Receive Acknowledge and Error Report Packet Interrupt Enable)

The RXAKE bit controls the Receive Acknowledge and Error Report Packet Interrupt permission.

When a Receive Acknowledge and Error Report Packet Interrupt occurs, the SQCH0SR.RXAKE flag is set to 1.

To enable the Receive Acknowledge and Error Report Packet Interrupt, set the RXAKE bit to 1.

58.2.57 SQCH1SET0R : Sequence Channel 1 Set 0 Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x600

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	START
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	START	Sequence Operation Start 0: No operation 1: Start the sequence operation	W
22:1	—	These bits are read as 0. The write value should be 0.	W
23	—	This bit is read as 1. The write value should be 1.	W
31:24	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

START bit (Sequence Operation Start)

The START bit is the trigger to start the sequence operation. The sequence operations are executed in sequence from Descriptor-0.

Setting the START bit to 1 when the SQCH0SR.RUNNING flag or the SQCH1SR.RUNNING flag is set to 1 is prohibited.

58.2.58 SQCH1SR : Sequence Channel 1 Status Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x610

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	RXAKE	RXCORERR	RXPFAIL	RXFAIL	RXFERR	—	TXIBERR	—	—	—	—	SIZEERR	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADESFIN	—	—	—	AACTFIN	—	RUNNING	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0.	R
2	RUNNING	Sequence Operation Running Status 0: Sequence operation is stopped 1: Sequence operation is running	R
3	—	This bit is read as 0.	R
4	AACTFIN	All Actions Finish Interrupt Flag 0: All actions of a descriptor have not finished 1: All actions of a descriptor finished operations	R

Bit	Symbol	Function	R/W
7:5	—	These bits are read as 0.	R
8	ADESFIN	All-Descriptors Finish Interrupt Flag 0: All descriptors have not finished 1: All descriptors finished operations	R
18:9	—	These bits are read as 0.	R
19	SIZEERR	Packet Size Error Interrupt Flag 0: No error 1: Sequence packet size is too large	R
23:20	—	These bits are read as 0.	R
24	TXIBERR	Tx Internal Bus Error Interrupt Flag 0: No error 1: Internal AXI bus read error occurred	R
25	—	This bit is read as 0.	R
26	RXFERR	Receive Fatal Error Interrupt Flag 0: No error 1: Fatal timeout occurred during BTA	R
27	RXFAIL	Receive Fail Interrupt Flag 0: No error 1: Expected receive did not take place	R
28	RXPFAIL	Receive Packet Data Fail Interrupt Flag 0: No error 1: Received packet data is not stored correctly	R
29	RXCORERR	Receive Correctable Error Interrupt Flag 0: No error 1: Received packets have correctable error	R
30	RXAKE	Receive Acknowledge and Error Report Packet Interrupt Flag 0: No error 1: Acknowledge and Error Report Packet is received	R
31	—	This bit is read as 0.	R

Note: S-TYPE-3, P-TYPE-3

RUNNING bit (Sequence Operation Running Status)

The RUNNING bit indicates the state of the sequence operation.

Changing the settings of the descriptor is prohibited while the RUNNING bit is 1.

AACTFIN flag (All Actions Finish Interrupt Flag)

The AACTFIN flag is set if this descriptor's all actions are finished with SQCHnDSCmCR.FINACT =1.

The AACTFIN flag is not cleared automatically, so set the SQCH1SCR.AACTFIN bit to 1 to clear the AACTFIN flag.

ADESFIN flag (All-Descriptors Finish Interrupt Flag)

The ADESFIN flag is set when all descriptors finish operations.

When Descriptor-m whose the SQCHnDSCmAR.NXACT[1:0] bits are set to 00b finishes the operation, the ADESFIN flag is set to 1.

The ADESFIN flag is not cleared automatically, so set the SQCH1SCR.ADESFIN bit to 1 to clear the ADESFIN flag.

SIZEERR flag (Packet Size Error Interrupt Flag)

The SQCH1SR.SIZEERR flag is set to 1 when the sequence packet is too large.

The SQCH1SR.SIZEERR flag is not cleared automatically, so set the SQCH1SCR.SIZEERR bit to 1 to clear the SQCH1SR.SIZEERR flag.

TXIBERR flag (Tx Internal Bus Error Interrupt Flag)

The TXIBERR flag is set to 1 when the internal AXI bus read is failed.

The TXIBERR flag is not cleared automatically, so set the SQCH1SCR.TXIBERR bit to 1 to clear the TXIBERR flag.

RXFERR flag (Receive Fatal Error Interrupt Flag)

The RXFERR flag is set to 1 when the fatal timeout occurred during BTA.

The FERRSR.TATO flag or the FERRSR.LRXHTO flag is also set to 1.

The RXFERR flag is not cleared automatically, so set the SQCH1SCR.RXFERR bit to 1 to clear the RXFERR flag.

RXFAIL flag (Receive Fail Interrupt Flag)

The RXFAIL flag is set to 1 when the expected receive did not done.

The RXSR.PRTOERR flag, the RXSR.ECCERRM flag, the RXSR.MLFERR flag, or the RXSR.NORESERR flag is also set to 1.

The RXFAIL flag is not cleared automatically, so set the SQCH1SCR.RXFAIL bit to 1 to clear the RXFAIL flag.

RXPFAIL flag (Receive Packet Data Fail Interrupt Flag)

The RXPFAIL flag is set to 1 when the received packet header is stored correctly, but the packet data is not stored correctly.

The RXSR.CRCERR flag, the RXSR.WCERR flag, the RXSR.UNEXERR flag, the RXSR.RSIZEERR flag, the RXSR.RXOVFERR flag, or the RXSR.IBERR flag is also set to 1.

The RXPFAIL flag is not cleared automatically, so set the SQCH1SCR.RXPFAIL bit to 1 to clear the RXPFAIL flag.

RXCORERR flag (Receive Correctable Error Interrupt Flag)

The RXCORERR flag is set to 1 when the received packet has correctable errors.

The RXSR.ECCERRS flag is also set to 1.

The RXCORERR flag is not cleared automatically, so set the SQCH1SCR.RXCORERR bit to 1 to clear the RXCORERR flag.

RXAKE flag (Receive Acknowledge and Error Report Packet Interrupt Flag)

The RXAKE flag is set to 1 when the Acknowledge and Error Report Packet is received.

The RXSR.RXAKE flag is also set to 1.

The RXAKE flag is not cleared automatically, so set the SQCH1SCR.RXAKE bit to 1 to clear the RXAKE flag.

58.2.59 SQCH1SCR : Sequence Channel 1 Status Clear Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x614

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	RXAKE	RXCORERR	RXPFAIL	RXFAIL	RXFERR	—	TXIBERR	—	—	—	—	SIZEERR	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADESFIN	—	—	—	AACTFIN	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	W
4	AACTFIN	All Actions Finish Interrupt Flag Clear 0: No operation 1: Clear the SQCH1SR.AACTFIN flag	W
7:5	—	These bits are read as 0. The write value should be 0.	W

Bit	Symbol	Function	R/W
8	ADESFIN	All-Descriptors Finish Interrupt Flag Clear 0: No operation 1: Clear the SQCH1SR.ADESFIN flag	W
18:9	—	These bits are read as 0. The write value should be 0.	W
19	SIZEERR	Packet Size Error Interrupt Flag Clear 0: No operation 1: Clear the SQCH1SR.SIZEERR flag	W
23:20	—	These bits are read as 0. The write value should be 0.	W
24	TXIBERR	Tx Internal Bus Error Interrupt Flag Clear 0: No operation 1: Clear the SQCH1SR.TXIBERR flag	W
25	—	This bit is read as 0. The write value should be 0.	W
26	RXFERR	Receive Fatal Error Interrupt Flag Clear 0: No operation 1: Clear the SQCH1SR.RXFERR flag	W
27	RXFAIL	Receive Fail Interrupt Flag Clear 0: No operation 1: Clear the SQCH1SR.RXFAIL flag	W
28	RXPFAIL	Receive Packet Data Fail Interrupt Flag Clear 0: No operation 1: Clear the SQCH1SR.RXPFAIL flag	W
29	RXCORERR	Receive Correctable Error Interrupt Flag Clear 0: No operation 1: Clear the SQCH1SR.RXCORERR flag	W
30	RXAKE	Receive Acknowledge and Error Report Packet Interrupt Flag Clear 0: No operation 1: Clear the SQCH1SR.RXAKE flag	W
31	—	This bit is read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

AACTFIN bit (All Actions Finish Interrupt Flag Clear)

Set the AACTFIN bit to 1 to clear the SQCH1SR.AACTFIN flag.

ADESFIN bit (All-Descriptors Finish Interrupt Flag Clear)

Set the ADESFIN bit to 1 to clear the SQCH1SR.ADESFIN flag.

SIZEERR bit (Packet Size Error Interrupt Flag Clear)

Set the SIZEERR bit to 1 to clear the SQCH1SR.SIZEERR flag.

TXIBERR bit (Tx Internal Bus Error Interrupt Flag Clear)

Set the TXIBERR bit to 1 to clear the SQCH1SR.TXIBERR flag.

RXFERR bit (Receive Fatal Error Interrupt Flag Clear)

Set the RXFERR bit to 1 to clear the SQCH1SR.RXFERR flag.

RXFAIL bit (Receive Fail Interrupt Flag Clear)

Set the RXFAIL bit to 1 to clear the SQCH1SR.RXFAIL flag.

RXPFAIL bit (Receive Packet Data Fail Interrupt Flag Clear)

Set the RXPFAIL bit to 1 to clear the SQCH1SR.RXPFAIL flag.

RXCORERR bit (Receive Correctable Error Interrupt Flag Clear)

Set the RXCORERR bit to 1 to clear the SQCH1SR.RXCORERR flag.

RXAKE bit (Receive Acknowledge and Error Report Packet Interrupt Flag Clear)

Set the RXAKE bit to 1 to clear the SQCH1SR.RXAKE flag.

58.2.60 SQCH1IER : Sequence Channel 1 Interrupt Enable Register

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x618

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	RXAKE	RXCORERR	RXPFAIL	RXFAIL	RXFERR	—	TXIBERR	—	—	—	—	SIZEERR	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADESFIN	—	—	—	AACTFIN	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	AACTFIN	All Actions Finish Interrupt Enable 0: Disable 1: Enable	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	ADESFIN	All-Descriptors Finish Interrupt Enable 0: Disable 1: Enable	R/W
18:9	—	These bits are read as 0. The write value should be 0.	R/W
19	SIZEERR	Packet Size Error Interrupt Enable 0: Disable 1: Enable	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
24	TXIBERR	Tx Internal Bus Error Interrupt Enable 0: Disable 1: Enable	R/W
25	—	This bit is read as 0. The write value should be 0.	R/W
26	RXFERR	Receive Fatal Error Interrupt Enable 0: Disable 1: Enable	R/W
27	RXFAIL	Receive Fail Interrupt Enable 0: Disable 1: Enable	R/W
28	RXPFAIL	Receive Packet Data Fail Interrupt Enable 0: Disable 1: Enable	R/W
29	RXCORERR	Receive Correctable Error Interrupt Enable 0: Disable 1: Enable	R/W
30	RXAKE	Receive Acknowledge and Error Report Packet Interrupt Enable 0: Disable 1: Enable	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

AACTFIN bit (All Actions Finish Interrupt Enable)

The AACTFIN bit controls the All Actions Finish Interrupt permission.

When an All Actions Finish Interrupt occurs, the SQCH1SR.AACTFIN flag is set to 1.

To enable the All Actions Finish Interrupt, set the AACTFIN bit to 1.

ADESFIN bit (All-Descriptors Finish Interrupt Enable)

The ADESFIN bit controls the All-Descriptors Finish Interrupt permission.

When an All-Descriptors Finish Interrupt occurs, the SQCH1SR.ADESFIN flag is set to 1.

To enable the All-Descriptors Finish Interrupt, set the ADESFIN bit to 1.

SIZEERR bit (Packet Size Error Interrupt Enable)

The SIZEERR bit controls the Packet Size Error Interrupt permission.

When a Packet Size Error Interrupt occurs, the SQCH1SR.SIZEERR flag is set to 1.

To enable the Packet Size Error Interrupt, set the SIZEERR bit to 1.

TXIBERR bit (Tx Internal Bus Error Interrupt Enable)

The TXIBERR bit controls the Tx Internal Bus Error Interrupt permission.

When a Tx Internal Bus Error Interrupt occurs, the SQCH1SR.TXIBERR flag is set to 1.

To enable the Tx Internal Bus Error Interrupt, set the TXIBERR bit to 1.

RXFERR bit (Receive Fatal Error Interrupt Enable)

The RXFERR bit controls the Receive Fatal Error Interrupt permission.

When a Receive Fatal Error Interrupt occurs, the SQCH1SR.RXFERR flag is set to 1.

To enable the Receive Fatal Error Interrupt, set the RXFERR bit to 1.

RXFAIL bit (Receive Fail Interrupt Enable)

The RXFAIL bit controls the Receive Fail Interrupt permission.

When a Receive Fail Interrupt occurs, the SQCH1SR.RXFAIL flag is set to 1.

To enable the Receive Fail Interrupt, set the RXFAIL bit to 1.

RXPFAIL bit (Receive Packet Data Fail Interrupt Enable)

The RXPFAIL bit controls the Receive Packet Data Fail Interrupt permission.

When a Receive Packet Data Fail Interrupt occurs, the SQCH1SR.RXPFAIL flag is set to 1.

To enable the Receive Packet Data Fail Interrupt, set the RXPFAIL bit to 1.

RXCORERR bit (Receive Correctable Error Interrupt Enable)

The RXCORERR bit controls the Receive Correctable Error Interrupt permission.

When a Receive Correctable Error Interrupt occurs, the SQCH1SR.RXCORERR flag is set to 1.

To enable the Receive Correctable Error Interrupt, set the RXCORERR bit to 1.

RXAKE bit (Receive Acknowledge and Error Report Packet Interrupt Enable)

The RXAKE bit controls the Receive Acknowledge and Error Report Packet Interrupt permission.

When a Receive Acknowledge and Error Report Packet Interrupt occurs, the SQCH1SR.RXAKE flag is set to 1.

To enable the Receive Acknowledge and Error Report Packet Interrupt, set the RXAKE bit to 1.

58.2.61 SQCHnDSCmAR : Sequence Channel n Descriptor-m A Register (n = 0 to 1, m = 0 to 7)

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x780 + (0x10 × m) + (0x80 × n)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	NXACT[1:0]		BTA[1:0]		SPD	FMT	VC[1:0]		DT[5:0]					
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DATA1[7:0]								DATA0[7:0]							
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	DATA0[7:0]	Data 0 Set Data0 of Tx Packet Header.*1	R/W
15:8	DATA1[7:0]	Data 1 Set Data1 of Tx Packet Header.*1	R/W
21:16	DT[5:0]	Data Type Set Data type of Tx Packet Header.	R/W
23:22	VC[1:0]	Virtual Channel Set identifier of virtual channel of Tx Packet Header.	R/W
24	FMT	Format Set packet format of Tx Packet Header. 0: Short Packet 1: Long Packet	R/W
25	SPD	Speed Set the speed type for Tx request. 0: High Speed 1: Low Power	R/W
27:26	BTA[1:0]	Bus Turn Around Sets either Tx Request without Bus Turn-Around (BTA) or no-operation. Or else, sets Tx Request with BTA. 0 0: Tx Request without BTA or no-operation 0 1: Tx non-Read Request with BTA 1 0: Tx Read Request with BTA 1 1: BTA only	R/W
29:28	NXACT[1:0]	Next Action 0 0: Terminate the sequence operation after this descriptor processing is finished. 0 1: Start the next descriptor processing after this descriptor processing is finished. Others: Settings prohibited	R/W
31:30	—	The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The maximum size for Command Transfer mode Long Packet by using Packet Payload Data Register (SQCHnDSCmBR.DTSEL[1:0] = 00b and SQCHnDSCmAR.FMT = 1) is 16 bytes.

This register is not initialized by any reset. Therefore, if the user uses sequence operation, be sure to set the values according to the above description for the reserved area as well.

DATA0[7:0] bits (Data 0)

When Tx Packet format is Long Packet (FMT = 1), this value is lower 8 bits of the Word Count (WC).

DATA1[7:0] bits (Data 1)

When Tx Packet format is Long Packet (FMT = 1), this value is upper 8 bits of the Word Count (WC).

DT[5:0] bits (Data Type)

The DT[5:0] bits set the data type of Tx Packet Header.

VC[1:0] bits (Virtual Channel)

The VC[1:0] bits set the identifier of the virtual channel of Tx Packet Header.

FMT bit (Format)

The FMT bit set the packet format of Tx Packet Header. This field should be set by the data type.

SPD bit (Speed)

The SPD bit selects the speed type for Tx request. It is prohibited to set SPD bit to 1 during the Video Mode operation.

BTA[1:0] bits (Bus Turn Around)

The BTA[1:0] bits set either Tx Request without Bus Turn-Around (BTA) or no-operation. Or else, sets Tx Request with BTA.

NXACT[1:0] bits (Next Action)

The NXACT[1:0] bits select the next action after the Descriptor-m processing is finished.

When the SQCHnDSC7AR.NXACT[1:0] bits for Descriptor-7 are set to 01b (Start the next descriptor processing after this descriptor processing is finished), the next descriptor after Descriptor-7 is Descriptor-0.

58.2.62 SQCHnDSCmBR : Sequence Channel n Descriptor-m B Register (n = 0 to 1, m = 0 to 7)

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x784 + (0x10 × m) + (0x80 × n)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	DTSEL[1:0]		—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
23:0	—	The write value should be 0.	R/W
25:24	DTSEL[1:0]	Data Select Select the data buffer for Long Packet payload 0 0: Use Packet Payload Data Register (TXPPDxR, RXPPDxR) 0 1: Use Memory space Others Setting prohibited	R/W
31:26	—	The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register is not initialized by any reset. Therefore, if the user uses sequence operation, be sure to set the values according to the above description for the reserved area as well.

DTSEL[1:0] bits (Data Select)

The DTSEL[1:0] bits select the data source for Long Packet payload. Set the DTSEL[1:0] bits to 00b, in the case of the write request with the short packet.

When the SQCHnDSCmBR.DTSEL[1:0] bits are 00b, DSI Host uses Packet Payload Data Register (TXPPDxR and RXPPDxR, x = 0 to 3) for the Long Packet payload data buffer. In this setting, the Long Packet write request data size shall be same or lower than 16 bytes. When the read request data to save has over 16 bytes word count, it reports as error.

When the SQCHnDSCmBR.DTSEL[1:0] bits are 01b, DSI Host uses Memory space for the Long Packet payload data buffer. DSI Host transfer data stored in Memory space at the address specified in the SQCHnDSCmDR register at the write request. And stores data to Memory space at the address specified in the SQCHnDSCmDR register at the read request. The read header always saved to the RXRSSxR register.

58.2.63 SQCHnDSCmCR : Sequence Channel n Descriptor-m C Register (n = 0 to 1, m = 0 to 7)

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x788 + (0x10 × m) + (0x80 × n)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ACTCODE[7:0]								—	AUXOP	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FINACT
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
0	FINACT	Finish Action Select whether to set the SQCHnSR.AACTFIN bit to 1 when all actions of this Descriptor-m are finished. 0: Disable 1: Enable	R/W
21:1	—	The write value should be 0.	R/W
22	AUXOP	Auxiliary Operation 0: Not use auxiliary operation 1: Execute auxiliary operation*1	R/W
23	—	The write value should be 0.	R/W
31:24	ACTCODE[7:0]	Action Code When not use auxiliary operation (SQCHnDSCmCR.AUXOP = 0): Set the slot number for storing the received result during the BTA action.*2 0 0 0 0 0 0 0 0: Slot-0 (RXRSS0R) 0 0 0 0 0 0 0 1: Slot-1 (RXRSS1R) 0 0 0 0 0 0 1 0: Slot-2 (RXRSS2R) 0 0 0 0 0 0 1 1: Slot-3 (RXRSS3R) Other settings are prohibited. When execute auxiliary operation (SQCHnDSCmCR.AUXOP = 1): Select the action for the auxiliary operation. 0 0 0 0 0 0 0 0: Send Reset-Trigger 0 0 0 0 1 0 0 0: No-operation Other settings are prohibited.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. It is prohibited to set this bit to 1 during the Video mode operation is running.
Note 2. Setting the same slot number to the SQCHnDSCmCR.ACTCODE[7:0] bits of a different descriptor is prohibited.

This register is not initialized by any reset. Therefore, if the user uses sequence operation, be sure to set the values according to the above description for the reserved area as well.

FINACT bit (Finish Action)

The FINACT bit controls whether to set the SQCHnSR.AACTFIN bit to 1 when all actions of this Descriptor-m are finished. Since each descriptor's operation is not same, the descriptors may finish out of order. For example, Descriptor-1 may finish before Descriptor-0.

AUXOP bit (Auxiliary Operation)

The AUXOP bit selects whether executes the auxiliary operation or not. The action of the auxiliary operation is specified by the SQCHnDSCmCR.ACTCODE[7:0] bits. When this AUXOP bit is set to 1, the SQCHnDSCmAR.BTA[1:0] bits should be set to 00b.

ACTCODE[7:0] bits (Action Code)

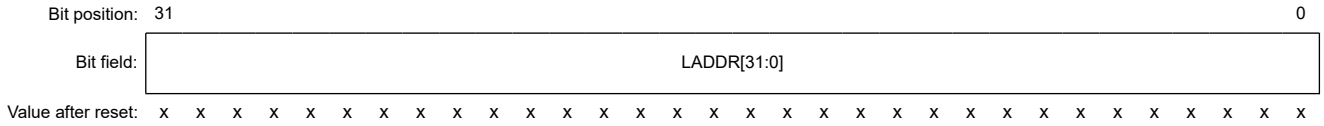
When not use auxiliary operation (SQCHnDSCmCR.AUXOP = 0), the ACTCODE[7:0] bits select the slot number for storing the received result during the BTA action. Set these bits to 0x00 for non-BTA action. Setting the same slot number to the SQCHnDSCmCR.ACTCODE[7:0] bits of a different descriptor is prohibited.

When execute the auxiliary operation (SQCHnDSCmCR.AUXOP = 1), the ACTCODE[7:0] bits select the action that can be taken as the auxiliary operation.

58.2.64 SQCHnDSCmDR : Sequence Channel n Descriptor-m D Register (n = 0 to 1, m = 0 to 7)

Base address: MIPI_DSI = 0x4034_6000
MIPI_DSI_NS = 0x5034_6000

Offset address: 0x78C + (0x10 × m) + (0x80 × n)



Bit	Symbol	Function	R/W
31:0	LADDR[31:0]	Lower Address Set the lower 32-bit address of Memory space to store the long packet payload data for Sequence Operation Channel-n Descriptor-m.	R/W

Note: S-TYPE-3, P-TYPE-3

The SQCHnDSCmDR register is not initialized by any reset. Therefore, if the user uses sequence operation, be sure to set all the bits.

LADDR[31:0] bits (Lower Address)

When the Memory space is selected for Long Packet payload (SQCHnDSCmBR.DTSEL[1:0] = 01b) and the Sequence Mode Operation Format is Long packet (SQCHnDSCmAR.FMT = 1), the DSI Host accesses Memory space at the address specified in this LADDR[31:0] bits. In the case of write operation, DSI Host transfer data stored in Memory space. In the case of read operation, DSI Host store data to Memory space.

58.3 Operation

58.3.1 Interrupt

Table 58.2 Interrupt Signals (1 of 3)

Interrupt Signal	Category	Interrupt Condition	Interrupt Source Register
DSI_SEQ0	Sequence Operation channel 0 interrupt	Receive Acknowledge and Error Report Packet	SQCH0SR.RXAKE
		Receive Correctable Error	SQCH0SR.RXCORERR
		Receive Packet Data Fail	SQCH0SR.RXPFAIL
		Receive Fail	SQCH0SR.RXFAIL
		Receive Fatal Error	SQCH0SR.RXFERR
		Tx Internal Bus Error	SQCH0SR.TXIBERR
		All-Descriptors Finish	SQCH0SR.ADESFIN
		All Actions Finish	SQCH0SR.AACTFIN

Table 58.2 Interrupt Signals (2 of 3)

Interrupt Signal	Category	Interrupt Condition	Interrupt Source Register
DSI_SEQ1	Sequence Operation channel 1 interrupt	Receive Acknowledge and Error Report Packet	SQCH1SR.RXAKE
		Receive Correctable Error	SQCH1SR.RXCORERR
		Receive Packet Data Fail	SQCH1SR.RXPFAIL
		Receive Fail	SQCH1SR.RXFAIL
		Receive Fatal Error	SQCH1SR.RXFERR
		Tx Internal Bus Error	SQCH1SR.TXIBERR
		Packet Size Error	SQCH1SR.SIZEERR
		All-Descriptors Finish	SQCH1SR.ADESFIN
		All Actions Finish	SQCH1SR.AACTFIN
DSI_VIN1	Video mode operation channel1 interrupt	Video Buffer Overflow Error	VMSR.VBUFOVF
		Video Buffer Underflow	VMSR.VBUFUDF
		Timing Error	VMSR.TIMERR
		Video Mode Operation Ready	VMSR.VIRDY
		Video Mode Operation Stop	VMSR.STOP
		Video Mode Operation Start	VMSR.START
DSI_RCV	DSI Packet Receive interrupt	Acknowledge and Error Report Receive	RXSR.RXAKE
		Single Bit ECC Error	RXSR.ECCERRS
		Return Packet Size Error	RXSR.RSIZEERR
		No Response Error	RXSR.NORESERR
		Peripheral Response Timeout Error	RXSR.PRTOERR
		Receive Buffer Overflow Error	RXSR.RXOVFERR
		Internal Bus Error	RXSR.IBERR
		CRC Error	RXSR.CRCERR
		Word Count Error	RXSR.WCERR
		Unexpected Packet Error	RXSR.UNEXERR
		Multi Bit ECC Error	RXSR.ECCERRM
		Malformed Error	RXSR.MLFERR
		External Tearing Effect Detect	RXSR.EXTEDET
		ACK Trigger Receive	RXSR.RXACK
		EoTp Receive	RXSR.RXEOTP
		Response Packet Receive	RXSR.RXRESP
		Turnaround Acknowledge Timeout	RXSR.TATO
		LP-RX Host Processor Timeout	RXSR.LRXHTO
		BTA Request End	RXSR.BTAREND

Table 58.2 Interrupt Signals (3 of 3)

Interrupt Signal	Category	Interrupt Condition	Interrupt Source Register
DSI_FERR	DSI Fatal Error interrupt	LP1 Contention Error	FERRSR.CLP1
		LP0 Contention Error	FERRSR.CLP0
		Control Error	FERRSR.CTRL
		LPDT Sync Error	FERRSR.SYNCESC
		Escape mode Entry Error	FERRSR.ESCENT
		Turnaround Acknowledge Timeout	FERRSR.TATO
		LP-RX Host Processor Timeout	FERRSR.LRXHTO
		HS TX Timeout Interrupt	FERRSR.HTXTO
DSI_PPI	DSI D-PHY PPI interrupt	Data Lane transition from ULPS	PLSR.DLULPEXT
		Data Lane transition to ULPS	PLSR.DLULPENT
		Clock Lane mode transition from HS to LP	PLSR.CLHS2LP
		Clock Lane mode transition from LP to HS	PLSR.CLLP2HS
		Clock Lane transition from ULPS	PLSR.CLULPEXT
		Clock Lane transition to ULPS	PLSR.CLULPENT
		Data Lane-0 direction transition from transmitter to receiver	PLSR.DL0TX2RX
		Data Lane-0 direction transition from receiver to transmitter	PLSR.DL0RX2TX

The relationship between Interrupt Signals and Interrupt Conditions is shown in Table 58.2

The interrupt signal consists of combinatorial logic shown in Figure 58.3. The interrupt signal from a Status register is enabled by the signal related Interrupt Enable register. These interrupt signals are bound by the OR logic. All interrupt signals are synchronized to pclk and the level signal, and are active-high.

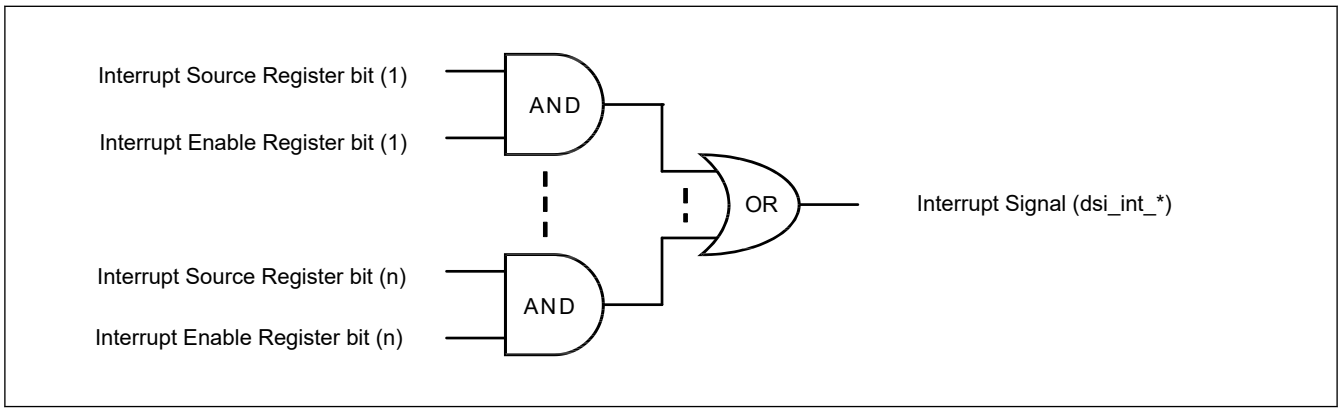


Figure 58.3 Interrupt Signal Output

58.3.2 Software Reset

The software reset is for the initialization of the DSI Host.*1

Initializing by the software reset is optional. To initialize the DSI Host using the software reset, follow the steps below.

1. Set the RSTCR.SWRST bit to 1 for the software reset.
2. Wait for the software reset process to start.
   Wait until the following bits are set to 1.
   (RSTSR.RSTHS, RSTSR.RSTLP, RSTSR.RSTAPB, RSTSR.RSTAXI, RSTSR.RSTV)
3. Set the following registers (optional)

(TXSETR, ULPSSETR, DSISETR, CLSTPTSETR, LPTRNSTSETR, PRESPTOBTASETR, PRESPTOLPSETR, PRESPTOHSSETR, HSTXTOSETR, LRXHTOSETR, TATOSETR)
These registers are allowed to be changed only during this process or initialization at initial startup.

4. Set the RSTCR.FTXSTP bit to 1 to transition to TX-Stop state (LP-11) for the data lane.
5. Wait for the valid data lane to transition to the stop state.
Wait until the RSTSR.DL0DIR bit is cleared to 0 and the RSTSR.DL1STP and DL0STP bits are set to 1 (RSTSR.DL1STP = 0 and RSTSR.DL0STP = 1 for a single lane)
6. Clear the RSTCR.FTXSTP bit to 0 to finish the operation of the transition to TX-Stop state.
7. Clear the RSTCR.SWRST bit to 0 to finish the software reset process.
8. Wait for the software reset process to end.
Wait until the following bits are cleared to 0.
(RSTSR.RSTHS, RSTSR.RSTLP, RSTSR.RSTAPB, RSTSR.RSTAXI, RSTSR.RSTV)

Note 1. The following registers are not initialized by the software reset.
- RSTCR
- RSTSR
- SQCHnDSCmAR (n = 0 to 1, m = 0 to 7)
- SQCHnDSCmBR (n = 0 to 1, m = 0 to 7)
- SQCHnDSCmCR (n = 0 to 1, m = 0 to 7)
- SQCHnDSCmDR (n = 0 to 1, m = 0 to 7)
- TXSETR
- ULPSSETR
- DSISETR
- CLSTPTSETR
- LPTRNSTSETR
- PRESPTOBTASETR
- PRESPTOLPSETR
- PRESPTOHSSETR
- HSTXTOSETR
- LRXHTOSETR
- TATOSETR

58.3.3 Start of HS Clock

HS clock is controlled by the software. Set the registers according to the following flow.

1. Set the TXSETR.CLEN bit to 1 to enable the D-PHY clock lane before starting the Video Mode operation or Sequence mode operation.
2. Set the HSCLKSETR.HSCLST bit to 1. At the same time, set the HSCLKSETR.HSCLMD bit to 1 when using the continuous clock mode.
In the continuous clock mode, the clock lane is always put the HS state.
In the non-continuous clock mode, the DSI Host automatically controls the transition between LP and HS states of the clock lane.
3. Wait until the PLSR.CLLP2HS bit is set 1. (only in the continuous clock mode)

58.3.4 Stop of HS Clock

1. Clear the HSCLKSETR.HSCLST bit to 0 to transition the clock lane to the LP state.
2. Wait until the PLSR.CLHS2LP bit is set 1. (only in the continuous clock mode)
The user can also check the status of the clock lane by reading the PLSR.CLSTP bit.

58.3.5 Command Mode Operation

The DSI Host supports two basic modes of operation: Command Mode and Video Mode. This section describes how to operate in Command Mode.

The Command Mode of this DSI Host is intended to be used for communication such as configuration settings for Peripheral before sending video signals.

In Command Mode, a descriptor list that describes the sending and receiving process in advance is used.

Processing using descriptors is referred to here as Sequence Operation.

The DSI Host has two channels for Sequence Operation (Sequence Operation Channel-n, n = 0 to 1).

Channel-0 supports only LP mode (LP-TX, LP-RX), while Channel-1 supports LP mode (LP-TX, LP-RX) and HS mode (HS-TX). There are 8 descriptors for each channel.

The Command Mode of this DSI Host supports only Sequence Operation using these descriptors.

Each descriptor has four control registers for Sequence Operation (SQCHnDSCmAR, SQCHnDSCmBR, SQCHnDSCmCR, SQCHnDSCmDR) that specify the transmit and receive operations.

The payload data for Long Packet is stored in Memory space or Packet Payload Data Register, which is shared among descriptors. Memory space for transfer is dedicated for Channel-0 and Channel-1, respectively. (Figure 58.4)

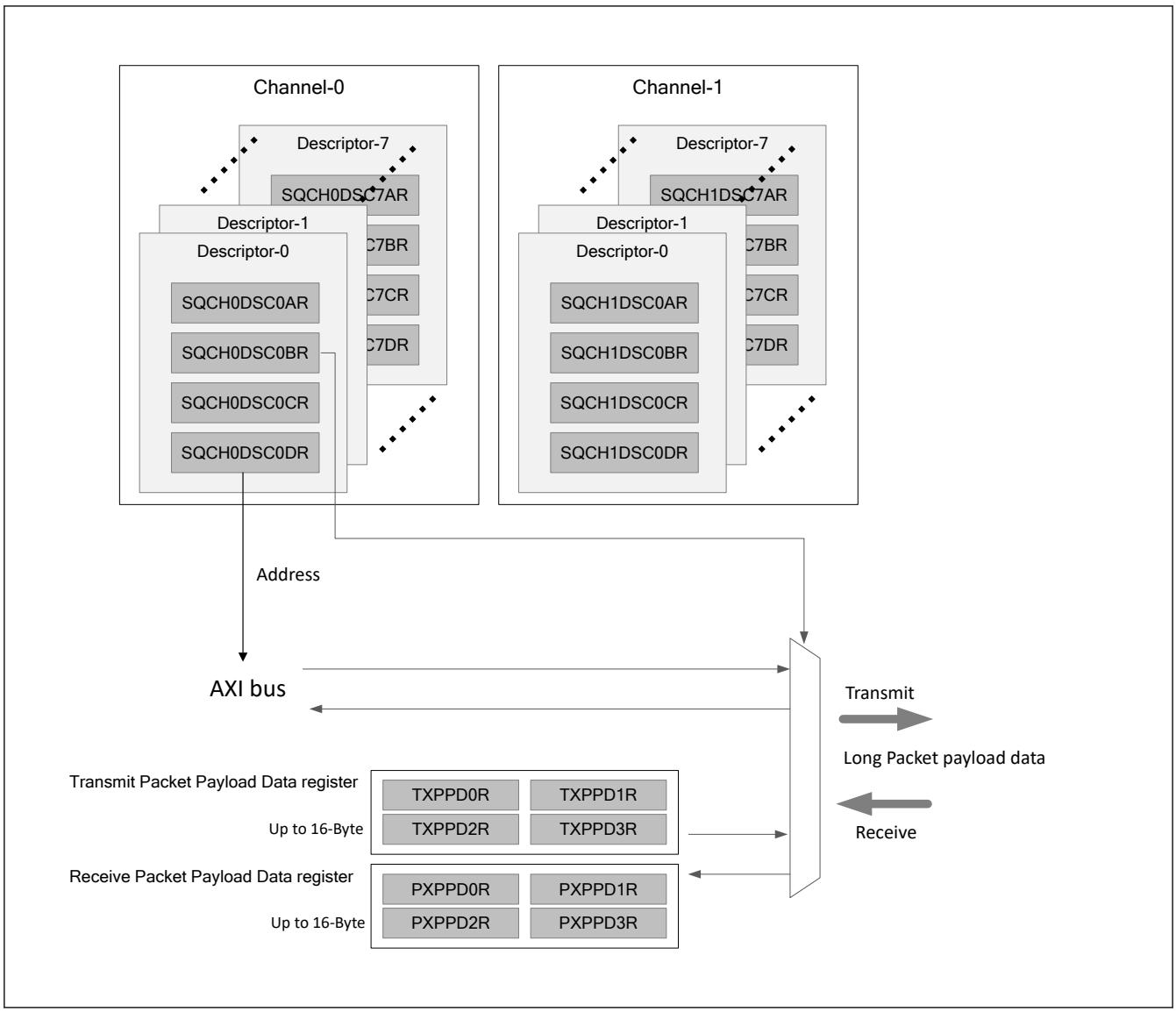


Figure 58.4 Descriptors and Packet Data Buffers for Sequence Operation

58.3.5.1 Basic Running Sequence

By setting the SQCHnSET0R.START bit to 1, the Sequence Operation Channel-n starts processing the descriptor. When the process of Descriptor-m which the SQCHnDSCmAR.NXACT[1:0] bits are set to 00b is completed, the sequence operations stop.

After setting the SQCHnSET0R.START bit to 1, the SQCHnSR.RUNNING bit is set to 1. Once the SQCHnSR.RUNNING bit is set to 1, it is prohibited to change the settings of the following registers until the sequence operation is stopped.

- SQCHnDSCmAR (n = 0 to 1, m = 0 to 7)
- SQCHnDSCmBR (n = 0 to 1, m = 0 to 7)
- SQCHnDSCmCR (n = 0 to 1, m = 0 to 7)
- SQCHnDSCmDR (n = 0 to 1, m = 0 to 7)

58.3.5.2 Single Packet Transmission

This section describes the transmission of the non-read packets in Command mode using Sequence operation.

In the non-read packet transmission, set the following registers before sending the packets.

- SQCHnDSCmAR.NXACT[1:0] : 00b
- SQCHnDSCmAR.FMT : 0 (Short Packet) or 1 (Long Packet)
- SQCHnDSCmAR.SPD : 0 (High Speed) or 1 (Low Speed)
- SQCHnDSCmAR.BTA[1:0] : 00b (without BTA) or 01b (with BTA)
- SQCHnDSCmAR.VC[1:0]
- SQCHnDSCmAR.DT[5:0]
- SQCHnDSCmAR.DATA0[7:0]
- SQCHnDSCmAR.DATA1[7:0]
- SQCHnDSCmBR.DTSEL[1:0] : Specify where to store the payload data for the non-read packets

  00b: TXPPDxR (x = 0 to 3). Payloads size is limited up to 16 bytes.

  01b: Memory space. Payloads size is limited up to 4KB.
- SQCHnDSCmCR.AUXOP : 0

When setting up a Long Packet (SQCHnDSCmAR.FMT = 1), the payload data should be in the format of little endian at the location specified by the SQCHnDSCmBR.DTSEL[1:0] bits.

When setting up a Short Packet (SQCHnDSCmAR.FMT = 0), no payload data is required.

After the above settings, send a non-read packet by setting the SQCHnSET0R.START bit to 1. When Single Packet Transmission is complete, the SQCHnSR.ADESFIN bit is set to 1.

When the SQCHnDSCmAR.BTA[1:0] bits are set to 01b, BTA will be taken place after sending a non-read packet. After DSI Host gives bus ownership to the peripheral, An ACK trigger Message or an Acknowledge and Error Report Packet may be returned from the peripheral.

The user can get a summary of the reception result by reading the RXRSSxR register with the number-x specified in the SQCHnDSCmCR.ACTCODE[7:0] bits, and can get the reception result details by reading the RXSR register.

58.3.5.3 Single Packet Reception

This section describes sending the read packets and receiving the response packets in Command mode using Sequence operation.

Since the read packet is a Short Packet, set the following registers before sending the packet.

- SQCHnDSCmAR.NXACT[1:0] : 00b
- SQCHnDSCmAR.FMT : 0 (Short Packet)
- SQCHnDSCmAR.SPD : 0 (High Speed) or 1 (Low Speed)
- SQCHnDSCmAR.BTA[1:0] : 10b (Read Request with BTA)
- SQCHnDSCmAR.VC[1:0]
- SQCHnDSCmAR.DT[5:0]
- SQCHnDSCmAR.DATA0[7:0]
- SQCHnDSCmAR.DATA1[7:0]
- SQCHnDSCmBR.DTSEL[1:0] : Specify where to store the payload data for response packets

00b : RXPPDxR (x = 0 to 3). Payloads size is limited up to 16 bytes.

01b : Memory space. Payloads size is limited up to 128 bytes.

- SQCHnDSCmCR.AUXOP : 0
- SQCHnDSCmCR.ACTCODE[7:0] : Received Result Save Slot number

When the SQCHnDSCmBR.DTSEL[1:0] bits are set to 00b, the payload data reception larger than 16 bytes is not supported. If there is a possibility of receiving the payload data larger than 16 bytes, set the SQCHnDSCmBR.DTSEL[1:0] bits to 01b.

After the above settings, send a read packet by setting the SQCHnSET0R.START bit to 1.

BTA takes place after sending the read packet. After DSI Host gives bus ownership to the peripheral, DSI Host receives a response packet. At this time, DSI Host may also receive an Acknowledge and Error Report Packet at the same time.

When Single Packet Reception is complete, the SQCHnSR.ADESFIN bit is set to 1.

The user can get a summary of the reception result by reading the RXRSSxR register with the number-x specified in the SQCHnDSCmCR.ACTCODE[7:0] bits, and can get the reception result details by reading the RXSR register.

If the response packet received is a Short Packet or Long Packet with WC = 0, the SQCHnDSCmBR.DTSEL[1:0] bits setting is not meaningful.

58.3.5.4 Reset Trigger Transmission

DSI Host can send the reset trigger in Command mode using Sequence operation.

Set the following registers before starting the sequence operation.

- SQCHnDSCmAR.NXACT[1:0] : 00b
- SQCHnDSCmCR.AUXOP : 1
- SQCHnDSCmCR.ACTCODE[7:0] : 0x00

After the above settings, send a reset trigger by setting the SQCHnSET0R.START bit to 1.

The SQCHnSR.ADESFIN bit is set to 1 when a reset trigger transmission is complete.

58.3.5.5 Execute Operation in Sequence

Explains the behavior of the descriptor in Command mode using Sequence operation.

By setting the SQCHnSET0R.START bit to 1, Sequence Operation Channel-n is enabled, and processing is performed from Descriptor-0.

Once Descriptor-m process is complete, the next process follows the value of the SQCHnDSCmAR.NXACT[1:0] bits.

When the SQCHnDSCmAR.NXACT[1:0] bits to be 00b:

After Descriptor-m processing is complete, Sequence Operation Channel-n stops.

When the SQCHnDSCmAR.NXACT[1:0] bits to be 01b:

After Descriptor-m processing is complete, Descriptor-(m + 1) processing begins.

58.3.5.6 Receiving EoTp

DSI Host can receive the End of Transmission Packet (EoTp). When DSI Host receives an EoTp, the RXSR.RXEOTP bit is set to be 1.

58.3.5.7 Acknowledge and Error Reporting Mechanism

The user can know the reliability of the serial bus by using Acknowledge and Error Reporting Mechanism.

If the peripheral detects an error on the serial bus in the peripheral-to-host transmission, the peripheral respond with Acknowledge and Error Report packet. When DSI Host receives this packet, it sets the RXRSSxR.RXAKE bit, the RXSR.RXAKE bit, and the SQCHnSR.RXAKE bit to 1. The virtual channel identifier and the error report for the latest received Acknowledge and Error Report packet is stored in the AKEPLATIR register and the accumulated information is displayed in the AKEPACMSR register.

58.3.6 Video Mode Operation

58.3.6.1 Start of Video Mode Operation

The procedure to start operating in video mode is as follows:

1. Set the video mode parameters (VMPPSETR, VMVSSETR, VMVPSETR, VMHSSETR, VMHPSETR)
2. Set the VMSET1R.DLY[11:0] bits.
3. Set the VMSET0R.VSTART bit to 1 with other VMSET0R setting values.
4. Wait until the VMSR.VIRDY bit is set to 1.
5. Set the BG_EN.VEN bit and GB_EN.EN bit to 1 at the same time for the GLCDC. By this setting, the output of the vertical and horizontal synchronization signals VSYNC, HSYNC, and DE, and the image data (RGB) from the GLCDC starts. (See section 58.5.1. GLCDC register setting when using Video Mode operation)

58.3.6.2 End of Video Mode Operation

The procedure to stop operating in video mode is as follows:

1. Set the VMSET0R.VSTOP bit to 1 to request that the video mode operation stops.
   When the DSI Host detects the start of the first frame after setting the VMSET0R.VSTOP bit to 1, it stops sending video mode packets and sets the VMSR.STOP flag to 1 and VMSR.RUNNING flag to 0.
2. Stop the GLCDC after checking that the VMSR.STOP flag is set to 1.
3. After Stopping the GLCDC, wait for the LINKSR.HSBUSY flag is set to 0.

58.3.6.3 Burst Mode

The DSI Host supports the burst mode. The burst mode is a mode for time-compression of pixel data.

Make the bandwidth of D-PHY wider than the bandwidth of the data stream from GLCDC to time compress the pixel data.

58.3.6.4 Non-Burst Mode

The DSI Host supports the non-burst mode.

Equalize the bandwidth of D-PHY with the bandwidth of the data stream from GLCDC.

58.3.6.5 Selection of Blanking Packets or LP-11 for HSA, HBP, and HFP

When both of the following two conditions are satisfied, the DSI Host automatically transitions the Data Lane from HS mode to LP mode during the HSA, HBP, and HFP periods respectively.

If the DSI Host does not transition from HS mode to LP mode, it sends the blanking packets with the same VC (Virtual Channel) as the VC of the previous packet to maintain HS mode.

1. There is time to transition from HS mode to LP mode and back to HS mode before the next HS transfer starts. (The minimum period required to transition to LP mode is specified by the LPTRNSTSETR.GOLPBKT[9:0] bits)
2. The following registers are set for each period of HSA, HBP, and HFP.
   HSA : VMSET0R.HSANOLP = 0
   HBP : VMSET0R.HBPNOLP = 0
   HFP : VMSET0R.HFPNOLP = 0

58.3.6.6 Non video packet action in Video Mode Operation

The DSI Host can send/receive packets using Sequence Operation (See section 58.3.5.2. Single Packet Transmission and section 58.3.5.3. Single Packet Reception) during Video Mode Operation.

Make sure that all Sequence Operation Channel's operations are finished before starting the Vide Mode Operation (before starting the process described in section 58.3.6.1. Start of Video Mode Operation).

During Video Mode Operation, Sequence Operation is used to send and receive packets after the HSE packet on the first horizontal line if the HSE transfer is permitted (VMPPSETR.TXESYNC = 1) and after the VSS packet if the HSE transfer is not permitted (VMPPSETR.TXESYNC = 0).

Also, sending packets using Sequence Operation is only possible with HS mode (SQCHnDSCmAR.SPD = 0). Transmission in LP mode (SQCHnDSCmAR.SPD = 1) is not supported.

Send and receive packets so that they are completed within the BLLP period of the first horizontal line.

Do not use any other functions of Sequence Operation.

If Sequence Operation is executed during Video Mode Operation, stop Video Mode Operation (section 58.3.6.2. End of Video Mode Operation) after confirming that Sequence Operation is completed.

The period of Video Mode Operation begins the moment the VMSET0R.VSTART bit is set to 1.

Note: The VMSR.RUNNING bit, which indicates that Vide Mode is operating, is reflected after a delay.

58.3.6.7 Delay Adjustment for Video Mode Operation

The DSI Host has a 4 KB video buffer to input video stream from GLCDC.

It is necessary to use the VMSET1R.DLY[11:0] bits to adjust the delay time for reading from the video buffer to prevent the video buffer from overflowing or underflowing during Video Mode operation.

The delay time is derived from the following formula:

Delay time [µs] = VMSET1R.DLY[11:0] × (period of HS serial UI) [µs] × 32

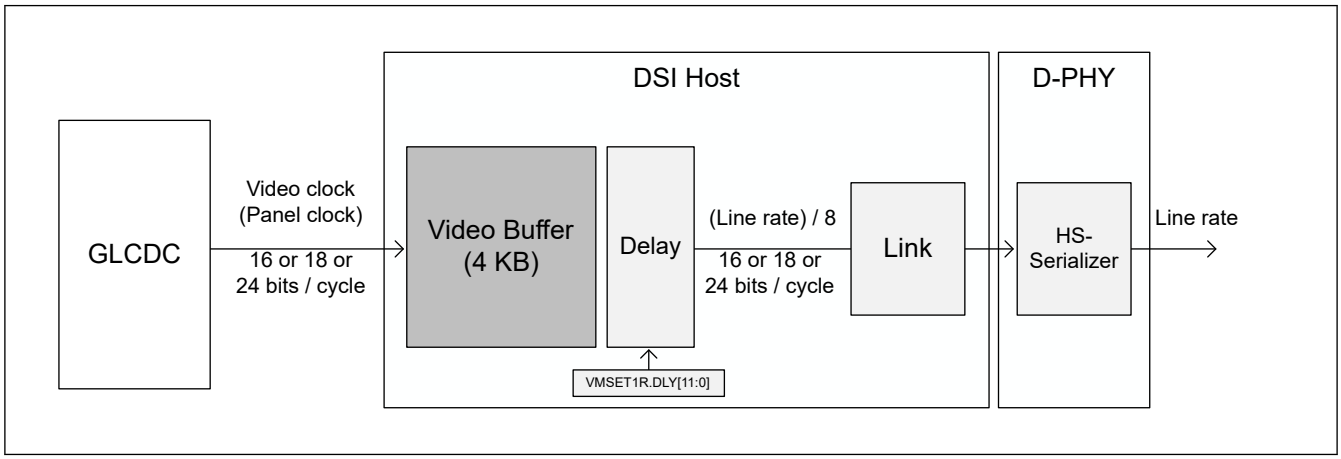


Figure 58.5 Video Buffer and Delay Control

58.3.7 Transition To/From ULPS

This section describes the sequence of transition to or return from ULPS.

The Clock Lane and Data Lanes (Lane-0 and Lane-1) can be transitioned to/from ULPS independently.

When the TXSETR.CLEN bit is set to 1, the Clock Lane can transition to ULPS. It is prohibited to change the TXSETR.CLEN bit to 0 while the Clock Lane is in ULPS.

When the TXSETR.DLEN bit is set to 1, the Data Lanes can transition to ULPS. It is prohibited to change the TXSETR.DLEN bit to 0 while the Data Lanes are in ULPS.

The transition to/from ULPS of the Data Lanes is performed simultaneously for the Data Lane specified by the TXSETR.NUMLANE[1:0] bits.

Only a specific Data Lane cannot be transitioned to ULPS.

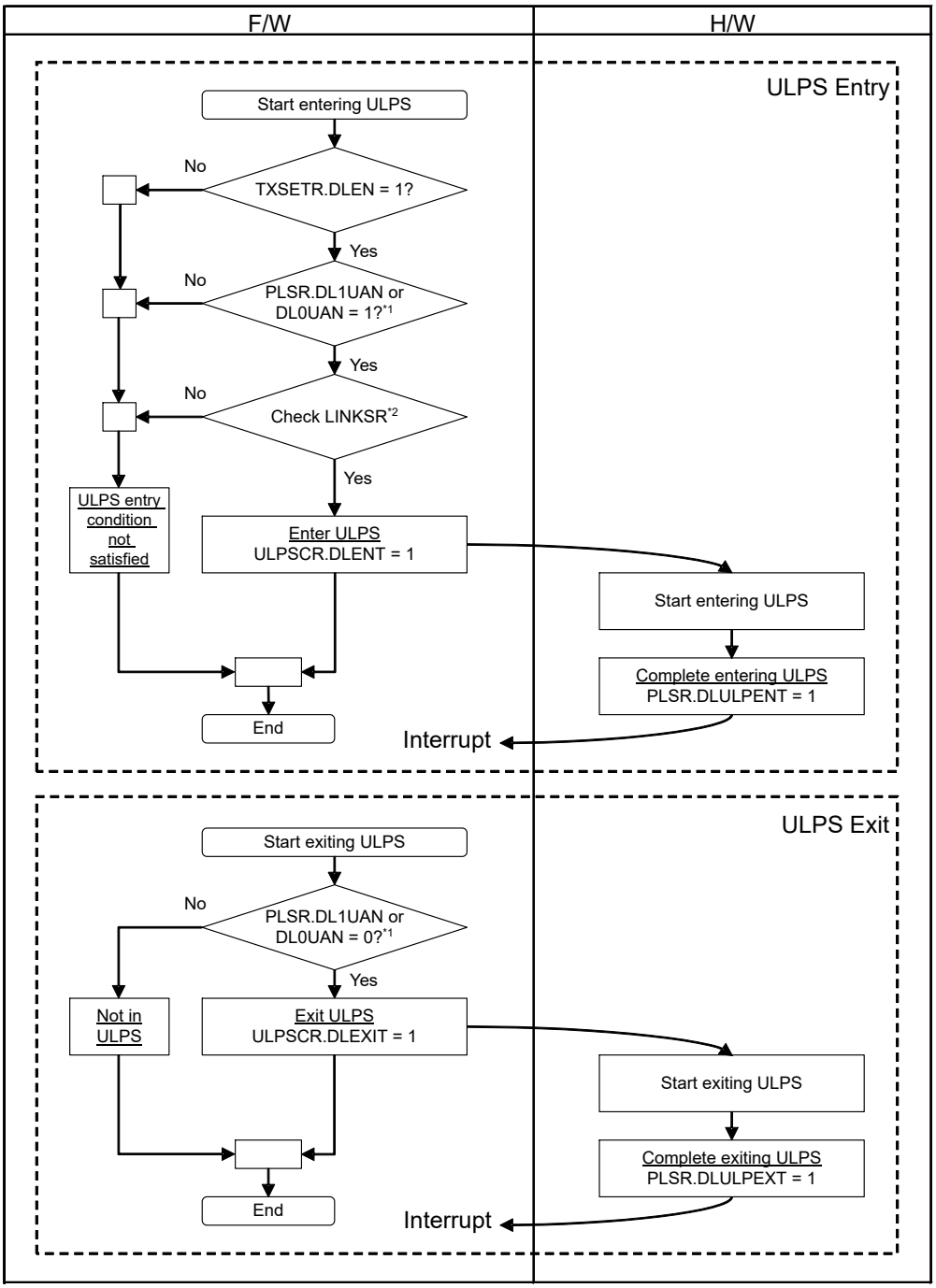
When Data Lanes enter ULPS, another operation*1 which uses Data Lanes cannot be performed until exit from ULPS.

When Clock Lane enters ULPS, another operation*2 which uses Clock Lane cannot be performed until exit from ULPS.

Note that the High-Speed Transmit Word clock (HS mode clock) or Escape More Transmit clock (LP mode clock) might be stopped while Data Lanes and Clock Lane are in ULPS depending on the specifications of the system incorporating the DSI-Tx module.

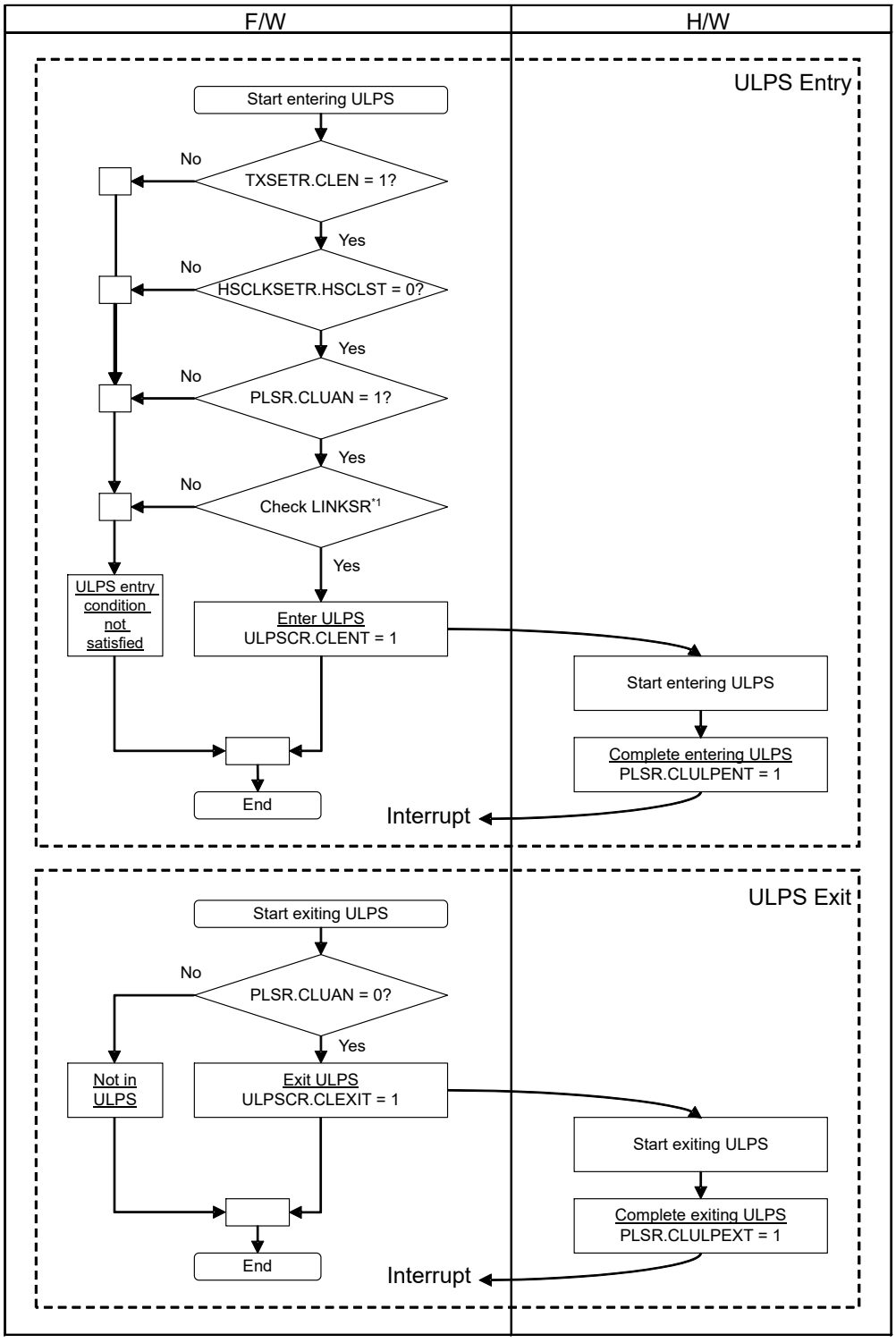
To perform software reset processing (section1.3.2) in this state, restart the stopped clock before executing the software reset flow.

Note 1. Sequence Operation, Video mode Operation.
Note 2. HS Sequence Operation, Video mode Operation.



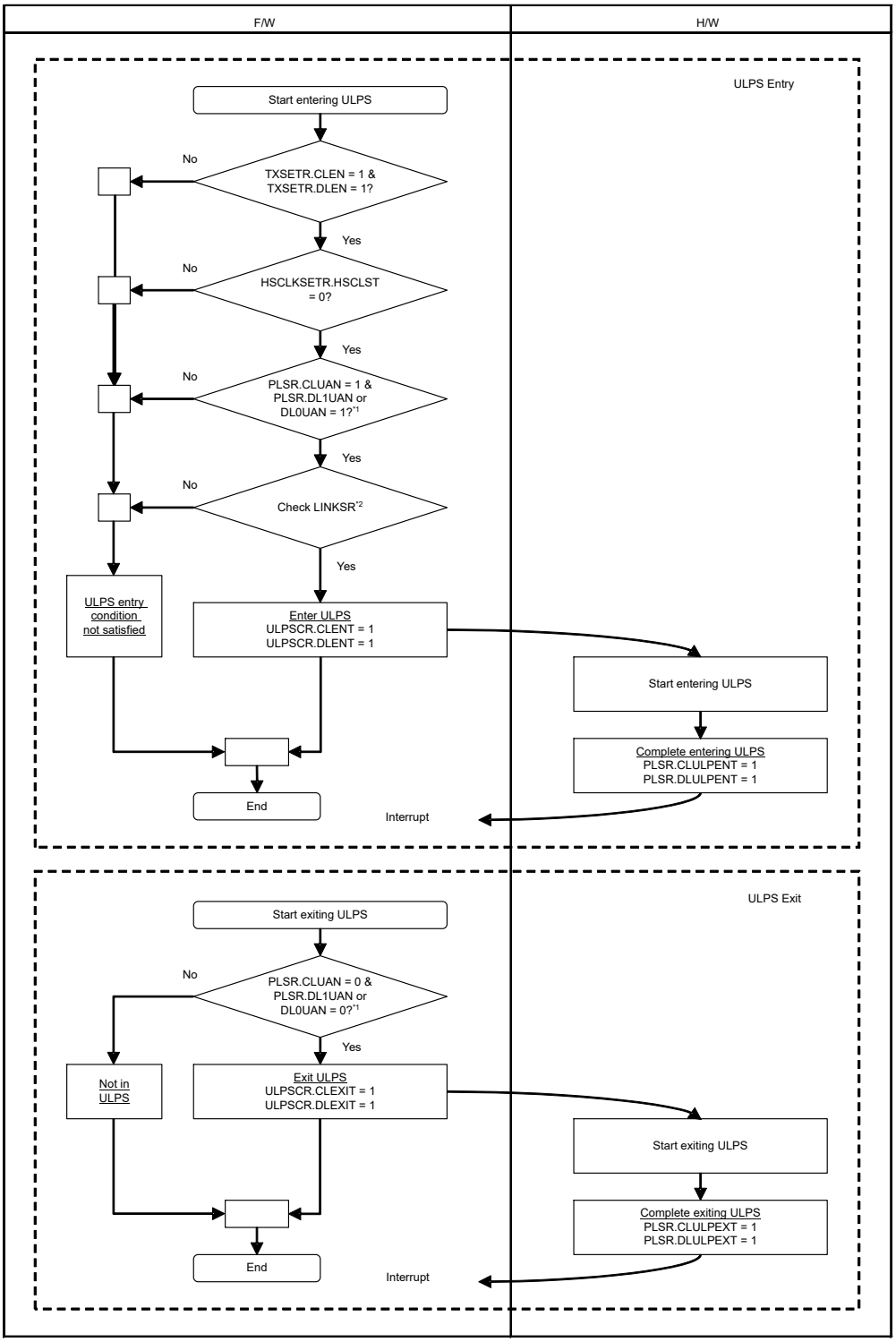
Note 1. Check all enabled data lane which selected by TXSETR.NUMLANE
Note 2. (SQ0RUN = 0, SQ1RUN = 0, VRUN = 0, HSBUSY = 0 and LPBUSY = 0)?

Figure 58.6 ULPS Entry/Exit (Data Lanes)



Note 1. (SQ1RUN = 0, VRUN = 0 and HSBUSY = 0)?

Figure 58.7 ULPS Entry/Exit (Clock Lane)



Note 1. Check all enabled Data Lanes which are selected by TXSETR.NUMLANE
Note 2. (SQ0RUN = 0, SQ1RUN = 0, VRUN = 0, HSBUSY = 0 and LPBUSY = 0)?

Figure 58.8 ULPS Entry/Exit (Clock and Data Lanes)

58.3.8 Error Handling

58.3.8.1 Error Detection by Peripheral

The peripheral notifies the DSI Host of errors it detects with the Acknowledge and Error Report packet.

When the DSI Host receives an Acknowledge and Error Report packet, it displays the latest result in the AKEPLATIR register and the accumulated result in the AKEPACMSR register.

If the user receives an Acknowledge and Error Report packet, review and change various settings as necessary.

58.3.8.2 Error Detection in D-PHY

When the D-PHY detects an error, the DSI Host sets the corresponding register flag.

The following errors are supported by the DSI Host. Refer to Annex A Logical PHY-Protocol Interface Description of MIPI Alliance Specification for D-PHY Version 2.1, Annex A Logical PHY-Protocol Interface Description.

- ErrEsc
- ErrSyncEsc
- ErrControl
- ErrContentionLP0
- ErrContentionLP1

If this error is detected, perform Software Reset (See section 58.3.2. Software Reset) and Reset Trigger Transmission (See section 58.3.5.4. Reset Trigger Transmission) as necessary.

When the ErrContentionLP0/ ErrContentionLP1 error occurs, be sure to execute Software Reset.

In this case, after writing 1 to the RSTCR.FTXSTP bit, wait for a time (equivalent to LRX-H_TO) after which Contention is resolved, and then write 0 to the RSTCR.FTXSTP bit. Also, if necessary, execute Reset Trigger Transmission.

For more information on recovery from contention, refer to 7.2 Contention Detection and Recovery in the MIPI Alliance Specification for Display Serial Interface 2 Version 1.1.

58.3.8.3 Timeout Error

The DSI Host has the timers defined in 7.2.2., Contention Recovery Using Timers in the DSI standard and additional timers.

Timers defined in MIPI DSI Specification

- HS TX Timeout Error (HTX_TO)
- LP-RX Host Processor Timeout Error (LRX-H_TO)
- Turnaround Acknowledge Timeout Error (TA_TO)

Additional Timers

- Peripheral Response Timeout Error

HS TX Timeout (HTX_TO) Error

If this error occurs, execute Software Reset (see section 58.3.2. Software Reset). In addition, if necessary, execute Reset Trigger Transmission (see section 58.3.5.4. Reset Trigger Transmission).

If this error is detected, the setting value of the HSTXTOSETR register may be too small.

Set the HSTXTOSETR register to a value greater than the HS transmission period.

LP-RX Host Processor Timeout (LRX-H_TO) Error

If this error occurs, execute Software Reset (see section 58.3.2. Software Reset). In addition, if necessary, execute Reset Trigger Transmission (see section 58.3.5.4. Reset Trigger Transmission).

If this error is detected, the setting value of the LRXHTOSETR register may be too small.

It is possible to receive multiple packets in one reception, such as a response packet and an Acknowledge and Error Report packet.

Set the LRXHTOSETR register to a value greater than the receive period.

Turnaround Acknowledge Timeout (TA_TO) Error

If this error occurs, if necessary, execute Software Reset (see section 58.3.2. Software Reset) and Reset Trigger Transmission (see section 58.3.5.4. Reset Trigger Transmission).

If this error is detected, the setting value of the TATOSETR register may be too small.

Check the specifications of the Peripheral, if the Peripheral does not support bidirectional communication, this error will also occur.

Peripheral Response Timeout Error

Refer to the description of Peripheral Response Timeout Error in section 58.3.8.4. Receive Related Error in Sequence Operation.

58.3.8.4 Receive Related Error in Sequence Operation

This section describes the errors in received packets detected by the DSI Host.

A maximum of one error is detected per packet.

Note: A single-bit ECC Error may occur at the same time as other errors.

Errors are determined in the following order:

1. Peripheral Response Timeout Error
2. Malformed Packet Error
3. ECC Error
4. Unexpected Packet Error
5. Word Count Error
6. CRC Error
7. No Response Error

Peripheral Response Timeout Error

If no packets or triggers are received within the time set in the register after the bus ownership has been changed from Host to Peripheral by the BTA, the RXSR.PRTOERR flag is set to 1.

When this error is detected, no packets or triggers will be received until the bus ownership shifts from Peripheral to Host.

Timeout Time Setting Registers

- Peripheral Response Timeout value for BTA is specified by the PRESPTOBTASETR register.
- Peripheral Response Timeout value for LP Read with BTA is specified by the PRESPTOLPSETR.LPRTO[15:0] bits.
- Peripheral Response Timeout value for LP Write with BTA is specified by the PRESPTOLPSETR.LPWTO[15:0] bits.
- Peripheral Response Timeout value for HS Read with BTA is specified by the PRESPTOHSSETR.HSRTO[15:0] bits.
- Peripheral Response Timeout value for HS Write with BTA is specified by the PRESPTOHSSETR.HSWTO[15:0] bits.

If this error is detected, the register setting value may be too small, so check the Peripheral specifications.

If this error occurs, perform the necessary processing, such as resending the previous command.

Malformed Packet Error

When a packet of less than 4 bytes is received, the RXSR.MLFERR flag is set to 1. Since there is an error in the received data, perform the necessary processing, such as resending the previous command.

ECC Error

ECC generation is mandatory for Peripherals that comply with the MIPI DSI-2 v1.1 standard. However, ECC generation is optional for "earlier revision of DSI Peripherals", so the DSI Host can control to enable the ECC check by the DSISETR.ECCEN bit.

When connecting to a Peripheral that does not support ECC generation, disable the ECC check by setting the DSISETR.ECCEN bit to 0.

Single Bit ECC Error

When a single-bit ECC error is detected, the RXSR.ECCERRS flag is set to 1, and single-bit ECC error correction is performed.

The DSI Host automatically corrects the error and restores the normal packet header, so that the subsequent packet reception process can continue.

Multi Bit ECC Error

When a multi-bit ECC error is detected, the RXSR.ECCERRM flag is set to 1, and the received packet is discarded.

When this error occurs, perform the necessary processing, such as resending the previous command.

Unexpected Packet Error

If an Unexpected Packet is received, the RXSR.UNEXERR flag is set to 1.

When this error occurs, the packet header is stored in the RXRSSxR register, and subsequent reception processing is stopped until the bus ownership shifts from Peripheral to Host.

The DSI Host judges the following packets as Unexpected Packet.

- Reception of a packet whose Data Type field in the packet header is defined as "Reserved" in Table 23 Data Types for Peripheral-Sourced Packets of the MIPI DSI-2 v1.1 standard.
- Reception of a packet other than Acknowledge and Error Report packet and EoTp after sending a BTA request when the SQCHnDSCmAR.BTA[1:0] bits are set to 01b.
- Reception of a packet other than Acknowledge and Error Report packet and EoTp after sending a BTA request when the SQCHnDSCmAR.BTA[1:0] bits are set to 11b.
- After receiving response packet after sending a read request following BTA when the SQCHnDSCmAR.BTA[1:0] bits are set to 10b, a response packet is received again.

Word Count Error

When the payload data of a long packet is shorter than the WC (Word Count) value, the RXSR.WCERR flag is set to 1.

When this error occurs, there is an error in the payload data length, but the payload data is not discarded, and the packet header is stored in the RXRSSxR register.

Since there is an error in the received data, perform the necessary processing, such as resending the previous command.

CRC Error

For the case of connecting to a Peripheral that does not support CRC generation, the DSI Host can select whether to enable or disable CRC checking for each Virtual Channel using the DSISETR.VCxCRCEN (x = 0 to 3) bit.

When connecting to a Peripheral that does not support CRC generation, disable the CRC check by setting the corresponding DSISETR.VCxCRCEN bit to 0.

When this error occurs, there is an error in the payload data, but the payload data is not discarded, and the packet header is stored in the RXRSSxR register.

If this error occurs, take the necessary action, such as resending the previous command.

No Response Error

After bus ownership is transferred from the host to the peripheral by BTA, if it is transferred back to the host without neither packet nor trigger being received, the RXSR.NORESERR flag is set to 1.

If this error occurs, take the necessary action such as resending the command.

If the error recurs, perform Software Reset (See section 58.3.2. Software Reset) or Reset Trigger Transmission (See section 58.3.5.4. Reset Trigger Transmission) as necessary.

Receive Long Packet Data Payload Related Error in Sequence Operation

Maximum Return Packet Size Error

If the WC (Word Count) value of the received long packet is greater than the value set in the DSISETR.MRPSZ[15:0] bits, the RXSR.RSIZEERR flag is set to 1.

When this error occurs, the packet header is stored in the RXRSSxR register and the payload data is discarded.

The payload data is discarded, but errors related to the payload data reception (WC Error, CRC Error) are detected.

If this error occurs, set the DSISETR.MRPSZ[15:0] bits to a value greater than or equal to the value set in the *Set Max Return Packet Size* command which limits the size of returning packets.

If this error occurs despite the correct settings, consider setting a larger value for the DSISETR.MRPSZ[15:0] bits, as Peripheral may be sending back large packets in violation of the DSI standard.

Internal Bus Error

If an error is detected in Internal AXI Bus Write, the RXSR.IBERR flag is set to 1.

Review the settings of the SQCHnDSCmDR.LADDR[31:0] bits and the DSISETR.MRPSZ[15:0] bits.

Receive Buffer Overflow Error

The RXSR.RXOVFERR flag is set to 1 when the following Receive Buffer Overflow Error is detected.

- In the case of using Packet Payload Data Register (SQCHnDSCmBR.DTSEL = 00b)
  The length of the received payload data is longer than 16 Bytes.
- In the case of using Memory space (SQCHnDSCmBR.DTSEL = 01b)
  Memory space overflows in LP-RX.

If this error occurs, do the following:

- In the case of using Packet Payload Data Register (SQCHnDSCmBR.DTSEL = 00b)
  - Change to use Memory space.
  - Change the maximum return packet size of peripheral device so that it does not exceed 16 bytes.
- In the case of using Memory space (SQCHnDSCmBR.DTSEL = 01b)
  - Reduce the maximum return packet size of peripheral device.

58.3.8.5 Transfer Packet Related Error in Sequence Operation

Internal Bus Error

When an error is detected in Internal AXI Bus Read, the SQCHnSR.TXIBERR flag is set to 1.

Review the settings of the SQCHnDSCmAR.DATA0[7:0] bits, the SQCHnDSCmAR.DATA1[7:0] bits, and the SQCHnDSCmDR.LADDR[31:0] bits.

Packet Size Error

If the packet of Sequence Operation is too large to be transmitted within the BLLP period of the first horizontal line of Video Mode operation, the SQCH1SR.SIZEERR bit will be set to 1.

When this error occurs, Sequence Operation packets will not be sent until the Video Mode operation is completed.

58.3.8.6 Error in Video Mode operation

Video Mode Buffer Overflow Error

The VMSR.VBUFOVF flag will be set to 1 if the video buffer overflows during the Video Mode operation.

If this error occurs, perform Software Reset (See section 58.3.2. Software Reset).

Also, take necessary measures such as increasing the line rate of the D-PHY or decreasing the value of the VMSET1R.DLY[11:0] bits.

In addition, perform Reset Trigger Transmission (See section 58.3.5.4. Reset Trigger Transmission) if necessary.

Video Mode Buffer Underflow Error

The VMSR.VBUFUDF flag will be set to 1 if the video buffer underflows during the Video Mode operation.

If this error occurs, perform Software Reset (See section 58.3.2. Software Reset).

Also, take necessary measures such as decreasing the line rate of the D-PHY or increasing the value of the VMSET1R.DLY[11:0] bits.

In addition, perform Reset Trigger Transmission (See section 58.3.5.4. Reset Trigger Transmission) if necessary.

Timing Error

The VMSR.TIMERR flag is set to 1 if a video packet (including Sync Event packet) is not sent at the intended timing during Video Mode operation.

If this error occurs, perform Software Reset (See section 58.3.2. Software Reset).

Also, take necessary measures such as reviewing the setting value of the LPTRNSTSETR.GOLPBKT[9:0] bits.

In addition, perform Reset Trigger Transmission (See section 58.3.5.4. Reset Trigger Transmission) if necessary.

58.4 Interface

58.4.1 Video mode Operation

58.4.1.1 Video mode Timing

Video mode Timing and Video mode Parameters are described in Figure 58.9 and Table 58.3.

In Table 58.3, the actual period of the following parameters depend on the specification of the D-PHY which is connected to the DSI-Tx module.

- Period for HS-LP-HS transition, period for HS-LP transition, and period for LP-HS transition.
- Period for BTA[host->peripheral].

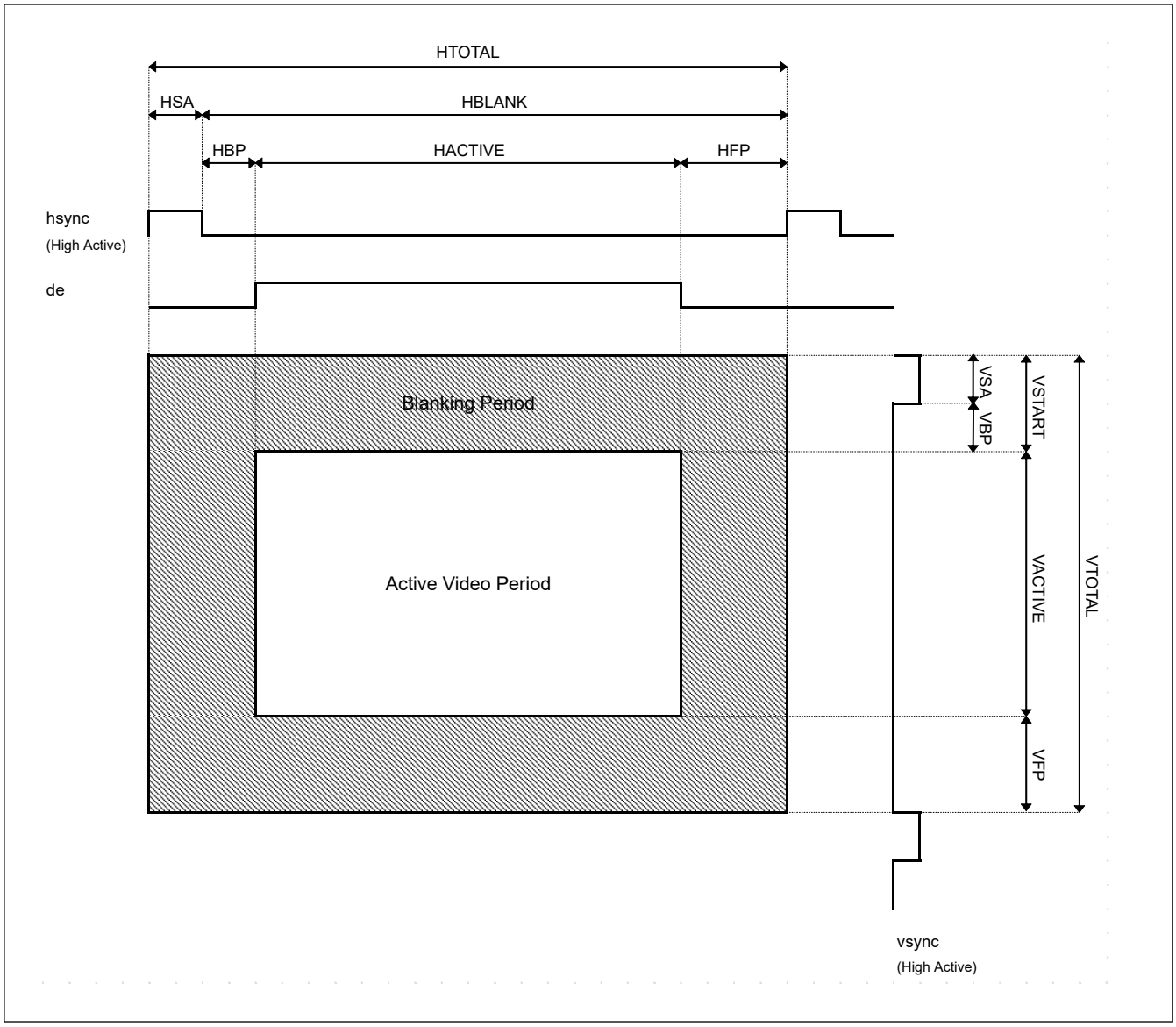


Figure 58.9 Video mode Timing

Table 58.3 Configurations for Video mode Interface (1 of 4)

Common Rules for All Modes		
Parameter	**Description**	
All	All parameters related to Input Video should be set to 1 or higher. Examples of the parameters: HSA, HBP, HACTIVE, HFP, VSA, VBP, VACTIVE, VFP	
Parameter	**VMPPSETR.DT**	**VMHSSETR.HACT**
HACTIVE	0x1E: Packed Pixel Stream, 18-bit RGB	This value should be a multiple of 4.
	0x2C: Packed Pixel Stream, 16-bit YCbCr 4: 2: 2 0x0C: Loosely Packed Pixel Stream, 20-bit YCbCr 4: 2: 2 0x1C: Packed Pixel Stream, 24-bit YCbCr 4: 2: 2	This value should be a multiple of 2.
	0x0E: Packed Pixel Stream, 16-bit RGB 0x2E: Loosely Packed Pixel Stream, 18-bit RGB 0x3E: Packed Pixel Stream, 24-bit RGB	No limitation
HFP	Rounddown (HFP*BPP/ 8, 0) >= 12, (12 = means Pixel Packet Header & footer 6 + Blanking Packet Header & footer 6)	

Table 58.3 Configurations for Video mode Interface (2 of 4)

Common Rules for All Modes		
Parameter	**Description**	
Parameter	**VMPPSETR.DT**	**VMHSSETR.HSA + VMHPSETR.HBP + VMHSSETR.HACT + VMHPSETR.HFP**
HTOTAL	0x1E: Packed Pixel Stream, 18-bit RGB	This value should be a multiple of 4.
	0x2E: Loosely Packed Pixel Stream, 18-bit RGB 0x0C: Loosely Packed Pixel Stream, 20-bit YCbCr 4: 2: 2 0x1C: Packed Pixel Stream, 24-bit YCbCr 4: 2: 2 0x2C: Packed Pixel Stream, 16-bit YCbCr 4: 2: 2 0x0E: Packed Pixel Stream, 16-bit RGB 0x3E: Packed Pixel Stream, 24-bit RGB	No limitation.

Table 58.3 Configurations for Video mode Interface (3 of 4)

Additional Rules for Non-burst Sync Pulse Mode	
Parameter	**Description**
Sequence Operation Packet	When a packet is transmitted by Sequence Operation during Video mode Operation (VMSR.RUNNING = 1), the period of (HBP+HACTIVE+HFP) should be longer than the sum of the following: ● Period for the Tx Sync Event packet (4 Bytes) ● Period for the packet sent by Sequence Operation ● Period for the Header and CRC of a Blanking packet (6 Bytes)
HBP + HACTIVE + HFP	HS-LP-HS transition is caused within the period between the current Sync Event packet and the next Sync Event packet, so the period of (HBP+HACTIVE+HFP) should be as follows. (HBP+HACTIVE+HFP) > (Period for the Tx Sync Event packet) + (Period for HS-LP-HS transition)
HSA	Blanking packet transmission is caused within the period between the current Sync Event packet and the next Sync Event packet, so the period of HSA should be as follows. (HSA*BPP)/8 >= 10 Note that HSA should not be less than the minimum value of the sum of (Period for the Tx Sync Event packet) + (Period for the Tx Blanking packet).
HBP	Blanking packet transmission is caused within the period between the current Sync Event packet and the next Sync Event packet, so the period of HBP should be as follows. (HBP*BPP)/8 >= 10 Note that HBP should not be less than the minimum value of the sum of (Period for the Tx Sync Event packet) + (Period for the Tx Blanking packet).

Table 58.3 Configurations for Video mode Interface (4 of 4)

Additional Rules for Non-burst Sync Event Mode and Burst Mode	
Parameter	**Description**
Sequence Operation Packet	When a packet is transmitted by Sequence Operation during Video mode Operation (VMSR.RUNNING = 1), the period of HTOTAL should be longer than the sum of the following: ● Period for the Tx Sync Event packet (4 Bytes) ● Period for the packet sent by Sequence Operation ● Period for the Header and CRC of a Blanking packet (6 Bytes)
HTOTAL	HS-LP-HS transition is caused within the period between the current Sync Event packet and the next Sync Event packet, so the period of HTOTAL should be as follows. HTOTAL > (Period for the Tx Sync Event packet) + (Period for HS-LP-HS transition)
HAS + HBP	Blanking packet transmission is caused within the period between the current Sync Event packet and the next Sync Event packet, so the period of (HSA+HBP) should be as follows. ((HSA+HBP)*BPP)/8 >= 10 Note that HSA+HBP should not be less than the minimum value of the sum of (Period for the Tx Sync Event packet) + (Period for the Tx Blanking packet). The minimum HSA+HBP value currently recognized is 8+8 and it is 32 even in the case of 16-bit RGB as the worst condition, which is not subject to this restriction. Therefore, this restriction is not presented up to the limit. Even if this condition is not satisfied, the result is Pixel-Packet is only slightly delayed.

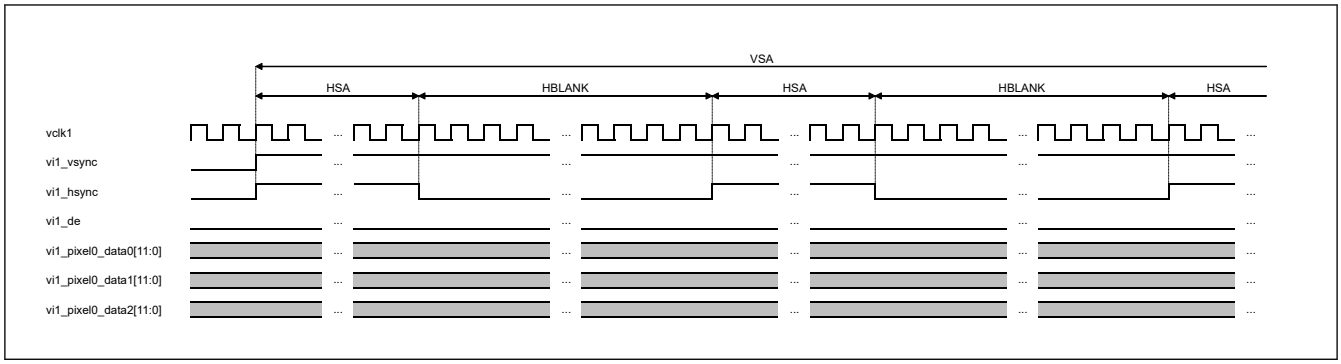


Figure 58.10 VSA Start (and VFP End)

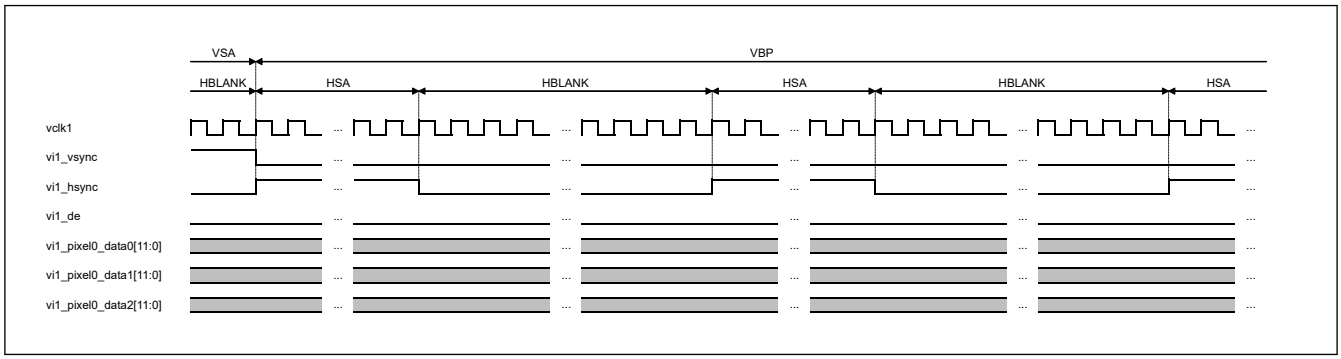


Figure 58.11 VSA End and VBP Start

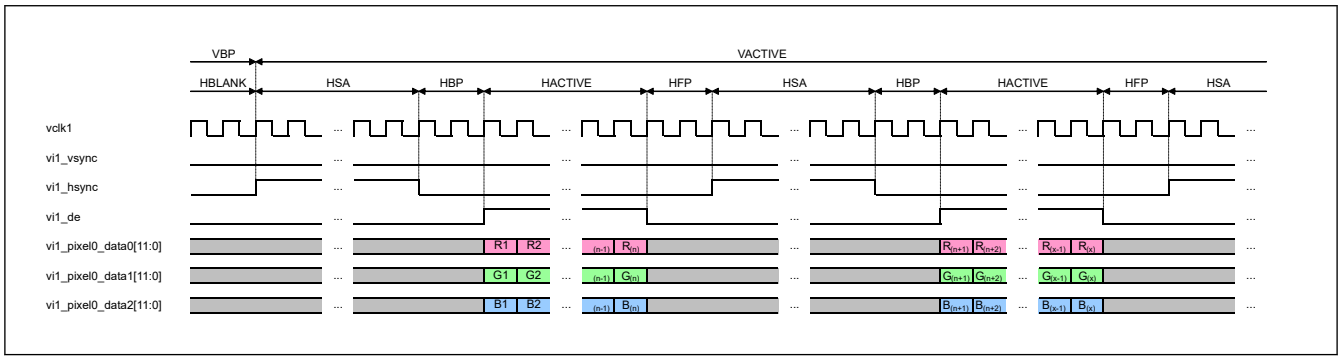


Figure 58.12 VBP End and VACTIVE Start (RGB)

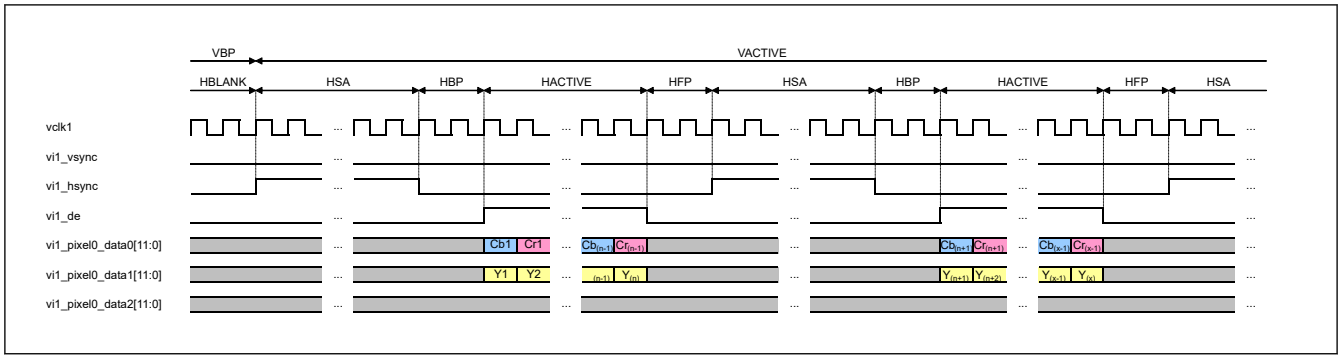


Figure 58.13 VBP End and VACTIVE Start (YCbCr 4: 2: 2)

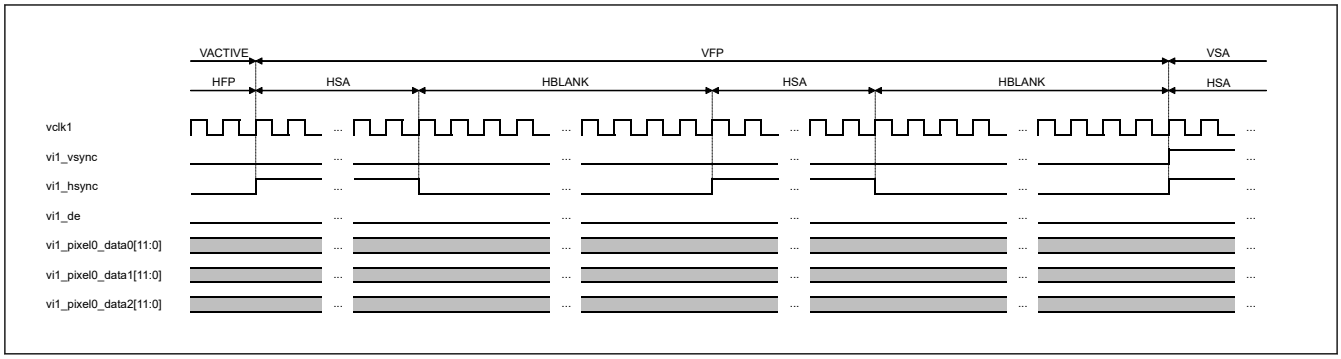


Figure 58.14 VACTIVE End and VFP Start

58.5 Usage Notes

58.5.1 GLCDC register setting when using Video Mode operation

When using Video Mode operation, set the GLCDC registers as shown below to specify the video signals (VSYNC, HSYNC, DE, RGB) and supply a video clock.

Common

- SYSCNT_PANEL_CLK.CLKEN = 1 (Enable Panel Clock output)
- SYSCNT_PANEL_CLK.PIXSEL = 0 (Parallel RGB)
- TCON_STVA1 = Value to specify VSYNC signal
- TCON_STHA1 = Value to specify HSYNC signal
- TCON_STVB1 = Value to specify VE signal
- TCON_STHB1 = Value to specify HE signal
- TCON_STVA2.SEL[2:0] = 000b (Assign STVA signal to LCD_TCON0 output)
- TCON_STVB2.SEL[2:0] = 010b (Assign STHA signal to LCD_TCON1 output)
- TCON_STHA2.SEL[2:0] = 111b (Assign DE signal to LCD_TCON2 output)
- OUT_SET.FRQSEL[1:0] = 00b (Parallel RGB)
- OUT_SET.SWAPON = 0 (RGB order)
- OUT_SET.ENDIANON = 0 (little endian)

In the case of RGB888 (24-bit) format

- OUT_SET.FORMAT[1:0] = 00b (RGB888 format)

In the case of RGB666 (18-bit) format

- OUT_SET.FORMAT[1:0] = 01b (RGB666 format)

In the case of RGB565 (16-bit) format

- OUT_SET.FORMAT[1:0] = 10b (RGB565 format)

58.5.2 Module-stop function

MIPI DSI operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The MIPI DSI module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

59. Internal Voltage Regulator

59.1 Overview

The MCU supports two power supply providing for core voltage (VDD):

- Switching regulator (DCDC)
- External power supply from external power device (External VDD)

Table 59.1 shows the specifications of Internal voltage regulator.

DCDC or External VDD is selected with the OFS setting.

- When OFS2.DCDCEN is 1, DCDC is selected (DCDC mode)
- When OFS2.DCDCEN is 0, External VDD is selected (External VDD mode)

Table 59.1 Power supply selection

OFS setting (OFS2.DCDCEN)	Power mode
1	DCDC mode
0	External VDD mode

59.2 Operation

59.2.1 DCDC mode

Table 59.2 lists the DCDC mode pin settings, and Figure 59.1 shows the DCDC mode settings.

In DCDC mode, VDD is supplied from VCL through VLO output and an external inductor and capacitor.

Table 59.2 DCDC mode pin settings

Pins	Setting descriptions
All VCC and VCC2 pins	• Connect each pin to the system power supply. • Connect each pin to VSS through a 0.1 µF multilayer ceramic capacitor. Place the capacitor close to the pin.
VCC_DCDC pins	• Connect the pin to the system power supply. • Connect the pin to VSS_DCDC through a 22 µF and 0.1-µF multilayer ceramic capacitor in parallel. Place the capacitor close to the pin.
VCL pins	• Connect each pin to VSS through a 0.22 µF multilayer ceramic capacitor. Place the capacitor close to the pin. • Connect each pin to an external inductor and capacitor. Place the inductor and capacitor close to the pin.(see Figure 59.1.)
All VLO pins	Connect each pin to an external inductor and capacitor. Place the 2.2-µH inductor and 47 µF capacitor close to the pin. This capacitor must be grounded to VSS_DCDC

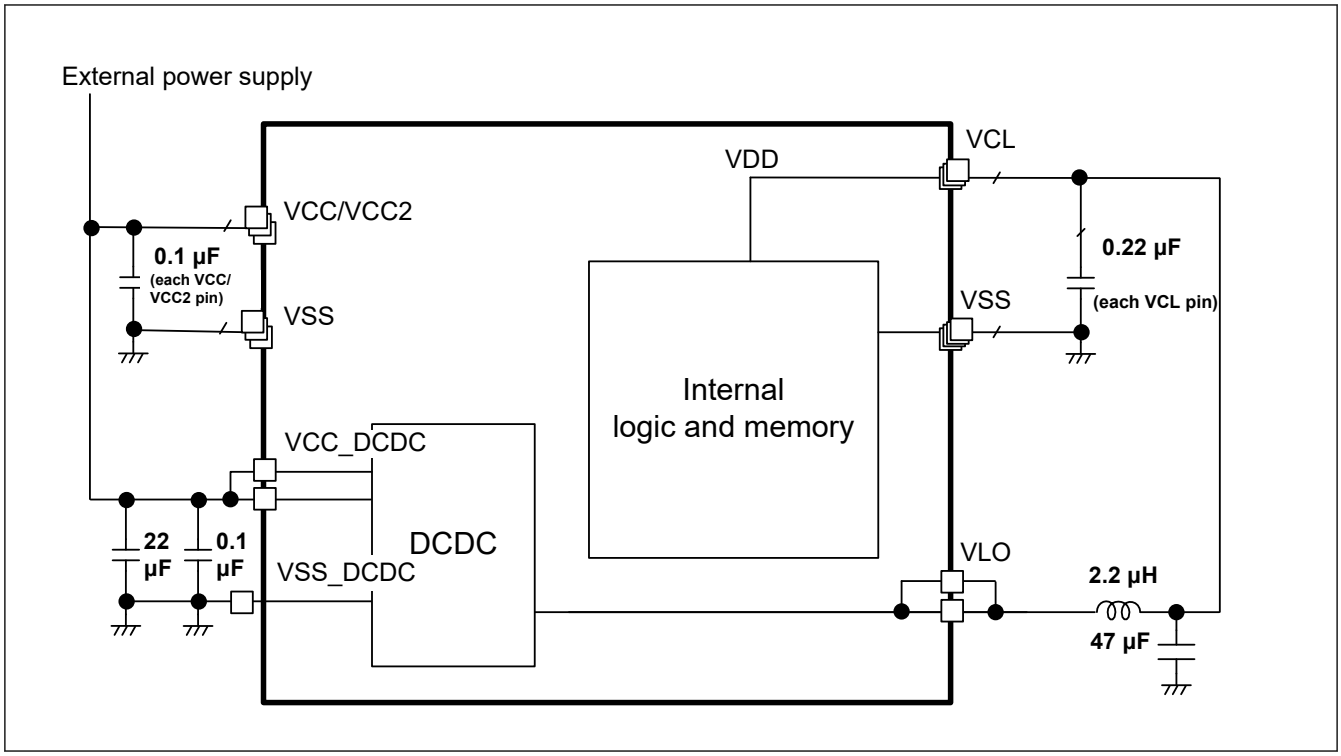


Figure 59.1 DCDC mode setting

59.2.2 External VDD mode

Table 59.3 lists the external VDD mode pin settings, and Figure 59.2 shows the external VDD mode settings.

VDD is supplied from VCL pins.

Note: Low-speed mode, Software Standby mode, Deep Software Standby mode 1, 2, 3 and Battery Backup Function are not supported in this mode.

Table 59.3 Setting Descriptions for external VDD mode

Pins	Setting descriptions
All VCC and VCC2 pins	• Connect each pin to the system power supply device for VCC. • Connect each pin to VSS through a 0.1 µF multilayer ceramic capacitor. Place the capacitor close to the pin.
VCC_DCDC pins	• Connect the pin to VSS_DCDC through 0.1 µF multilayer ceramic capacitor. Place the capacitor close to the pin.
VCL pins	• Connect each pin to the system power supply device for VDD. • Connect each pin to VSS through a 0.22 µF multilayer ceramic capacitor. Place the capacitor close to the pin.
All VLO pins	Keep pin open

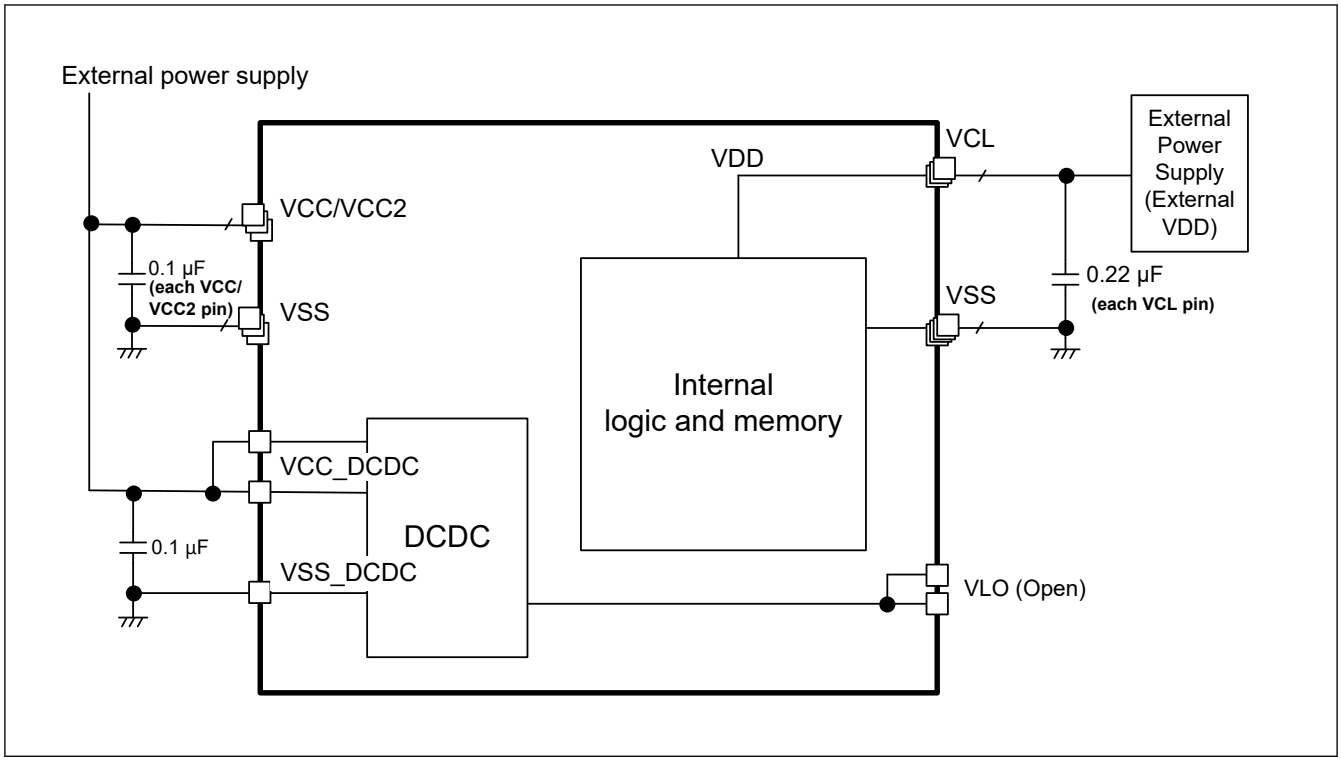


Figure 59.2 External VDD mode setting

In order to startup the device properly, use either of following procedure.

- VCL voltage must be raised 500 µs after VCC voltage is raised to minimum VCC voltage.
- Raise VCC voltage with RES pin is low, and release RES pin after 200 µs when VCL voltage is raised.

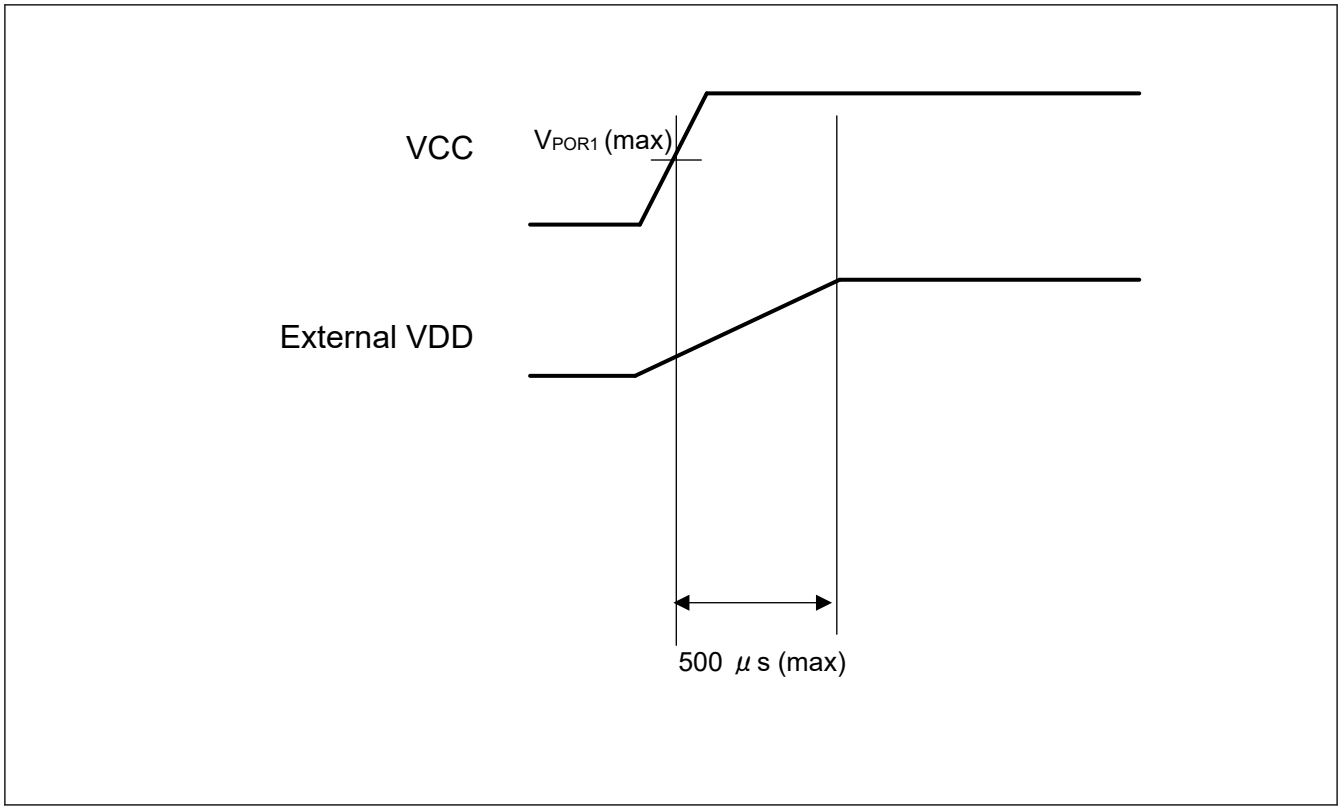
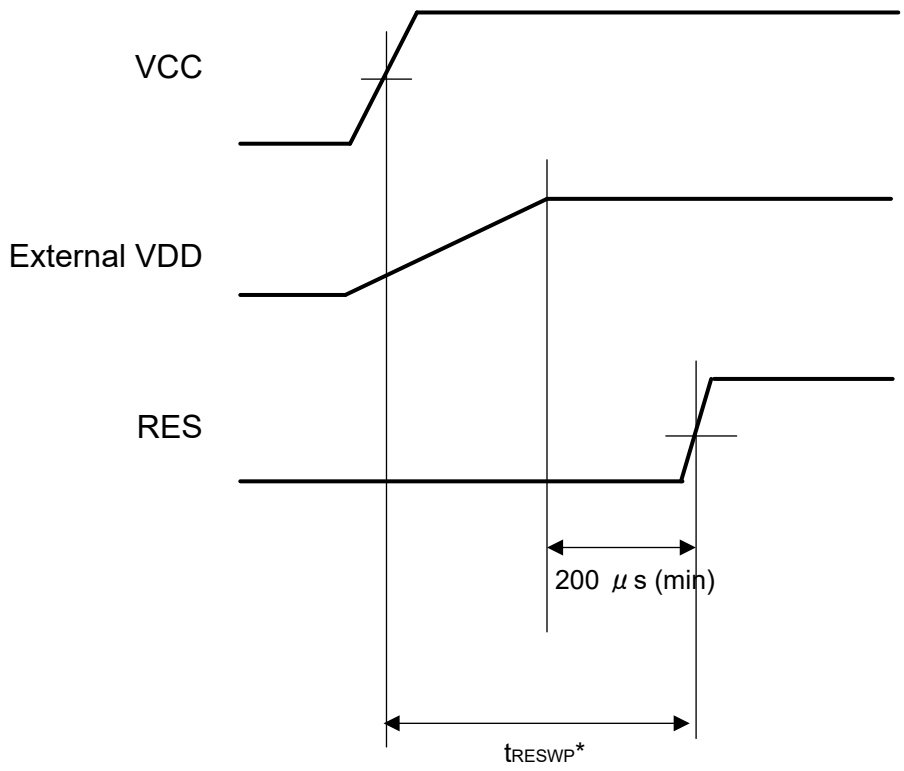


Figure 59.3 Power-up sequence of external VDD mode without using RES pin



Note: See Reset timing Characteristics for t_{RESWP}

Figure 59.4 Power-up sequence of external VDD mode with using RES pin

59.3 Usage Notes

In DCDC mode, to minimize power loss and maximize efficiency, Renesas recommends using a 2.2-µH inductor with a DC resistance of 100 mΩ or less. Short VCC and VCC_DCDC.

In external VDD mode, the voltage of VDD should always be below the voltage of VCC including the power-on and power-off sequence. Refer to the chapter of electrical characteristics for more information.

60. Electrical Characteristics

Unless otherwise specified, minimum and maximum values are guaranteed by either design simulation, characterization results or test in production.

Supported peripheral functions and pins differ from one product name to another.

Unless otherwise specified, the electrical characteristics of the MCU are defined under the following conditions:

- VCC = VCC_DCDC = VCC_USB = VBATT = 1.68 to 3.6 V
- VCC2 = 1.65 to 3.6 V
- AVCC0 = 1.65 to 3.6 V
- VCC_USBHS = AVCC_USBHS = 3.0 to 3.6 V
- AVCC_MIPI = 3.0 to 3.6 V
- VREFH0 = 2.7 V to AVCC0
- VREFH = 1.65 V to AVCC0
- VCC18_MIPI = 1.7 to 1.9 V
- VSS = VSS_DCDC = AVSS0 = VREFL0 /VREFL = VSS_USB = VSS1_USBHS = VSS2_USBHS = VSS_MIPI = 0 V
- VCC voltage is lower than 2.7 V : LVOCR.LVO0E = 1, otherwise LVOCR.LVO0E = 0
- VCC2 voltage is lower than 2.7 V : LVOCR.LVO1E = 1, otherwise LVOCR.LVO1E = 0
- $T_j = T_{opj}$

When not specified otherwise, typical values are measured at room temperature of 25 °C and VCC = VCC_DCDC = VCC_USB = VBATT = VCC_USBHS = AVCC_USBHS = AVCC0 = AVCC_MIPI = VREFH0 = VREFH = 3.3V, VCC18_MIPI = 1.8V.

Figure 60.1 shows the timing conditions.

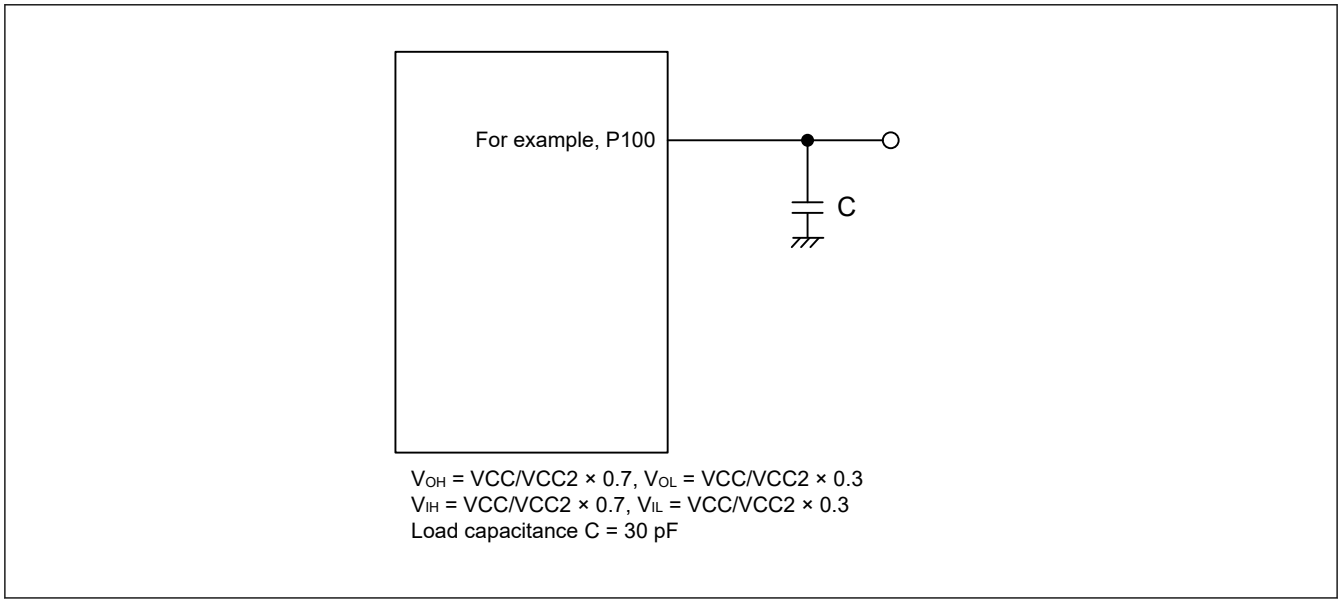


Figure 60.1 Input or output timing measurement conditions

The recommended measurement conditions for the timing specification of each peripheral provided are for the best peripheral operation. Make sure to adjust the driving abilities of each pin to meet your conditions.

60.1 Absolute Maximum Ratings

Table 60.1 Absolute maximum ratings

Parameter	Symbol	Value	Unit
Power supply voltage	VCC, VCC2, VCC_DCDC, VCC_USB*2	–0.3 to +4.0	V
External power supply voltage	VCL	–0.3 to +1.6	V
VBATT power supply voltage	VBATT	–0.3 to +4.0	V
Input voltage (except for 5 V-tolerant ports*1)	V_{in}	–0.3 to VCC + 0.3 or –0.3 to VCC2 + 0.3	V
Input voltage (5 V-tolerant ports*1)	V_{in}	–0.3 to + VCC + 4.0 (max. 5.8)	V
Reference power supply voltage	VREFH/VREFH0	–0.3 to AVCC0 + 0.3	V
USBHS power supply voltage	VCC_USBHS	–0.3 to +4.0	V
USBHS analog power supply voltage	AVCC_USBHS	–0.3 to +4.0	V
MIPI PHY analog power supply voltage	AVCC_MIPI	–0.3 to +4.0	V
MIPI PHY power supply voltage	VCC18_MIPI	–0.3 to +2.5	V
MIPI_DL0_P and MIPI_DL0_N input voltage	V_{PIN}	–0.5 to VCC18_MIPI + 0.5	V
Analog power supply voltage	AVCC0	–0.3 to +4.0	V
Analog input voltage	V_{AN}	–0.3 to AVCC0 + 0.3	V
Operating junction temperature*3 *4	T_{opj}	–40 to +125	°C
Storage temperature	T_{stg}	–55 to +125	°C

Note 1. Ports P205, P206, P402 to P404, P406 to P415, P511, P512, P709 to P715 and PB01 are 5 V tolerant.
Note 2. Connect VCC_DCDC and VCC_USB to VCC.
Note 3. See section 60.2.1. Tj/Ta Definition.
Note 4. Contact a Renesas Electronics sales office for information on derating operation when Tj = +105°C to +125°C. Derating is the systematic reduction of load for improved reliability.

Caution: Permanent damage to the MCU might result if absolute maximum ratings are exceeded.

Table 60.2 Recommended operating conditions (1 of 2)

Parameter	Symbol		Min	Typ	Max	Unit
Power supply voltages	VCC, VCC_DCDC	Other than the following	1.68	—	3.60	V
		When ETHERC/IIC Fast-mode+ is used	2.70	—	3.60	V
		When USB/SDRAM is used	3.00	—	3.60	V
	VCC2		1.65	—	3.60	V
	VCL	When external VDD is used*2	1.20	—	1.25	V
		When DCDC is used (High-speed mode)	—	1.21	—	V
		When DCDC is used (Low-speed mode or Software Standby mode)	—	1.18	—	V
	VSS, VSS_DCDC		—	0	—	V
USB power supply voltages	VCC_USB, VCC_USBHS, AVCC_USBHS		—	VCC	—	V
	VSS_USB, VSS1_USBHS, VSS2_USBHS		—	0	—	V
MIPI PHY power supply voltages	VCC18_MIPI		1.70	1.80	1.90	V
	AVCC_MIPI		3.00	—	3.60	V
	VSS_MIPI		—	0	—	V
VBATT power supply voltage	VBATT		1.62	—	3.60	V

Table 60.2 Recommended operating conditions (2 of 2)

Parameter	Symbol		Min	Typ	Max	Unit
Analog power supply voltages	AVCC0*1	When ADC is not used	1.65	—	3.60	V
		When ADC is used	2.70	—	3.60	V
	AVSS0		—	0	—	V

Note 1. When the A/D converter, the D/A converter and the High-Speed Analog Comparator are not in use, do not leave the AVCC0, VREFH/VREFH0, AVSS0, and VREFL/VREFL0 pins open. Connect the AVCC0 and VREFH/VREFH0 pins to VCC, and the AVSS0 and VREFL/VREFL0 pins to VSS, respectively.
Note 2. VCL voltage must never be higher than VCC voltage.

60.2 DC Characteristics

60.2.1 Tj/Ta Definition

Table 60.3 DC characteristics

Parameter	Symbol	Typ	Max	Unit	Test conditions
Permissible operating junction temperature	T_j	—	125	°C	High-speed mode Low-speed mode

Note: Make sure that $T_j = T_a + \theta_{ja} \times$ total power consumption (W), where total power consumption = $(VCC - V_{OH}) \times \Sigma I_{OH} + V_{OL} \times \Sigma I_{OL} + (I_{CC}max + I_{CC\_DCDC}max) \times VCC$.
Note: Minimum Ambient Temperature(Ta) is -40℃

60.2.2 I/O V_{IH}, V_{IL}

Table 60.4 I/O V_{IH}, V_{IL} except for Schmitt trigger input pins

Parameter		VCC/VCC2/ AVCC0	Symbol	Min	Typ	Max	Unit
Peripheral function pins	EXTAL (external clock input), WAIT, SPI*1 (except RSPCK)	1.68 V or above	V_{IH}	VCC × 0.8	—	—	V
			V_{IL}	—	—	VCC × 0.2	
	SPI*2 (except RSPCKB_A)	1.65 V or above	V_{IH}	VCC2 × 0.8	—	—	
			V_{IL}	—	—	VCC2 × 0.2	
	OSPI (except OM_RSTO1 and OM_ECSINT1)	2.70 V or above	V_{IH}	VCC2 × 0.8	—	—	
			V_{IL}	—	—	VCC2 × 0.2	
		1.65 V or above	V_{IH}	VCC2 × 0.7	—	VCC2 + 0.3	
			V_{IL}	VSS - 0.3	—	VCC2 × 0.3	
	SD*3	2.70 V or above	V_{IH}	VCC × 0.625	—	VCC + 0.3	
			V_{IL}	VSS − 0.3	—	VCC × 0.25	
		1.70 V ~ 1.95 V	V_{IH}	1.27	—	2	
			V_{IL}	VSS − 0.3	—	0.58	
	SD*4	2.70 V or above	V_{IH}	VCC2 × 0.625	—	VCC2 + 0.3	
			V_{IL}	VSS − 0.3	—	VCC2 × 0.25	
		1.70 V ~ 1.95 V	V_{IH}	1.27	—	2	
			V_{IL}	VSS − 0.3	—	0.58	
	MMC*5	2.70 V or above	V_{IH}	VCC × 0.625	—	VCC + 0.3	
			V_{IL}	VSS − 0.3	—	VCC × 0.25	
		1.70 V ~ 1.95 V	V_{IH}	VCC × 0.65	—	VCC + 0.3	
			V_{IL}	VSS − 0.3	—	VCC × 0.35	
	MMC*6	2.70 V or above	V_{IH}	VCC2 × 0.625	—	VCC2 + 0.3	
			V_{IL}	VSS − 0.3	—	VCC2 × 0.25	
		1.70 V ~ 1.95 V	V_{IH}	VCC2 × 0.65	—	VCC2 + 0.3	
			V_{IL}	VSS − 0.3	—	VCC2 × 0.35	
	D00 to D31, TMS, TDI, TCK, SWDIO, SWCLK	1.68 V or above	V_{IH}	VCC × 0.7	—	—	
			V_{IL}	—	—	VCC × 0.3	
	DQ00 to DQ31	3.00 V or above	V_{IH}	VCC × 0.7	—	—	
			V_{IL}	—	—	VCC × 0.3	
	ETHERC	2.70 V or above	V_{IH}	2.3	—	—	
			V_{IL}	—	—	VCC × 0.2	
	IIC (SMBus)	2.70 V or above	V_{IH}	2.1	—	VCC + 3.6 (max 5.8)	
			V_{IL}	—	—	0.8	
	RTCIC0, RTCIC1, RTCIC2, EXCIN when VCC power supply is selected	1.68 V or above	V_{IH}	0.9	—	3.9	
			V_{IL}	—	—	0.3	
	RTCIC0, RTCIC1, RTCIC2, EXCIN when VBATT power supply is selected		V_{IH}	0.9	—	3.9	
			V_{IL}	—	—	0.3	

Note 1. SPI0_A, SPI0_B, SPI0_C and SPI1_B
Note 2. SPI1_A
Note 3. SD_A ch0, SD_B ch1 and SD_B ch1
Note 4. SD_A ch1
Note 5. MMC_A ch0, MMC_A ch1 and MMC_B ch1
Note 6. MMC_A ch1 (Up to 4-bit bus width)

Table 60.5 I/O V_{IH}, V_{IL} of Schmitt trigger input pins

Parameter		VCC/VCC2/ AVCC0	Symbol	Min	Typ	Max	Unit
Peripheral function pins	IIC (except for SMBus)	1.68 V or above	V_{IH}	VCC × 0.7	—	VCC + 3.6 (max 5.8)	V
			V_{IL}	—	—	VCC × 0.3	
			ΔV_T	VCC × 0.05	—	—	
	I3C	1.68 V or above	V_{IH}	VCC × 0.7	—	VCC + 0.3	
			V_{IL}	—	—	VCC × 0.3	
			ΔV_T	VCC × 0.1	—	—	
	5 V-tolerant ports*1*5	1.68 V or above	V_{IH}	VCC × 0.8	—	VCC + 3.6 (max 5.8)	
			V_{IL}	—	—	VCC × 0.2	
			ΔV_T	VCC × 0.05	—	—	
	Other VCC input pins*2	1.68 V or above	V_{IH}	VCC × 0.8	—	—	
			V_{IL}	—	—	VCC × 0.2	
			ΔV_T	VCC × 0.05	—	—	
	Other VCC2 input pins*2	1.65 V or above	V_{IH}	VCC2 × 0.8	—	—	
			V_{IL}	—	—	VCC2 × 0.2	
			ΔV_T	VCC2 × 0.05	—	—	
	Other AVCC0 input pins*2	1.65 V or above	V_{IH}	AVCC0 × 0.8	—	—	
			V_{IL}	—	—	AVCC0 × 0.2	
			ΔV_T	AVCC0 × 0.05	—	—	
Ports	5 V-tolerant port*3*5	1.68 V or above	V_{IH}	VCC × 0.8	—	VCC + 3.6 (max 5.8)	V
			V_{IL}	—	—	VCC × 0.2	
	Other VCC input pins*4	1.68 V or above	V_{IH}	VCC × 0.8	—	—	
			V_{IL}	—	—	VCC × 0.2	
	Other VCC2 input pins*4	1.65 V or above	V_{IH}	VCC2 × 0.8	—	—	
			V_{IL}	—	—	VCC2 × 0.2	
	Other AVCC0 input pins*4	1.65 V or above	V_{IH}	AVCC0 × 0.8	—	—	
			V_{IL}	—	—	AVCC0 × 0.2	

Note 1. RES and peripheral function pins associated with P205, P206, P402 to P404, P406 to P415, P511, P512, P709 to P715, PB01 (total 26 pins).
Note 2. All input pins except for the peripheral function pins already described in the table. There is an item for each power supply voltage for each port. Refer to the IO chapter for the power supply of the port.
Note 3. P205, P206, P402 to P404, P406 to P415, P511, P512, P709 to P715, PB01 (total 25 pins).
Note 4. All input pins except for the ports already described in the table. There is an item for each power supply voltage for each port. Refer to the IO chapter for the power supply of the port.
Note 5. When VCC is less than 1.68 V, the input voltage of 5 V-tolerant ports should be less than 3.6 V, otherwise breakdown may occur because 5 V-tolerant ports are electrically controlled so as not to violate the break down voltage.

60.2.3 I/O I_{OH}, I_{OL}

Table 60.6 I/O I_{OH}, I_{OL} (1 of 2)

Parameter			VCC/ VCC2/ AVCC0	Symbol	Min	Typ	Max	Unit
Permissible output current (average value per pin)	Ports P000 to P011, P014, P015, P201	—	—	I_{OH}	—	—	−2.0	mA
				I_{OL}	—	—	2.0	mA
	Ports P205, P206, P402 to P404, P406 to P415, P511, P512, P709 to P715, PA09, PB01 (total 26 pins)	Low drive*1	—	I_{OH}	—	—	−2.0	mA
				I_{OL}	—	—	2.0	mA
		Middle drive*2	—	I_{OH}	—	—	−4.0	mA
				I_{OL}	—	—	4.0	mA
		High drive*3	—	I_{OH}	—	—	−20	mA
				I_{OL}	—	—	20.0	mA
	Ports P100 to P103, P304 to P308, P800 to P804, P808 to P810, PA09 (total 18 pins)	Low drive*1	—	I_{OH}	—	—	−2.0	mA
				I_{OL}	—	—	2.0	mA
		Middle drive*2	—	I_{OH}	—	—	−4.0	mA
				I_{OL}	—	—	4.0	mA
		High drive*3	—	I_{OH}	—	—	−16	mA
				I_{OL}	—	—	16.0	mA
		High-speed high drive*4	—	I_{OH}	—	—	−20	mA
				I_{OL}	—	—	20.0	mA
	Port PA09	Low drive*1	—	I_{OH}	—	—	−2.0	mA
				I_{OL}	—	—	2.0	mA
		Middle drive*2	—	I_{OH}	—	—	−4.0	mA
				I_{OL}	—	—	4.0	mA
		High drive*3	—	I_{OH}	—	—	−20	mA
				I_{OL}	—	—	20.0	mA
		High-speed high drive*4	—	I_{OH}	—	—	−20	mA
				I_{OL}	—	—	20.0	mA
	Other output pins*5	Low drive*1	—	I_{OH}	—	—	−2.0	mA
				I_{OL}	—	—	2.0	mA
		Middle drive*2	—	I_{OH}	—	—	−4.0	mA
				I_{OL}	—	—	4.0	mA
		High drive*3	—	I_{OH}	—	—	−16	mA
				I_{OL}	—	—	16.0	mA

Table 60.6 I/O I_{OH}, I_{OL} (2 of 2)

Parameter			VCC/ VCC2/ AVCC0	Symbol	Min	Typ	Max	Unit
Permissible output current (max value per pin)	Ports P000 to P011, P014, P015, P201	—	—	I_{OH}	—	—	−4.0	mA
				I_{OL}	—	—	4.0	mA
	Ports P205, P206, P402 to P404, P406 to P415, P511, P512, P709 to P715, PA09, PB01 (total 26 pins)	Low drive*1	—	I_{OH}	—	—	−4.0	mA
				I_{OL}	—	—	4.0	mA
		Middle drive*2	—	I_{OH}	—	—	−8.0	mA
				I_{OL}	—	—	8.0	mA
		High drive*3	—	I_{OH}	—	—	−40	mA
				I_{OL}	—	—	40.0	mA
	Ports P100 to P103, P304 to P308, P800 to P804, P808 to P810, PA09 (total 18 pins)	Low drive*1	—	I_{OH}	—	—	−4.0	mA
				I_{OL}	—	—	4.0	mA
		Middle drive*2	—	I_{OH}	—	—	−8.0	mA
				I_{OL}	—	—	8.0	mA
		High drive*3	—	I_{OH}	—	—	−32	mA
				I_{OL}	—	—	32.0	mA
		High-speed high drive*4	—	I_{OH}	—	—	−40	mA
				I_{OL}	—	—	40.0	mA
	Other output pins*5	Low drive*1	—	I_{OH}	—	—	−4.0	mA
				I_{OL}	—	—	4.0	mA
		Middle drive*2	—	I_{OH}	—	—	−8.0	mA
				I_{OL}	—	—	8.0	mA
		High drive*3	—	I_{OH}	—	—	−32	mA
				I_{OL}	—	—	32.0	mA
Permissible output current (max value of total of all pins)	Maximum of all output pins	VCC I/O	1.68 V or above	$\Sigma I_{OH\ (max)}$	—	—	−80	mA
		VCC2 I/O	1.65 V or above		—	—	−80	
		AVCC0 I/O	1.65 V or above		—	—	−33	
		VCC and VCC2 I/O	1.65 V or above	$\Sigma I_{OL\ (max)}$	—	—	80	mA
		AVCC0 I/O	1.65 V or above		—	—	33	

Note 1. This is the value when low driving ability is selected in the Port Drive Capability bit in the PmnPFS register. The selected driving ability except for 400 and P401 is retained in Deep Software Standby mode.
Note 2. This is the value when middle driving ability is selected in the Port Drive Capability bit in the PmnPFS register. The selected driving ability except for 400 and P401 is retained in Deep Software Standby mode.
Note 3. This is the value when high driving ability is selected in the Port Drive Capability bit in the PmnPFS register. The selected driving ability except for 400 and P401 is retained in Deep Software Standby mode.
Note 4. This is the value when high-speed high driving ability is selected in the Port Drive Capability in the PmnPFS register. The selected driving ability is retained in Deep Software Standby mode.
Note 5. Except for P200, which is an input port.

Caution: To protect the reliability of the MCU, the output current values should not exceed the values in this table. The average output current indicates the average value of current measured during 100 µs.

60.2.4 I/O V_{OH}, V_{OL}, and Other Characteristics

Table 60.7 I/O V_{OH}, V_{OL}, and other characteristics (1 of 3)

Parameter		VCC/VCC2/ AVCC0	Symbol	Min	Typ	Max	Unit	Test conditions
Output voltage	IIC	2.70 V or above	V_{OL}	—	—	0.4	V	I_{OL} = 3.0 mA
			V_{OL}	—	—	0.6		I_{OL} = 6.0 mA
		1.68 V or above	V_{OL}	—	—	VCC × 0.2		I_{OL} = 3.0 mA
			V_{OL}	—	—	0.6		I_{OL} = 6.0 mA
	IIC*1	2.70 V or above	V_{OL}	—	—	0.4		I_{OL} = 15.0 mA (ICFER.FMPE = 1)
			V_{OL}	—	0.4	—		I_{OL} = 20.0 mA (ICFER.FMPE = 1)
	I3C	2.70 V or above	V_{OL}	—	—	0.4		I_{OL} = 3.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 0, BFCTL.HSME = 0)
			V_{OL}	—	—	0.6		I_{OL} = 6.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 0, BFCTL.HSME = 0)
			V_{OL}	—	—	0.4		I_{OL} = 15.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 1, BFCTL.HSME = 0)
			V_{OL}	—	0.4	—		I_{OL} = 20.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 1, BFCTL.HSME = 0)
			V_{OL}	—	—	0.4		I_{OL} = 3.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 0, BFCTL.HSME = 1)
			V_{OH}	VCC − 0.27	—	—		I_{OH} = 3.0 mA (PRTS.PRTMD = 0, BFCTL.FMPE = 0, BFCTL.HSME = 0)
			V_{OL}	—	—	0.27		I_{OL} = 3.0 mA (PRTS.PRTMD = 0, BFCTL.FMPE = 0, BFCTL.HSME = 0)
		1.68 V or above	V_{OL}	—	—	VCC × 0.2		I_{OL} = 3.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 0, BFCTL.HSME = 0)
			V_{OL}	—	—	0.6		I_{OL} = 6.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 0, BFCTL.HSME = 0)
			V_{OL}	—	—	VCC × 0.2		I_{OL} = 3.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 0, BFCTL.HSME = 1)
			V_{OH}	VCC − 0.27	—	—		I_{OH} = 3.0 mA (PRTS.PRTMD = 0, BFCTL.FMPE = 0, BFCTL.HSME = 0)
			V_{OL}	—	—	0.27		I_{OL} = 3.0 mA (PRTS.PRTMD = 0, BFCTL.FMPE = 0, BFCTL.HSME = 0)
	ETHERC	2.70 V or above	V_{OH}	VCC − 0.5	—	—		I_{OH} = −1.0 mA
			V_{OL}	—	—	0.4		I_{OL} = 1.0 mA
	SD	2.70 V or above	V_{OH}	VCC × 0.75	—	—		I_{OH} = −2.0 mA
			V_{OL}	—	—	VCC × 0.125		I_{OL} = 3.0 mA
			V_{OH}	VCC2 × 0.75	—	—		I_{OH} = −2.0 mA
			V_{OL}	—	—	VCC2 × 0.125		I_{OL} = 3.0 mA
		1.70 V to 1.95 V	V_{OH}	1.4	—	—		I_{OH} = −2.0 mA
			V_{OL}	—	—	0.45		I_{OL} = 2.0 mA

Table 60.7 I/O V_{OH}, V_{OL}, and other characteristics (2 of 3)

Parameter		VCC/VCC2/ AVCC0	Symbol	Min	Typ	Max	Unit	Test conditions
Output voltage	MMC	2.70 V or above	V_{OH}	VCC × 0.75	—	—	V	I_{OH} = −0.1 mA (VCC = 2.7V)
			V_{OL}	—	—	VCC × 0.125		I_{OL} = 0.1 mA (VCC = 2.7V)
			V_{OH}	VCC2 × 0.75	—	—		I_{OH} = −0.1mA (VCC2 = 2.7V)
			V_{OL}	—	—	VCC2 × 0.125		I_{OL} = 0.1 mA (VCC2 = 2.7V)
		1.70 V to 1.95 V	V_{OH}	VCC − 0.45	—	—		I_{OH} = −2.0 mA
			V_{OL}	—	—	0.45		I_{OL} = 2.0 mA
			V_{OH}	VCC2 − 0.45	—	—		I_{OH} = −2.0 mA
	Ports P205, P206, P402 to P404, P406 to P415, P511, P512, P709 to P715, PA09, PB01 (total 26 pins)*2	—	V_{OH}	VCC − 1.0	—	—		I_{OH} = −20 mA VCC = 3.3 V
		—	V_{OL}	—	—	1		I_{OL} = 20 mA VCC = 3.3 V
	Other output pins	1.68 V or above	V_{OH}	VCC − 0.5	—	—		I_{OH} = −1.0 mA
			V_{OL}	—	—	0.5		I_{OL} = 1.0 mA
		1.65V or above	V_{OH}	VCC2 − 0.5	—	—		I_{OH} = −1.0 mA
			V_{OL}	—	—	0.5		I_{OL} = 1.0 mA
			V_{OH}	AVCC0 − 0.5	—	—		I_{OH} = −1.0 mA
			V_{OL}	—	—	0.5		I_{OL} = 1.0 mA
Input leakage current	RES	1.68 V or above	$\|I_{in}\|$	—	—	5.0	µA	V_{in} = 0 V V_{in} = 5.5 V
	Port P200	1.68 V or above		—	—	1.0		V_{in} = 0 V V_{in} = VCC
Three-state leakage current (off state)	5 V-tolerant ports	1.68 V or above	$\|I_{TSI}\|$	—	—	5.0	µA	V_{in} = 0 V V_{in} = 5.5 V
	Other ports (except for port P200)	1.68 V or above		—	—	1.0		V_{in} = 0 V V_{in} = VCC
		1.65 V or above		—	—	1.0		V_{in} = 0 V V_{in} = VCC2, AVCC0
Input pull-up MOS current	Ports P0 to PB	2.70 V or above	I_p	−300	—	−10	µA	VCC, VCC2, AVCC0 = 2.7 to 3.6 V V_{in} = 0 V
		1.68 V or above		−300	—	−5	µA	VCC = 1.68 to 3.6 V V_{in} = 0 V
		1.65 V or above		−300	—	−5	µA	VCC2, AVCC0 = 1.65 to 3.6 V V_{in} = 0 V

Table 60.7 I/O V_{OH}, V_{OL}, and other characteristics (3 of 3)

Parameter		VCC/VCC2/ AVCC0	Symbol	Min	Typ	Max	Unit	Test conditions
Pull-up current serving as the SCL current source	I3C*3	3.0 V to 3.6 V	I_{cs}	3	—	12	mA	VCC = 3.0 to 3.6 V V_{in} = 0.3 × VCC to 0.7 × VCC
		1.68 V to 1.95 V						VCC = 1.68 to 1.95 V V_{in} = 0.3 × VCC to 0.7 × VCC
Input capacitance	Ports P014, P015	—	C_{in}	—	—	16	pF	Vbias = 0 V Vamp = 20 mV f = 1 MHz Ta = 25°C
	Ports P814/ USB_DP, P815/ USB_DM	—		—	—	12		
	Ports P400, P401, P409, P410, P511, P512, USBHS_DP, USBHS_DM, MIPI_DL0_P, MIPI_DL0_N	—		—	—	10		
	Other input pins	—		—	—	8		

Note 1. SCL0_A, SDA0_A, SCL1_A, SDA1_A (total 4 pins).
Note 2. This is the value when high driving ability is selected in the Port Drive Capability bit in the PmnPFS register. The selected driving ability is retained in Deep Software Standby mode.
Note 3. I3C_SCL0 (1 pin). This is the value when IIC high speed mode is selected.

60.2.5 Operating and Standby Current

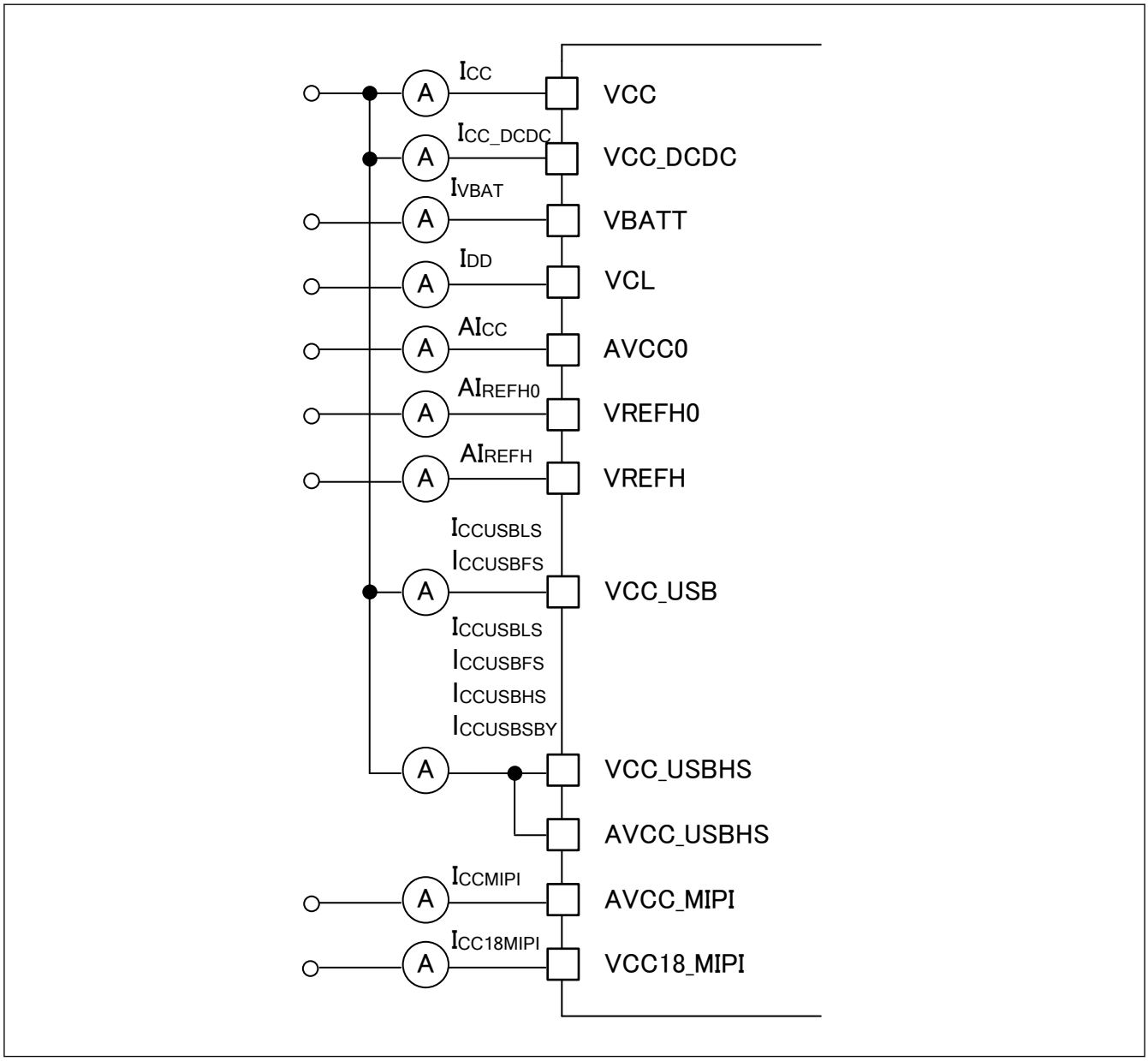


Figure 60.2 Consumption current measurement diagram

Table 60.8 Current of high-speed mode, maximum condition (MVE and peripheral operation) (DCDC mode)

Parameter			Symbol	Typ	Max		Unit	Test conditions
					105 °C	125 °C		
Supply current*1*2	—		I_{CC}	2.8	7.05	7.19	mA	
	CPUCLK = 480 MHz	VCC_DCDC ≥ 2.5 V	I_{CC_DCDC}*5	170	303	—	mA	VCC_DCDC = 3.3 V CPUCLK = 480 MHz, ICLK = 240 MHz, PCLKA = 120 MHz, PCLKB = 60 MHz, PCLKC = 60 MHz, PCLKD = 120 MHz, PCLKE = 240 MHz, FCLK = 60 MHz, BCLK = 120 MHz
			I_{DD}*3	398	624	—		
		VCC_DCDC < 2.5 V	I_{CC_DCDC}*5	311	320	—		VCC_DCDC = 1.8 V
			I_{DD}	398	400*4	—		
	CPUCLK = 400 MHz	VCC_DCDC ≥ 2.5 V	I_{CC_DCDC}*5	142	267	307	mA	VCC_DCDC = 3.3 V CPUCLK = 400 MHz, ICLK = 200 MHz, PCLKA = 100 MHz, PCLKB = 50 MHz, PCLKC = 50 MHz, PCLKD = 100 MHz, PCLKE = 100 MHz, FCLK = 50 MHz, BCLK = 100 MHz
			I_{DD}*3	333	550	632		
		VCC_DCDC < 2.5 V	I_{CC_DCDC}*5	260	320	320		VCC_DCDC = 1.8 V
			I_{DD}	333	400*4	400*4		
	CPUCLK = 360 MHz	VCC_DCDC ≥ 2.5 V	I_{CC_DCDC}*5	133	257	297	mA	VCC_DCDC = 3.3 V CPUCLK = 360 MHz, ICLK = 120 MHz, PCLKA = 120 MHz, PCLKB = 60 MHz, PCLKC = 60 MHz, PCLKD = 120 MHz, PCLKE = 120 MHz, FCLK = 60 MHz, BCLK = 120 MHz
			I_{DD}*3	311	530	612		
		VCC_DCDC < 2.5 V	I_{CC_DCDC}*5	243	320	320		VCC_DCDC = 1.8 V
			I_{DD}	311	400*4	400*4		
	CPUCLK = 240 MHz	VCC_DCDC ≥ 2.5 V	I_{CC_DCDC}*5	106	224	264	mA	VCC_DCDC = 3.3 V CPUCLK = 240 MHz, ICLK = 240 MHz, PCLKA = 120 MHz, PCLKB = 60 MHz, PCLKC = 60 MHz, PCLKD = 120 MHz, PCLKE = 120 MHz, FCLK = 60 MHz, BCLK = 120 MHz
			I_{DD}*3	248	460	544		
		VCC_DCDC < 2.5 V	I_{CC_DCDC}*5	194	320	320		VCC_DCDC = 1.8 V
			I_{DD}	248	400*4	400*4		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Measured with clocks supplied to the peripheral functions and peripherals being operated. This does not include the BGO operation.
Note 3. I_{DD} depends on f (CPUCLK and ICLK) as follows.
I_{DD} Max. (105 °C) = 0.68 × fCPUCLK + 0.51 × fICLK + 180 (unit : mA, fCPUCLK and fICLK are MHz)
I_{DD} Max. (125 °C) = 0.68 × fCPUCLK + 0.51 × fICLK + 258 (unit : mA, fCPUCLK and fICLK are MHz)
Note 4. Do not actual consumption current during operation exceed the current value described here in VCC_DCDC < 2.5 V.
Note 5. Typical DCDC efficiency is applied.

Table 60.9 Current of high-speed mode, maximum condition (MVE and peripheral operation) (External VDD mode)

Parameter	CPUCLK Frequency	Symbol	Typ	Max		Unit	Test conditions
				105 °C	125 °C		
Supply current*1*2	—	I_{CC}	2.8	7.05	7.19	mA	
	CPUCLK = 480 MHz	I_{DD}*3	398	624	—	mA	CPUCLK = 480 MHz, ICLK = 240 MHz, PCLKA = 120 MHz, PCLKB = 60 MHz, PCLKC = 60 MHz, PCLKD = 120 MHz, PCLKE = 240 MHz, FCLK = 60 MHz, BCLK = 120 MHz
	CPUCLK = 400 MHz	I_{DD}*3	333	550	632	mA	CPUCLK = 400 MHz, ICLK = 200 MHz, PCLKA = 100 MHz, PCLKB = 50 MHz, PCLKC = 50 MHz, PCLKD = 100 MHz, PCLKE = 100 MHz, FCLK = 50 MHz, BCLK = 100 MHz
	CPUCLK = 360 MHz	I_{DD}*3	311	530	612	mA	CPUCLK = 360 MHz, ICLK = 120 MHz, PCLKA = 120 MHz, PCLKB = 60 MHz, PCLKC = 60 MHz, PCLKD = 120 MHz, PCLKE = 120 MHz, FCLK = 60 MHz, BCLK = 120 MHz
	CPUCLK = 240 MHz	I_{DD}*3	248	460	544	mA	CPUCLK = 240 MHz, ICLK = 240 MHz, PCLKA = 120 MHz, PCLKB = 60 MHz, PCLKC = 60 MHz, PCLKD = 120 MHz, PCLKE = 120 MHz, FCLK = 60 MHz, BCLK = 120 MHz

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Measured with clocks supplied to the peripheral functions and peripherals being operated. This does not include the BGO operation.
Note 3. I_{DD} depends on f (CPUCLK and ICLK) as follows.
I_{DD} Max. (105 °C) = 0.68 × fCPUCLK + 0.51 × fICLK + 180 (unit : mA, fCPUCLK and fICLK are MHz)
I_{DD} Max. (125 °C) = 0.68 × fCPUCLK + 0.51 × fICLK + 258 (unit : mA, fCPUCLK and fICLK are MHz)

Table 60.10 Current of high-speed mode, maximum data processing (MVE operation), peripheral clock ON (DCDC mode)

Parameter		Symbol	Typ	Max		Unit	Test conditions
				105 °C	125 °C		
Supply current*1*2	CPUCLK = 480 MHz	I_{CC_DCDC} *4	156	276	—	mA	VCC_DCDC = 3.3 V *5
		I_{DD}*3	366	568	—		
	CPUCLK = 400 MHz	I_{CC_DCDC} *4	131	246	286		
		I_{DD}*3	308	506	589		
	CPUCLK = 360 MHz	I_{CC_DCDC} *4	121	232	272		
		I_{DD}*3	283	478	561		
	CPUCLK = 240 MHz	I_{CC_DCDC} *4	94	198	239		
		I_{DD}*3	220	408	492		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Measured with clocks supplied to the peripheral functions and peripherals being operated. This does not include the BGO operation.
Note 3. I_{DD} depends on f (CPUCLK and ICLK) as follows.
I_{DD} Max. (105 °C) = 0.67 × fCPUCLK + 0.30 × fICLK + 180 (unit : mA, fCPUCLK and fICLK are MHz)
I_{DD} Max. (125 °C) = 0.67 × fCPUCLK + 0.30 × fICLK + 258 (unit : mA, fCPUCLK and fICLK are MHz)
Note 4. Typical DCDC efficiency is applied.
Note 5. Same frequency condition is applied as in the maximum condition.

Table 60.11 Current of high-speed mode, maximum data processing (MVE operation), peripheral clock ON (External VDD mode)

Parameter		Symbol	Typ	Max		Unit	Test conditions
				105 °C	125 °C		
Supply current*1*2	CPUCLK = 480 MHz	I_{DD}*3	366	568	—	mA	*4
	CPUCLK = 400 MHz	I_{DD}*3	308	506	589		
	CPUCLK = 360 MHz	I_{DD}*3	283	478	561		
	CPUCLK = 240 MHz	I_{DD}*3	220	408	492		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Measured with clocks supplied to the peripheral functions and peripherals being operated. This does not include the BGO operation.
Note 3. I_{DD} depends on f (CPUCLK and ICLK) as follows.
I_{DD} Max. (105 °C) = 0.67 × fCPUCLK + 0.30 × fICLK + 180 (unit : mA, fCPUCLK and fICLK are MHz)
I_{DD} Max. (125 °C) = 0.67 × fCPUCLK + 0.30 × fICLK + 258 (unit : mA, fCPUCLK and fICLK are MHz)
Note 4. Same frequency condition is applied as in the maximum condition.

Table 60.12 Current of high-speed mode, maximum data processing (MVE operation), peripheral clock OFF (DCDC mode)

Parameter		Symbol	Typ	Max		Unit	Test conditions
				105 °C	125 °C		
Supply current*1*2	CPUCLK = 480 MHz	I_{CC_DCDC} *4	146	263	—	mA	VCC_DCDC = 3.3 V *5
		I_{DD}*3	342	541	—		
	CPUCLK = 400 MHz	I_{CC_DCDC} *4	123	235	274		
		I_{DD}*3	289	483	565		
	CPUCLK = 360 MHz	I_{CC_DCDC} *4	111	219	259		
		I_{DD}*3	261	450	533		
	CPUCLK = 240 MHz	I_{CC_DCDC} *4	83	184	224		
		I_{DD}*3	194	378	462		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 3. I_{DD} depends on f (CPUCLK and ICLK) as follows.
I_{DD} Max. (105 °C) = 0.68 × fCPUCLK + 0.17 × fICLK + 180 (unit : mA, fCPUCLK and fICLK are MHz)
I_{DD} Max. (125 °C) = 0.68 × fCPUCLK + 0.17 × fICLK + 258 (unit : mA, fCPUCLK and fICLK are MHz)
Note 4. Typical DCDC efficiency is applied.
Note 5. Same frequency condition is applied as in the maximum condition.

Table 60.13 Current of high-speed mode, maximum data processing (MVE operation), peripheral clock OFF (External VDD mode)

Parameter		Symbol	Typ	Max		Unit	Test conditions
				105 °C	125 °C		
Supply current*1*2	CPUCLK = 480 MHz	I_{DD}*3	342	541	—	mA	*4
	CPUCLK = 400 MHz	I_{DD}*3	289	483	565		
	CPUCLK = 360 MHz	I_{DD}*3	261	450	533		
	CPUCLK = 240 MHz	I_{DD}*3	194	378	462		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 3. I_{DD} depends on f (CPUCLK and ICLK) as follows.

I_{DD} Max. (105 °C) = 0.68 × fCPUCLK + 0.17 × fICLK + 180 (unit : mA, fCPUCLK and fICLK are MHz)
I_{DD} Max. (125 °C) = 0.68 × fCPUCLK + 0.17 × fICLK + 258 (unit : mA, fCPUCLK and fICLK are MHz)
Note 4. Same frequency condition is applied as in the maximum condition.

Table 60.14 Current of high-speed mode, CPU Sleep mode (DCDC mode and External VDD mode)

Parameter		Symbol	Typ	Max		Unit	Test conditions
				105 °C	125 °C		
Supply current*1*3*4	CPUCLK = 240 MHz	I_{DD}*2	29	221	315	mA	—

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. I_{DD} depends on f (CPUCLK and ICLK) as follows.
I_{DD} Typ. (25 °C) = 0.063 × fCPUCLK + 0.13 × fICLK + 17.6 (unit : mA, fCPUCLK and fICLK are MHz)
I_{DD} Max. (105 °C) = 0.063 × fCPUCLK + 0.13 × fICLK + 180 (unit : mA, fCPUCLK and fICLK are MHz)
I_{DD} Max. (125 °C) = 0.063 × fCPUCLK + 0.13 × fICLK + 258 (unit : mA, fCPUCLK and fICLK are MHz)
Note 3. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 4. ICLK, FCLK, BCLK, PCLKA, PCLKB, PCLKC, PCLKD and PCLKE are set to divided by 64.

Table 60.15 Current of high-speed mode, CPU Deep Sleep mode (DCDC mode and External VDD mode)

Parameter		Symbol	Typ	Max		Unit	Test conditions
				105 °C	125 °C		
Supply current*1*3*4	CPUCLK = 240 MHz	I_{DD}*2	12	90	117	mA	—

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. I_{DD} depends on f (ICLK) as follows.
I_{DD} Typ. (25 °C) = 0.13 × fICLK + 5.3 (unit : mA, fCPUCLK and fICLK are MHz)
I_{DD} Max. (105 °C) = 0.13 × fICLK + 72 (unit : mA, fCPUCLK and fICLK are MHz)
I_{DD} Max. (125 °C) = 0.13 × fICLK + 101 (unit : mA, fCPUCLK and fICLK are MHz)
Note 3. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 4. ICLK, FCLK, BCLK, PCLKA, PCLKB, PCLKC, PCLKD and PCLKE are set to divided by 64.

Table 60.16 Increase during BGO operation (DCDC mode and External VDD mode)

Parameter		Symbol	Typ	Max		Unit	Test conditions
				105 °C	125 °C		
Supply current*1	Data flash P/E	I_{CC}	6	—	—	mA	—
	Code flash P/E		8	—	—		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.

Table 60.17 Current of Low-speed mode (DCDC mode)

Parameter	Symbol	Typ	Max		Unit	Test conditions
			105 °C	125 °C		
Supply current*1*2*3	I_{DD}	14.5	—	—	mA	CPUCLK = ICLK = 1 MHz Graphics power domain is off

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 3. FCLK, BCLK, PCLKA, PCLKB, PCLKC, PCLKD and PCLKE are set to divided by 64 (15.6 kHz).

Table 60.18 Standby current (DCDC mode)

Parameter				Symbol	Typ	Max 125 °C	Unit	Test conditions
Supply current*1		Software Standby mode*2		I_{CC}	0.02	0.94	mA	—
		Data of SRAM and TCM is retained		I_{CC_DCDC}	0.88	28.29		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 1 (n = 0 to 6) PDRAMSCR1.RKEEP0 = 1
		Data of SRAM and TCM is not retained		I_{CC_DCDC}	0.83	26.64		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 0 (n = 0 to 6) PDRAMSCR1.RKEEP0 = 0
		Deep Software Standby mode 1		I_{CC}	5.21	148	µA	—
				I_{CC_DCDC}	0.57	5.50		—
		Increase when the function is activated	Data of Standby SRAM is retained	I_{CC}	0.12	2.60		—
			PVD0, PVD1,PVD2 or Battery power supply switch		See Table 60.20			—
			When the LOCO is in use		3.10	—		—
			Crystal oscillator and RTC		See Table 60.21			—
			IWDT and ULPT(all units) are operating		0.07	—		—
		Deep Software Standby mode 2		I_{CC}	1.68	43.99		—
				I_{CC_DCDC}	0.57	5.50		—
		Increase when the function is activated	PVD0, PVD1,PVD2 or Battery power supply switch	I_{CC}	See Table 60.20			—
			Crystal oscillator and RTC		See Table 60.21			—
		Deep Software Standby mode 3		I_{CC}	0.99	42.90		—
				I_{CC_DCDC}	0.57	5.50		—
		Increase when the function is activated	Crystal oscillator and RTC		See Table 60.21			—
	RTC operating while VCC is off (with the battery backup function, only the RTC operate)	When a crystal oscillator with low power mode 3 is in use		I_{VBAT}	0.52	—		VBATT = 1.8 V, VCC = 0 V
					1.05	—		VBATT = 3.3 V, VCC = 0 V
		When a crystal oscillator with low power mode 2 is in use			0.56	—		VBATT = 1.8 V, VCC = 0 V
					1.10	—		VBATT = 3.3 V, VCC = 0 V
		When a crystal oscillator with low power mode 1 is in use			0.62	—		VBATT = 1.8 V, VCC = 0V
					1.17	—		VBATT = 3.3 V, VCC = 0 V
		When a crystal oscillator with standard mode is in use			0.93	—		VBATT = 1.8 V, VCC = 0 V
					1.50	—		VBATT = 3.3 V, VCC = 0 V
		When EXCIN is in use			0.37	—		VBATT = 1.8 V, VCC = 0 V
					0.86	—		VBATT = 3.3 V, VCC = 0 V
		Increase when the function is activated	Common circuit when using RTCICn (n = 0 to 2) input or EXCIN		0.04	—		VBATT = 1.8 V, VCC = 0 V
					0.04	—		VBATT = 3.3 V, VCC = 0 V
			RTCICn (n = 0 to 2) input is in use per channel		0.02	—		VBATT = 1.8 V, VCC = 0 V
					0.02	—		VBATT = 3.3 V, VCC = 0 V

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. When an external clock is used, EXTAL pin is pull-up or pull-down. In case clock is toggled, software standby mode current consumption is increased by 130µA at 48MHz under typical conditions.

Table 60.19 Coremark and normal mode current (DCDC and External power supply mode)

Parameter			Symbol	Typ	Max	Unit	Test conditions
Supply current*1*2	Coremark	Cache on	I_{DD}	318	—	µA/MHz	CPUCLK = 480 MHz ICLK = 240 MHz PCLKA = 7.5 MHz PCLKB = 7.5 MHz PCLKC = 7.5 MHz PCLKD = 7.5 MHz PCLKE = 7.5 MHz FCLK = 7.5 MHz BCLK = 7.5 MHz
		Cache off, executing from ITCM		281	—		
		Cache off, executing from SRAM		178	—		
		Cache off, executing from flash		169	—		
	Normal mode	All peripheral disabled, Cache on, while (1) code		223	—		
		All peripheral disabled, Cache off, while (1) code executing from flash		138	—		
	Coremark	Cache off, executing from flash		165	—	µA/MHz	CPUCLK = 360 MHz ICLK = 120 MHz PCLKA = 30 MHz PCLKB = 30 MHz PCLKC = 30 MHz PCLKD = 30 MHz PCLKE = 30 MHz FCLK = 30 MHz BCLK = 30 MHz
	Normal mode	All peripheral disabled, Cache off, while (1) code executing from flash		137	—		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.

Table 60.20 Increase when the PVD0, PVD1, PVD2 or Battery power supply switch is enabled in Deep Software Standby mode 1 and 2.

Parameter		Symbol	Typ	Max 125 °C	Unit	Test conditions
Supply current	Common circuit when enabling PVDn (n = 0 to 2) or Battery power supply switch in Deep Software Standby mode 1	I_{CC}	4.51	—	µA	—
	Common circuit when enabling PVDn (n = 0 to 2) or Battery power supply switch in Deep Software Standby mode 2		4.97	—		—
	PVD0 enabled with OFS1(_SEC).PVDLPSEL = 1		2.16	—		—
	PVD1 enabled		2.16	—		—
	PVD2 enabled		2.16	—		—
	Battery power supply switch enabled with following conditions.*1 • Battery power supply switch enable (VBTBPCR1.BPWSWSTP = 0), voltage monitor 0 reset enable (OFS1(_SEC).PVDAS = 0) and low power consumption function of PVD0 disable (OFS1(_SEC).PVDLPSEL = 1). • Battery power supply switch enable (VBTBPCR1.BPWSWSTP=0) and voltage monitor 0 reset disable (OFS1(_SEC).PVDAS = 1).		2.16	—		—

Note 1. Consumption current is not increased in other condition.

Table 60.21 Increase when the sub-clock oscillator and RTC are enabled in Deep Software Standby mode 1, 2 and 3.

Parameter			Symbol	Typ	Max 125 °C	Unit	Test conditions
Supply current	When a crystal oscillator is in use	Low Power mode 3	I_{CC}	0.22	—	µA	—
		Low Power mode 2		0.27	—		—
		Low Power mode 1		0.34	—		—
		Standard mode		0.67	—		—
	RTC is operating			0.33	—		—

Table 60.22 Inrush current

Parameter				Symbol	Typ	Max 125 °C	Unit	Test conditions
Supply current	Inrush current on returning from deep software standby mode	Inrush current of VCC_DCDC*1	DPSBYCR.DCSSMODE = 0	I_{RUSH}	—	630	mA	—
			DPSBYCR.DCSSMODE = 1		—	1020		—

Note 1. Reference value

Table 60.23 Operating current (Analog) (1 of 2)

Parameter				Symbol	Typ	Max 125 °C	Unit	Test conditions
Supply current *1	Oscillator	Main clock oscillator		I_{CC}	0.48	—	mA	MOMCR.MODRV0[2:0] = 000b
					0.58	—	mA	MOMCR.MODRV0[2:0] = 011b
					0.90	—	mA	MOMCR.MODRV0[2:0] = 101b
	Analog power supply current	During 12-bit A/D conversion		AI_{CC}	0.8	1.1	mA	—
		During 12-bit A/D conversion with S/H amp			2.3	3.3	mA	—
		ACMPHS(1unit)			100	150	µA	—
		Temperature sensor			0.1	0.2	mA	—
		During D/A conversion(per unit)	Without AMP output		0.1	0.2	mA	—
			With AMP output		0.8	1.6	mA	—
		Waiting for A/D, D/A conversion (all units)			0.9	1.6	mA	—
		ADC12, DAC12 in standby modes (all units)*2			2	8	µA	—
	Reference power supply current (VREFH0)	During 12-bit A/D conversion (unit 0)		AI_{REFH0}	70	120	µA	—
		Waiting for 12-bit A/D conversion (unit 0)			0.07	0.5	µA	—
		ADC12 in standby modes (unit 0)			0.07	0.5	µA	—
	Reference power supply current (VREFH)	During 12-bit A/D conversion (unit 1)		AI_{REFH}	70	120	µA	—
		During D/A conversion(per unit)	Without AMP output		0.1	0.4	mA	—
			With AMP output		0.1	0.4	mA	—
		Waiting for 12-bit A/D (unit 1), D/A (all units) conversion			0.07	0.8	µA	—
		ADC12 in standby modes (unit 1)			0.07	0.8	µA	—
	USB operating current	Low speed	USBFS	$I_{CCUSBLS}$	3.5	6.5	µA	VCC_USB
			USBHS		10.5	13.6	µA	VCC_USBHS = AVCC_USBHS (PHYSET.HSEB = 0)
			USBHS		4.4	6.0	mA	VCC_USBHS = AVCC_USBHS (PHYSET.HSEB = 1)
		Full speed	USBFS	$I_{CCUSBFS}$	4.0	10.0	mA	VCC_USB
			USBHS		11.4	13.7	mA	VCC_USBHS = AVCC_USBHS (PHYSET.HSEB = 0)
			USBHS		5.2	6.0	mA	VCC_USBHS = AVCC_USBHS (PHYSET.HSEB = 1)
		High speed	USBHS	$I_{CCUSBHS}$	45.7	51.4	mA	VCC_USBHS = AVCC_USBHS
		Standby mode (direct power down)	USBHS	$I_{CCUSBSBY}$	0.5	3.00	µA	VCC_USBHS = AVCC_USBHS

Table 60.23 Operating current (Analog) (2 of 2)

Parameter			Symbol	Typ	Max 125 °C	Unit	Test conditions
Supply current *1	MIPI operating current	DSI_ULP1	$I_{CC18MPI}$	1.5	2.5	mA	2 lanes PLL = OFF
		DSI_ULP2		1.5	2.5	mA	2 lanes PLL = ON
		DSI-LP		5.6	11.3	mA	2 lanes CL = 60pF
		DSI-HS		9.8	15.1	mA	2 lanes
		Standby		0.001	0.2	mA	Software standby mode, Deep Software standby mode
				1.0	3.1	mA	DPHYPWRCR.PWRSEN = 1'b0 in Normal mode, CPU Sleep mode or CPU Deep Sleep mode
				0.001	0.2	mA	DPHYPWRCR.PWRSEN = 1'b1 in Normal mode, CPU Sleep mode or CPU Deep Sleep mode
		DSI_ULP1	I_{CCMIPI}	1.4	6.5	mA	2 lanes PLL = OFF
		DSI_ULP2		4.0	10.1	mA	2 lanes PLL = ON
		DSI-LP		4.0	10.1	mA	2 lanes
		DSI-HS		9.0	24.2	mA	2 lanes
		Standby		0.001	0.1	mA	Software standby mode, Deep Software standby mode
				0.001	0.1	mA	DPHYPWRCR.PWRSEN = 1'b0 in Normal mode, CPU Sleep mode or CPU Deep Sleep mode
				1.0	2.0	mA	DPHYPWRCR.PWRSEN = 1'b1 in Normal mode, CPU Sleep mode or CPU Deep Sleep mode

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. When the MCU is in Software Standby mode or the MSTPCRD.MSTPD16 (12-Bit A/D Converter 0 Module Stop bit) and MSTPCRD.MSTPD15 (12-bit A/D converter 1 module stop bit) are in the module-stop state.

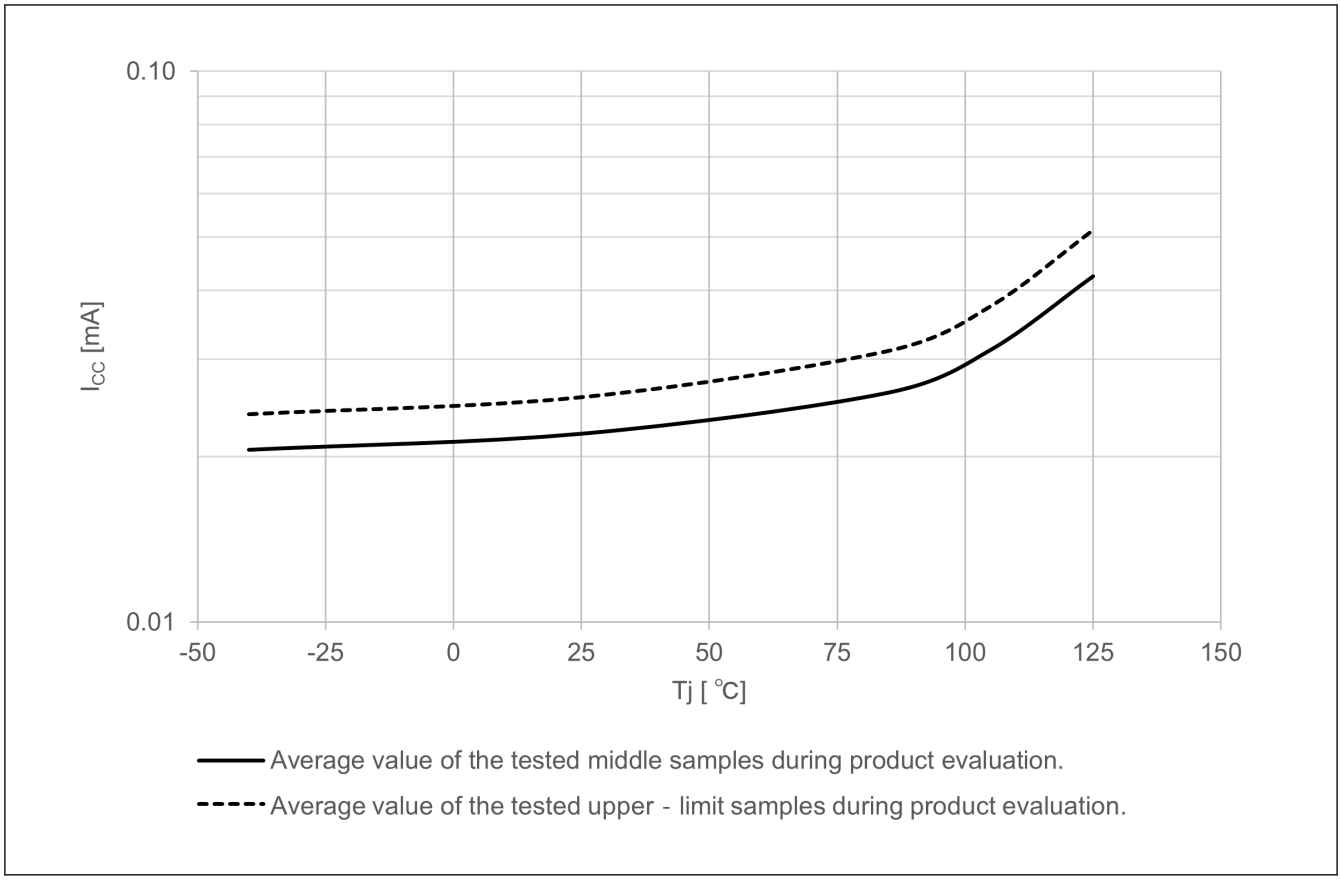


Figure 60.3 Temperature dependency in Software Standby mode (I_{CC}) (reference data)

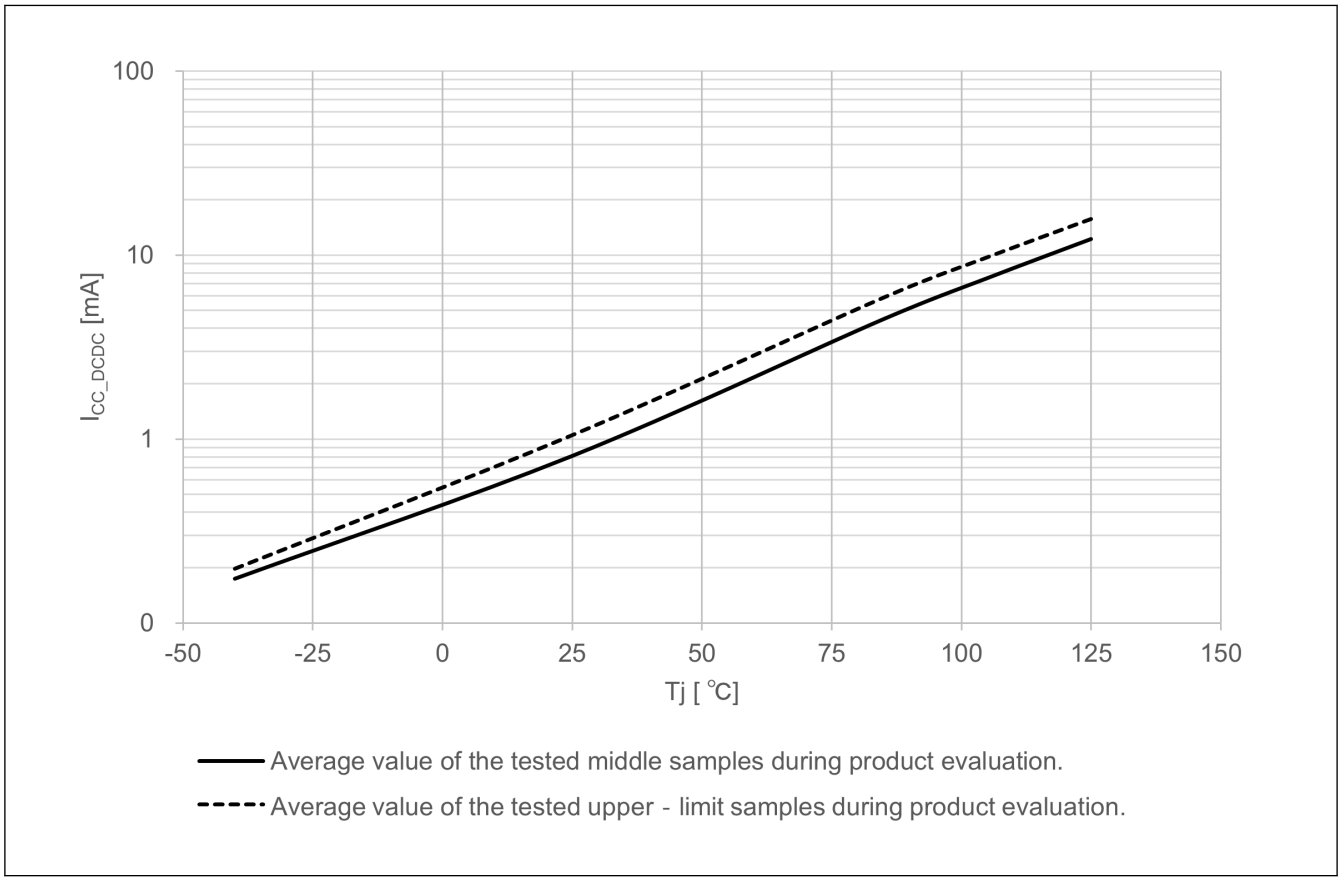


Figure 60.4 Temperature dependency in Software Standby mode (I_{CC_DCDC}) (reference data)

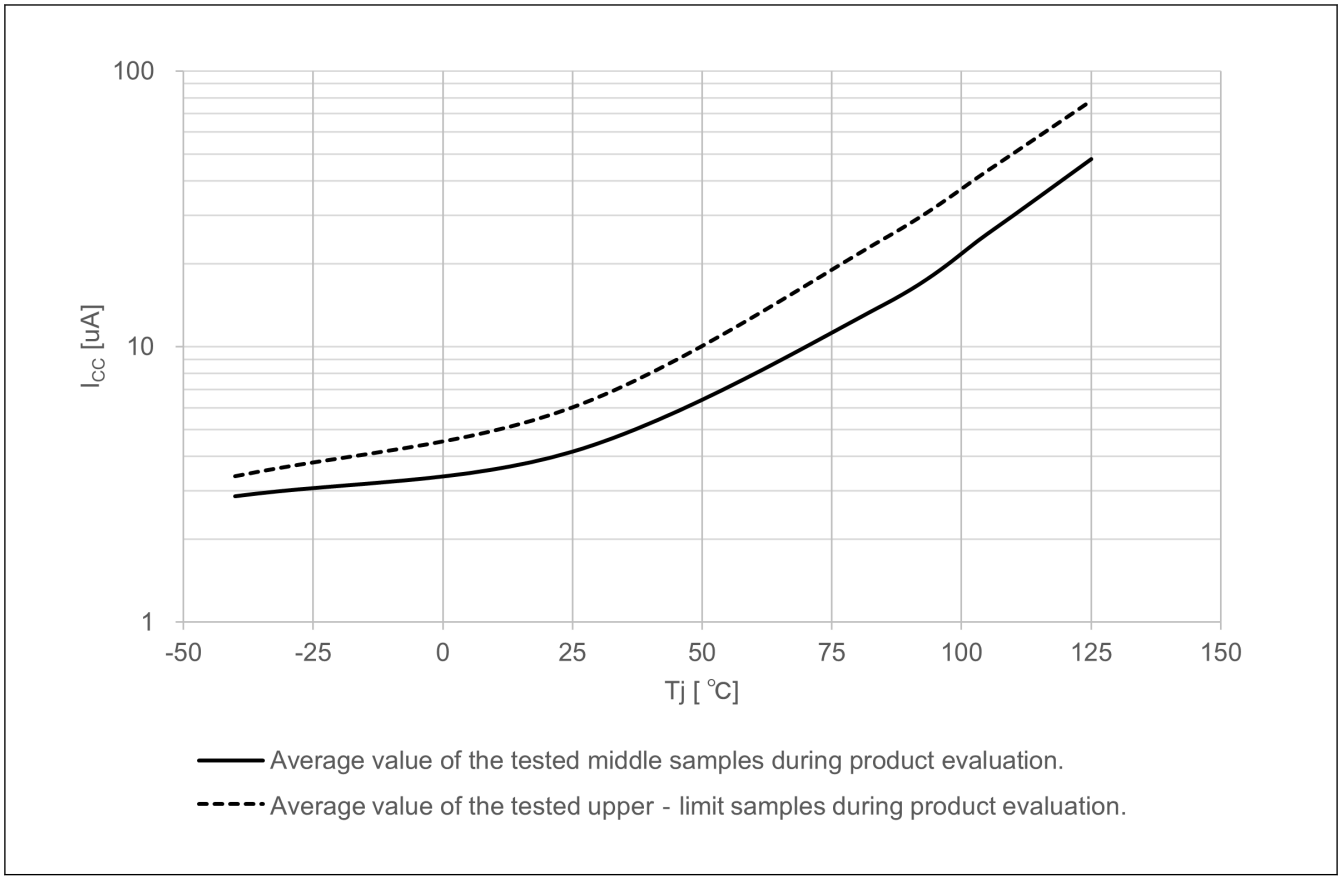


Figure 60.5 Temperature dependency in Deep Software Standby mode 1 (reference data)

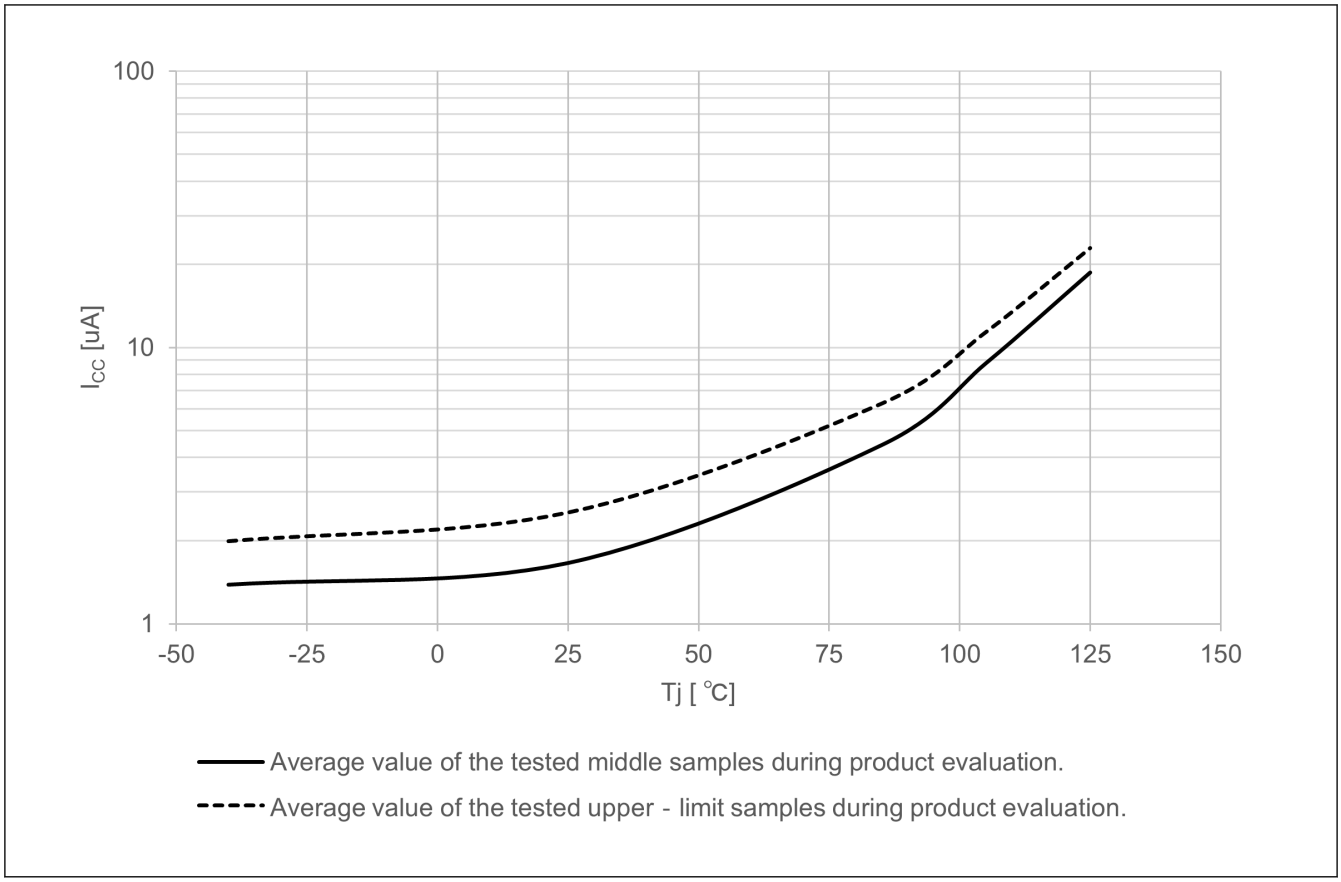


Figure 60.6 Temperature dependency in Deep Software Standby mode 2 (reference data)

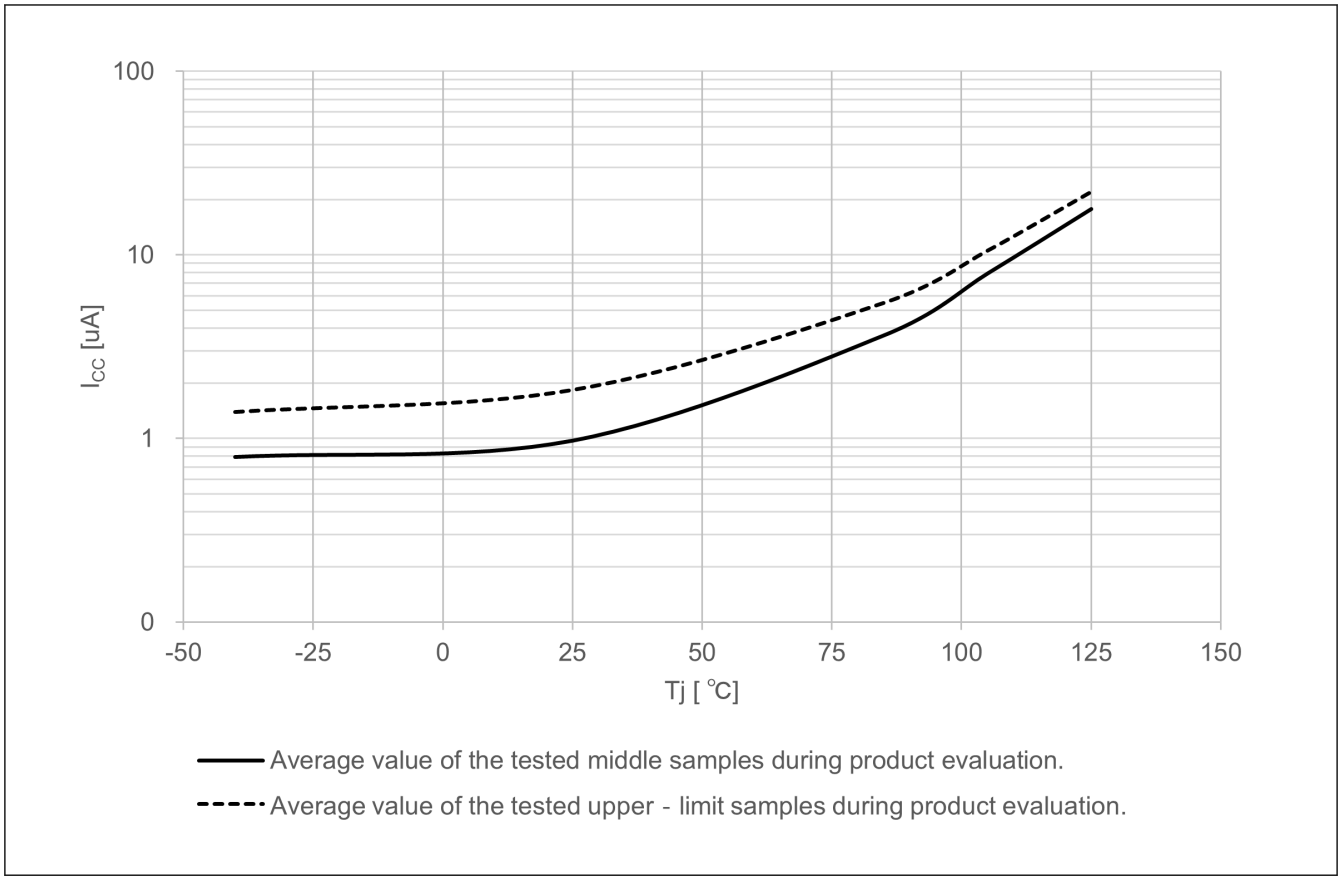


Figure 60.7 Temperature dependency in Deep Software Standby mode 3 (reference data)

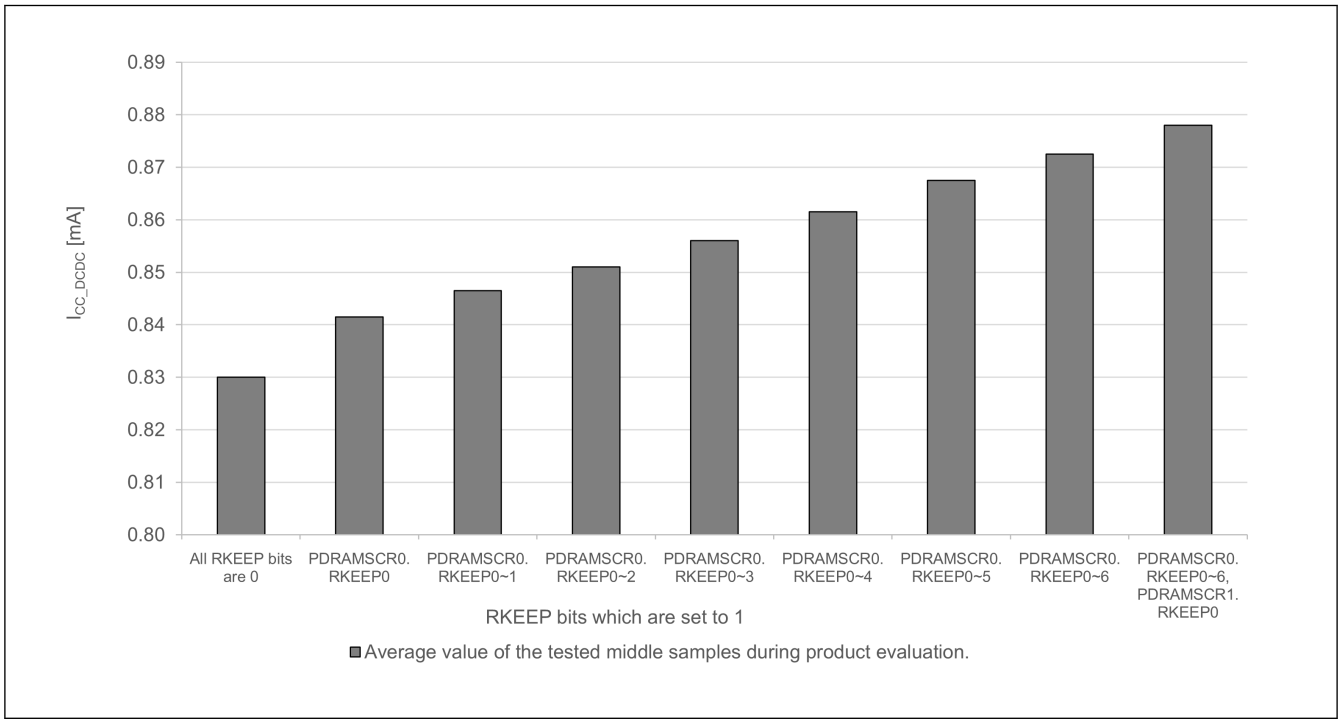


Figure 60.8 Software Standby current per SRAM state (reference data)

The more practical I_{CC_DCDC} value can be obtained with the following formula.

$I_{CC\_DCDC} = I_{DD} \times (VCL \div VCC) \div efficiency$

Where: VCL and VCC are the voltage of VCL pin and VCC pin respectively, and efficiency is shown in the following figures.

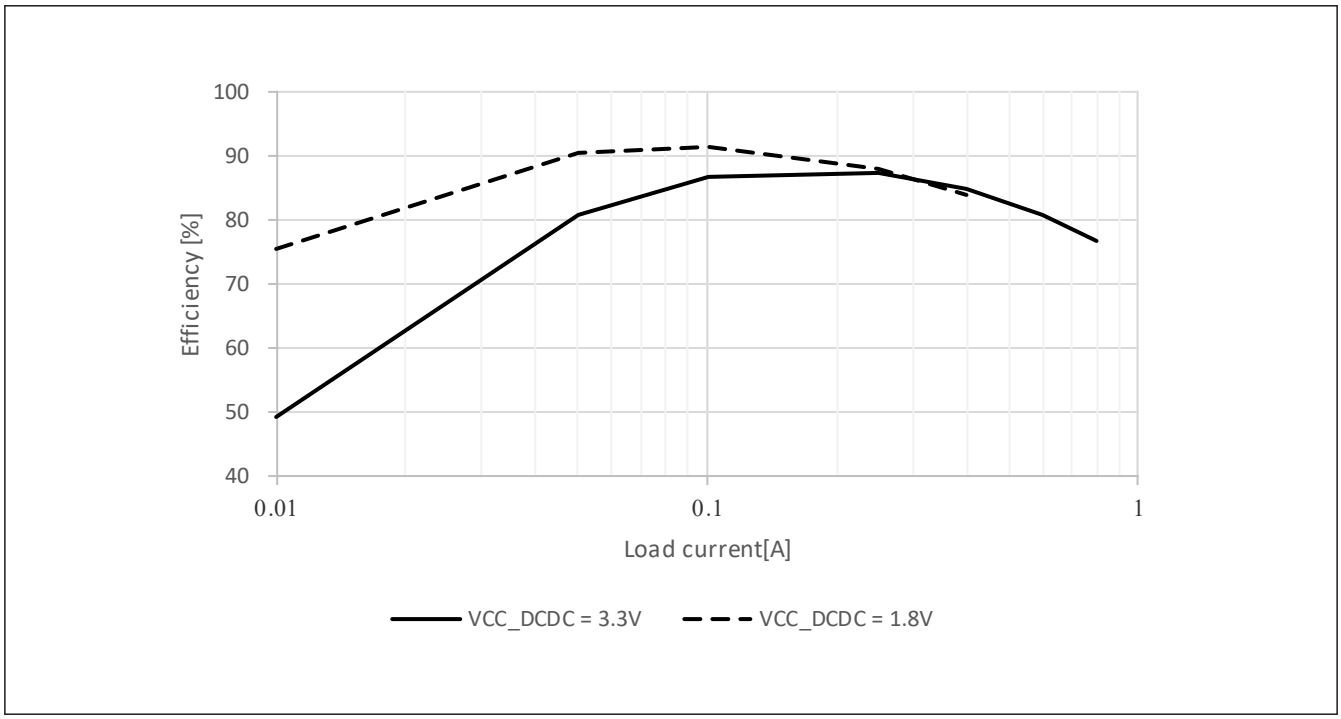


Figure 60.9 Typical DCDC efficiency (%) vs load current (A) in High-speed mode , Tj = 25 °C

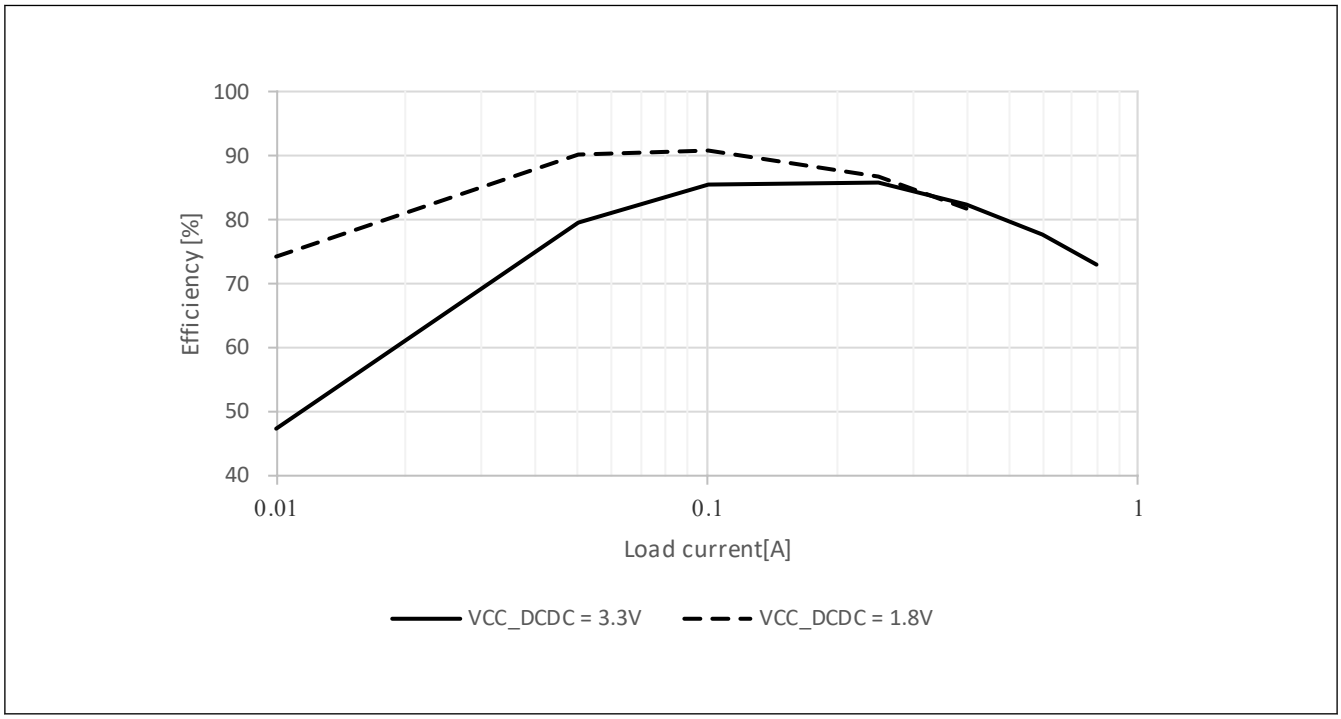


Figure 60.10 Typical DCDC efficiency (%) vs load current (A) in High-speed mode , Tj = 125 °C

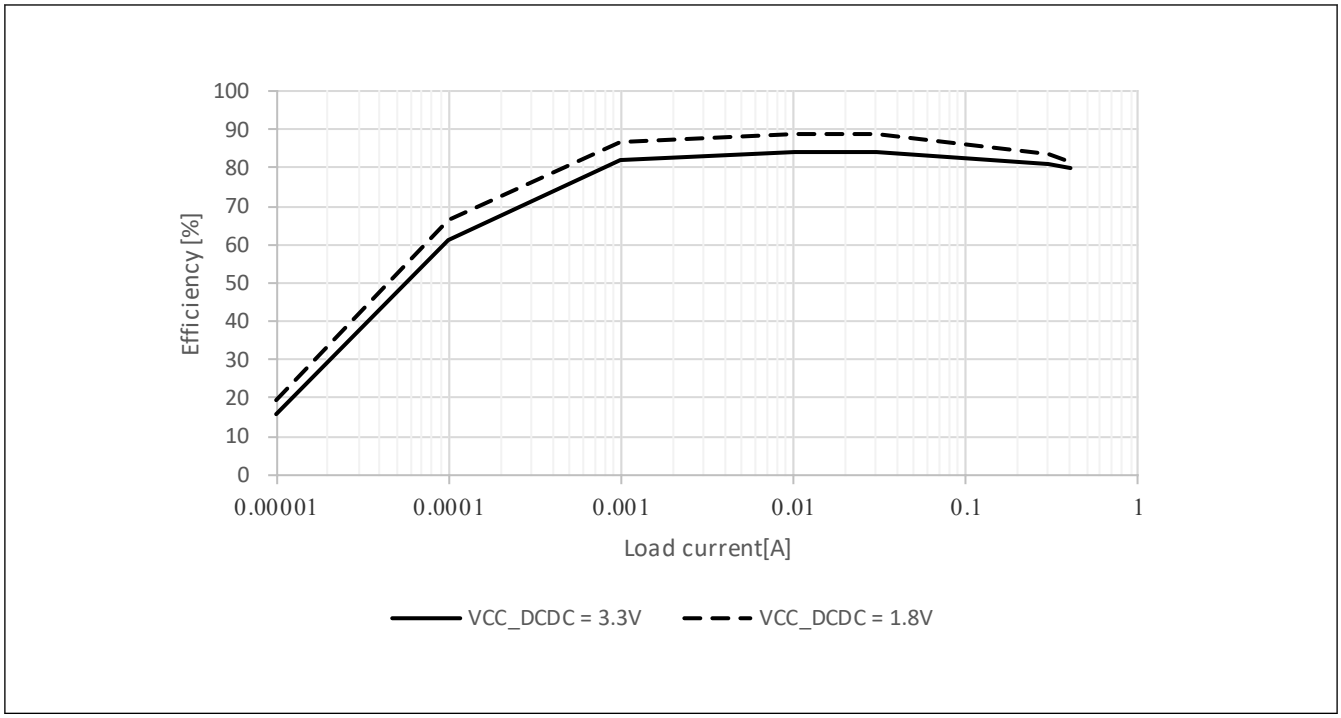


Figure 60.11 Typical DCDC efficiency (%) vs load current (A) in Low-speed mode and Software Standby mode, Tj = 25 °C

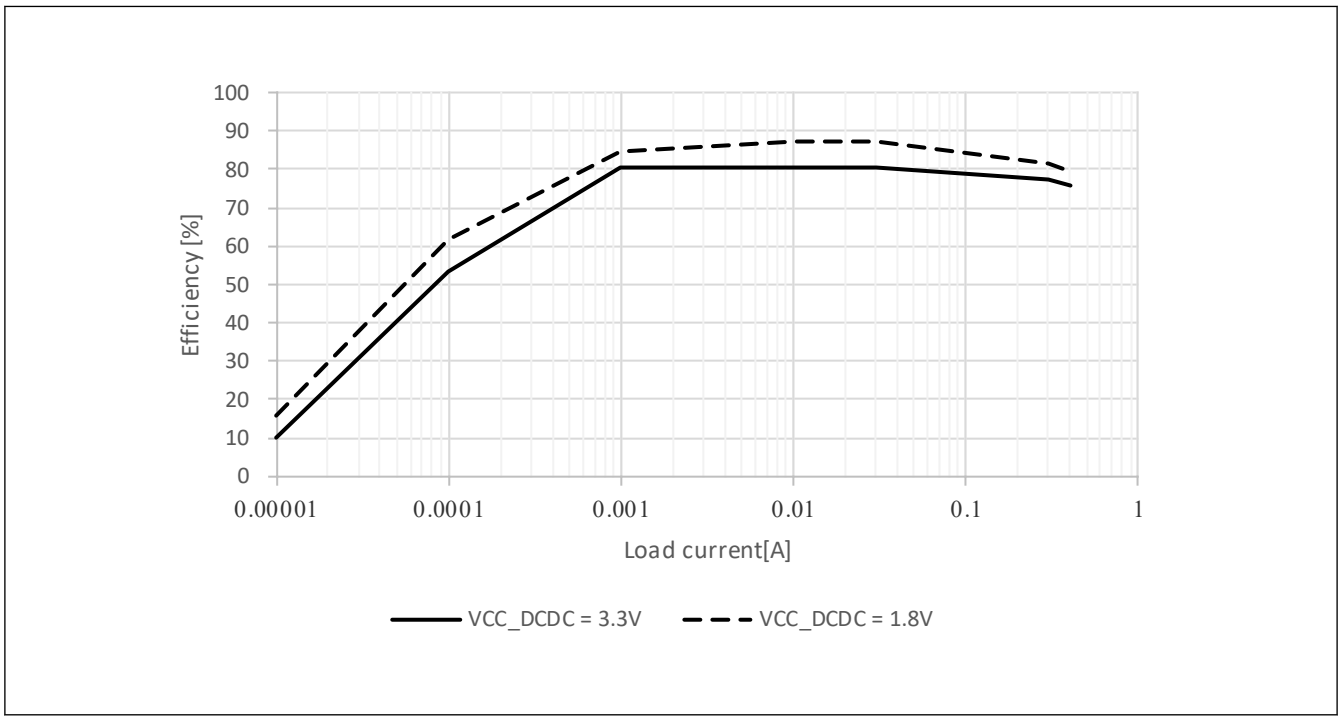


Figure 60.12 Typical DCDC efficiency (%) vs load current (A) in Low-speed mode and Software Standby mode, Tj = 125 °C

Note: DCDC efficiency is obtained based on the VCC_DCDC current.

60.2.6 VCC Rise and Fall Gradient and Ripple Frequency

Table 60.24 VCC rise and fall gradient characteristics at power on/off

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
VCC rising gradient at power on*1		SrVCC	0.0084	—	20	ms/V	—
VCC falling gradient at power off	VBATT function is disabled*1	SfVCC1	0.0084	—	—	ms/V	—
	VBATT function is enabled	SfVCC2	1.0000	—	—		—

Note 1. In case the VCC voltage crosses V_{POR1}.

Table 60.25 VCC ripple frequency and gradient characteristics during operation

The ripple voltage must meet the allowable ripple frequency $f_{r(VCC)}$ within the range between the VCC upper limit (3.6 V) and lower limit (1.68 V). When the VCC change exceeds VCC ±10%, the allowable voltage change rising and falling gradient dt/dVCC must be met.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Allowable ripple frequency	$f_{r\ (VCC)}$	—	—	10.0	kHz	Figure 60.13 $V_{r\ (VCC)} \leq VCC \times 0.2$
		—	—	1.0	MHz	Figure 60.13 $V_{r\ (VCC)} \leq VCC \times 0.08$
		—	—	10.0	MHz	Figure 60.13 $V_{r\ (VCC)} \leq VCC \times 0.06$
Allowable voltage change rising and falling gradient	dt/dVCC*1	1.0	—	—	ms/V	When VCC change exceeds VCC ±10%

Note 1. In case the VCC voltage does not cross V_{POR1}.

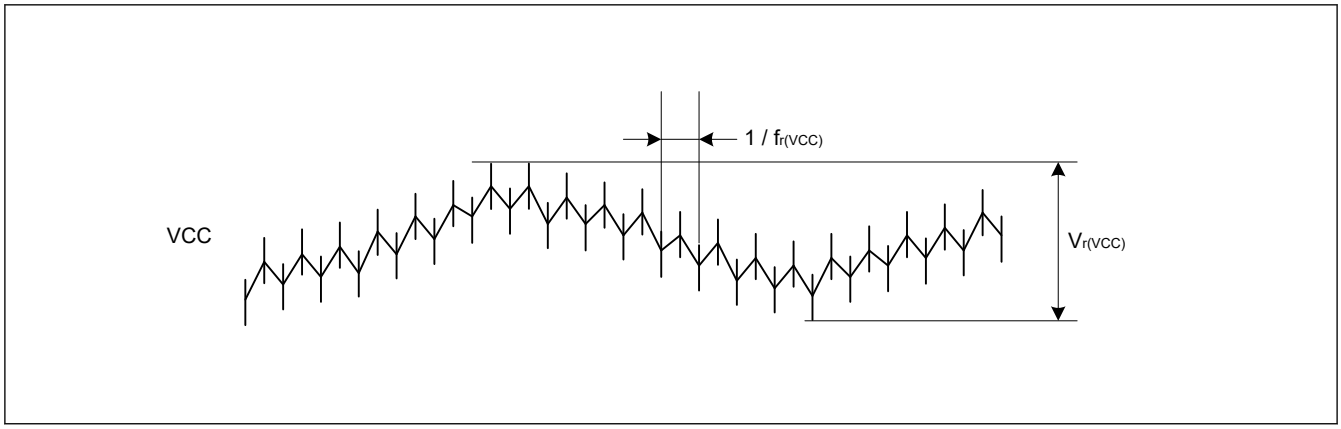


Figure 60.13 Ripple waveform

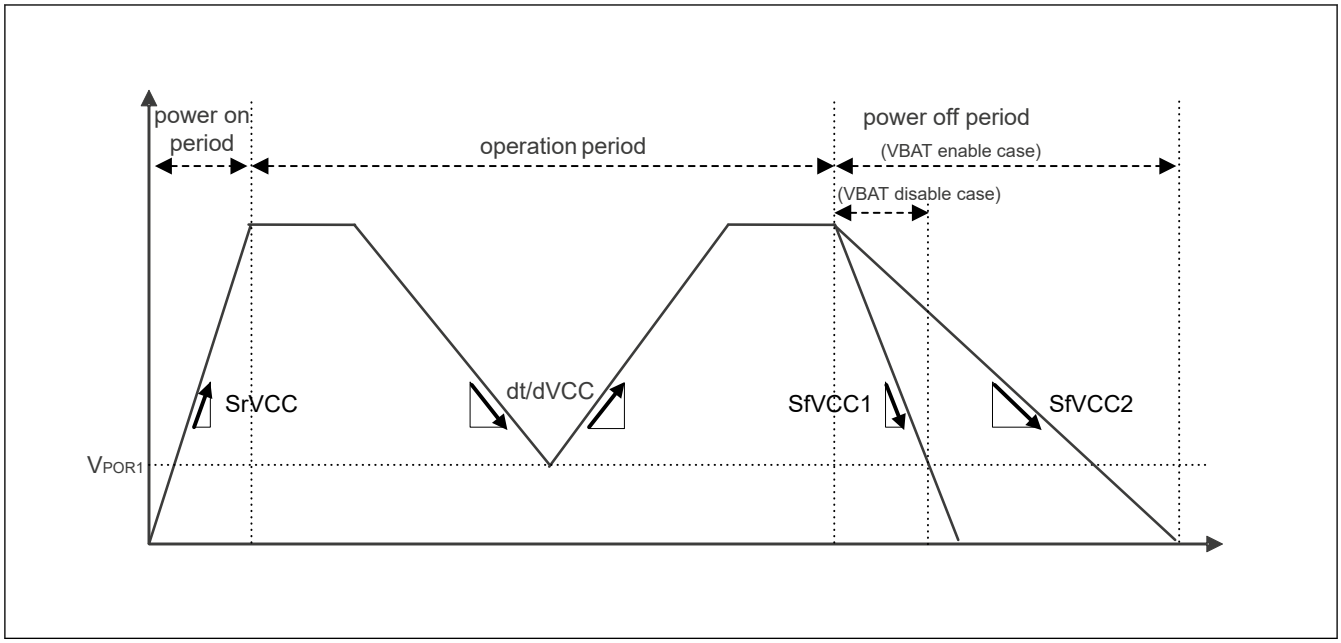


Figure 60.14 VCC rising and falling waveform

60.2.7 Thermal Characteristics

Maximum value of junction temperature (Tj) must not exceed the value of section 60.2.1. Tj/Ta Definition.

Tj is calculated by either of the following equations.

- Tj = Ta + θja × Total power consumption
- Tj = Tt + Ψjt × Total power consumption
  - Tj : Junction Temperature (°C)
  - Ta : Ambient Temperature (°C)
  - Tt : Top Center Case Temperature (°C)
  - θja : Thermal Resistance of “Junction”-to-“Ambient” (°C/W)
  - Ψjt : Thermal Resistance of “Junction”-to-“Top Center Case” (°C/W)
- Total power consumption = Voltage × (Leakage current + Dynamic current)
- Leakage current of IO = Σ (I_{OL} × V_{OL}) /Voltage + Σ (|I_{OH}| × |VCC – V_{OH}|) /Voltage
- Dynamic current of IO = Σ IO (C_{in} + C_{load}) × IO switching frequency × Voltage
  - C_{in}: Input capacitance
  - C_{load}: Output capacitance

Regarding θja and Ψjt, see Table 60.26.

Table 60.26 Thermal Resistance

Parameter	Package	Symbol	Value*1	Unit	Test conditions
Thermal Resistance	176-pin LQFP (PLQP0176KJ-A)	θja	32.0	°C/W	JESD 51-2 and 51-7 compliant
	224-pin BGA (PLBG0224GD-A)		21.5		JESD 51-2 and 51-9 compliant
	176-pin LQFP (PLQP0176KJ-A)	Ψjt	0.42	°C/W	JESD 51-2 and 51-7 compliant
	224-pin BGA (PLBG0224GD-A)		0.24		JESD 51-2 and 51-9 compliant

Note 1. The values are reference values when the 4-layer board is used. Thermal resistance depends on the number of layers or size of the board. For details, see the JEDEC standards.

60.2.7.1 Calculation guide of maximum current

Table 60.27 Power consumption of each unit (1 of 2)

Dynamic current/ Leakage current	MCU Domain	Category	Item	Symbol	Frequency [MHz]	Current [uA/MHz]	Current*1 [mA]	Condition
Leakage current	Analog	Regulator and Leak*1	Tj = 95°C	I_{CC}	—	—	0.54	—
			Tj = 105°C		—	—	0.64	
			Tj = 115°C		—	—	0.75	
			Tj = 125°C		—	—	0.85	
			Tj = 95°C	I_{CC_DCDC}	—	—	64	VCC_DCDC = 3.3V, High speed mode, PDCTRGD.PDDE = 0
			Tj = 105°C		—	—	77	
			Tj = 115°C		—	—	94	
			Tj = 125°C		—	—	111	
			Tj = 95°C		—	—	115	VCC_DCDC = 1.8V, High speed mode, PDCTRGD.PDDE = 0
			Tj = 105°C		—	—	138	
			Tj = 115°C		—	—	168	
			Tj = 125°C		—	—	199	
			Tj = 95°C	I_{DD}	—	—	150	PDCTRGD.PDDE = 0
			Tj = 105°C		—	—	180	
			Tj = 115°C		—	—	220	
			Tj = 125°C		—	—	260	
Dynamic current	CPU	Operation with Cache	CoreMark	I_{DD}	480	307	147	CPUCLK = 480MHz
	Peripheral Unit	Timer	RTC		60	1.299	0.078	—
			GPT16 (6ch)*2		120	16.988	2.039	
			GPT32 (8ch)*2		120	20.279	2.433	
			POEG (4 Groups)*2		60	1.363	0.082	
			AGT (2ch)*2		60	2.233	0.134	
			ULPT (2ch)*2		60	0.350	0.021	
			WDT		60	0.775	0.047	
			IWDT		60	0.100	0.006	

Table 60.27 Power consumption of each unit (2 of 2)

Dynamic current/ Leakage current	MCU Domain	Category	Item	Symbol	Frequency [MHz]	Current [uA/MHz]	Current*1 [mA]	Condition
Dynamic current	Peripheral Unit	Communication interfaces	ETHERC	I_{DD}	120	8.149	0.978	—
			USBFS		60	8.713	0.523	
			USBHS		120	11.911	1.429	
			SCI (6ch)*2		120	22.717	2.726	
			IIC (2ch)*2		60	2.867	0.172	
			I3C		120	15.274	1.833	
			CANFD (2ch)*2		120	9.050	1.086	
			SPI (2ch)*2		120	7.950	0.954	
			OSPI		60	40.796	2.448	
			SSIE (2ch)*2		60	6.818	0.409	
			SDHI (2ch)*2		60	16.742	1.005	
		Analog	ADC (2Units)*2		120	3.961	0.475	—
			DAC12 (2ch)*2		120	1.079	0.129	
			TSN		60	0.092	0.005	
			ACMPHS (2ch)*2		60	0.083	0.005	
		Human machine interfaces	GLCDC		120	44.530	5.344	—
			DRW		120	59.158	7.099	
			MIPI DSI		120	31.344	3.761	
			CEU		120	24.143	2.897	
		Event link	ELC		60	1.670	0.100	—
		Security	RSIP-E51A		120	311.301	37.4	—
			DOTF		60	63.393	3.804	
		Data processing	CRC		120	4.372	0.525	—
			DOC		120	0.427	0.051	
		System	CAC		60	0.738	0.044	—
		DMA	DMAC (per 1ch)		240	9.012	2.163	—
			DTC		240	11.175	2.682	
	FSBL operation				240	—	93.4	—
					120	—	72.9	—

Note 1. Regulator and Leak are Internal voltage regulator's current and MCU's leakage current.
It is selected according to the temperature of Tj.
Note 2. To determine the current consumption per channel or unit, divide Current [mA] by the number of channels, groups or units.

Table 60.28 Outline of operation for each unit (1 of 2)

Peripheral	Outline of operation
RTC	RTC is operating with LOCO.
GPT	Operating modes is set to saw-wave PWM mode.
POEG	Only clear module stop bit.
AGT	AGT is operating with PCLKB.
ULPT	ULPT is operating with LOCO.

Table 60.28 Outline of operation for each unit (2 of 2)

Peripheral	Outline of operation
WDT	WDT is operating with PCLKB.
IWDT	IWDT is operating with IWDTCLK.
ETHERC	Operation modes is set to full-duplex mode. ETHERC is operating using Reduced Media Independent Interface (RMII).
USBFS	Transfer types is set to bulk transfer. USBFS is operating using Full-speed transfer (12 Mbps).
USBHS	Transfer types is set to bulk transfer. USBHS is operating using High-speed transfer.
SCI	SCI is transmitting data in clock synchronous mode.
IIC	Communication format is set to I2C-bus format. IIC is transmitting data in master mode.
I3C	Communication format is set to I3C SDR format. I3C is transmitting data in master mode (12.5MHz).
CANFD	CANFD is transmitting and receiving data in self-test mode 1.
SPI	SPI mode is set to SPI operation (4-wire method). SPI master/slave mode is set to master mode. SPI is transmitting 32-bit width data.
OSPI	OSPI is issuing memory write command to HyperRAM.
SSIE	Communication mode is set to Master. System word length is set to 32 bits. Data word length is set to 20 bits. SSIE is transmitting data using I2S format.
SDHI	Transfer bus mode is set to 8-bit wide bus mode. SDHI is issuing CMD24 (single-block write).
ADC	Resolution is set to 12-bit accuracy. Data registers is set to A/D-converted value addition mode. ADC12 is converting the analog input in continuous scan mode.
DAC12	DAC12 is outputting the conversion result while updating the value of data register.
TSN	TSN is operating.
ACMPHS	ACMPHS is operating.
GLCDC	GLCDC is operating after writing data to CLUT.
DRW	DRW is doing rendering operation after sending data from SDRAM.
MIPI DSI	MIPI DSI is operating with HS mode using 2-lane. Data is input via GLCDC.
CEU	CEU is capturing data and transferring to the SRAM.
ELC	Only clear module stop bit.
RSIP-E51A	RSIP is doing self-test operation.
DOTF	DOTF is doing decryption with AES.
CRC	CRC is generating CRC code using 32-bit CRC32-C polynomial.
DOC	DOC is operating in data comparison mode.
CAC	Measurement target clocks is set to PCLKB. Measurement reference clocks is set to PCLKB. CAC is measuring the clock frequency accuracy.
DMAC	Bit length of transfer data is set to 32 bits. Transfer mode is set to block transfer mode. DMAC is transferring data from SRAM0 to SRAM0.
DTC	Bit length of transfer data is set to 32 bits. Transfer mode is set to block transfer mode. DTC is transferring data from SRAM0 to SRAM0.

60.2.7.2 Example of Tj calculation

Assumption :

- Package 224-pin BGA : θja = 21.5 °C/W
- Ta = 80 °C
- $I_{CC} + I_{CC\_DCDC}$ = 240 mA
- VCC = 3.5 V (VCC = VCC2 = AVCC0 = AVCC_USBHS = VCC_USB = VCC_USBHS)
- I_{OH} = 1 mA, V_{OH} = VCC – 0.5 V, 12 Outputs

- I_{OL} = 20 mA, V_{OL} = 1.0 V, 8 Outputs
- I_{OL} = 1 mA, V_{OL} = 0.5 V, 12 Outputs
- C_{in} = 8 pF, 32 pins, Input frequency = 10 MHz
- C_{load} = 30 pF, 32 pins, Output frequency = 10 MHz

Static current of IO = Σ (VOL × IOL) / Voltage + Σ ((VCC - VOH) × IOH) / Voltage
= (20 mA × 1 V) × 8 / 3.5 V + (1 mA × 0.5 V) × 12 / 3.5 V + ((VCC - (VCC - 0.5 V)) × 1 mA) × 12 / 3.5 V
= 45.7 mA + 1.71 mA + 1.71 mA
= 49.1 mA
Dynamic current of IO = Σ IO (Cin + Cload) × IO switching frequency × Voltage
= ((8 pF × 32) × 10 MHz + (30 pF × 32) × 10 MHz) × 3.5 V
= 42.6 mA
Total power consumption = Voltage × (Static current + Dynamic current)
= (240 mA × 3.5 V) + (49.1 mA + 42.6 mA) × 3.5 V
= 1161 mW (1.161 W)
Tj = Ta + θja × Total power consumption
= 80 °C + 21.5 °C/W × 1.161W
= 105.0 °C

60.3 AC Characteristics

60.3.1 Frequency

Table 60.29 Operation frequency value in high-speed mode

Parameter			Symbol	Min	Typ	Max	Unit
Operation frequency	CPU clock (CPUCLK)*2	DCDC, BGA package, Tj ≤ 105℃*4	f	—	—	480	MHz
		DCDC, BGA package, Tj ≤ 125℃*4		—	—	400	
		External VDD, BGA package, Tj ≤ 105℃		—	—	480	
		External VDD, BGA package, Tj ≤ 125℃		—	—	400	
		DCDC, 176-pin LQFP package, Tj ≤ 125℃*4		—	—	400	
		External VDD, 176-pin LQFP package, Tj ≤ 125℃		—	—	400	
	System clock (ICLK)*2			—	—	240	
	Peripheral module clock (PCLKA)*2			—	—	120	
	Peripheral module clock (PCLKB)*2			—	—	60	
	Peripheral module clock (PCLKC)*2			—*3	—	60	
	Peripheral module clock (PCLKD)*2			—	—	120	
	Peripheral module clock (PCLKE)*2			—	—	240	
	Flash interface clock (FCLK)*2			—*1	—	60	
	External bus clock (BCLK)*2	VCC ≥ 2.7 V		—	—	120	
		VCC ≥ 1.68 V		—	—	60	
	EBCLK pin output	VCC ≥ 2.7 V		—	—	60	
		VCC ≥ 1.68 V		—	—	30	
	SDCLK pin output	VCC ≥ 3.0 V		—	—	120	
	SCI clock (SCICLK)			—	—	120	
	SPI clock (SPICLK)			—	—	120	
	Octal SPI clock (OCTACLK)			—	—	200	
	CANFD core clock (CANFDCLK)			—	—	80	
	LCD clock (LCDCLK)			—	—	240	
	USB clock (USBCLK)			—	—	48	
	USB clock (USB60CLK)			—	—	60	
	I3C clock (I3CCLK)			—	—	200	

Note 1. FCLK must run at a frequency of at least 4 MHz when programming or erasing the flash memory.
Note 2. See section 8, Clock Generation Circuit for the relationship between the CPUCLK, ICLK, PCLKA, PCLKB, PCLKC, PCLKD, PCLKE, FCLK, and BCLK frequencies.
Note 3. When the ADC12 is used, the PCLKC frequency must be at least 1 MHz.
Note 4. When DCDC is used with VCC_DCDC < 2.5V, I_{DD} current must be less than the value specified in operating current. Please see Table 60.8.

Table 60.30 Operation frequency value in low-speed mode

Parameter			Symbol	Min	Typ	Max	Unit
Operation frequency	CPU clock (CPUCLK)*2		f	—	—	1	MHz
	System clock (ICLK)*2			—	—	1	
	Peripheral module clock (PCLKA)*2			—	—	1	
	Peripheral module clock (PCLKB)*2			—	—	1	
	Peripheral module clock (PCLKC)*2			—*3	—	1	
	Peripheral module clock (PCLKD)*2			—	—	1	
	Peripheral module clock (PCLKE)*2			—	—	1	
	Flash interface clock (FCLK) *2			—*1	—	1	
	External bus clock (BCLK)*2			—	—	1	
	EBCLK pin output			—	—	1	
	SDCLK pin output	VCC ≥ 3.0 V		—	—	1	
	SCI clock (SCICLK)			—	—	1	
	SPI clock (SPICLK)			—	—	1	
	Octal SPI clock (OCTACLK)			—	—	1	
	CANFD core clock (CANFDCLK)			—	—	1	
	LCD clock (LCDCLK)			—	—	1	
	USB clock (USBCLK)			—	—	1	
	USB clock (USB60CLK)			—	—	1	
	I3C clock (I3CCLK)			—	—	1	

Note 1. Programming or erasing the flash memory is disabled in low-speed mode.
Note 2. See section 8, Clock Generation Circuit for the relationship between the CPUCLK, ICLK, PCLKA, PCLKB, PCLKC, PCLKD, PCLKE, FCLK, and BCLK frequencies.
Note 3. When the ADC12 is used, the PCLKC frequency must be at least 1 MHz.

60.3.2 Clock Timing

Table 60.31 Clock timing except for sub-clock oscillator (1 of 2)

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
EBCLK pin output cycle time	VCC = 2.70 V or above	t_{Bcyc}	16.6	—	—	ns	Figure 60.15
	VCC = 1.68 V or above		33.3	—	—		
EBCLK pin output high pulse width	VCC = 2.70 V or above	t_{CH}	3.3	—	—	ns	
	VCC = 1.68 V or above		9.6	—	—		
EBCLK pin output low pulse width	VCC = 2.70 V or above	t_{CL}	3.3	—	—	ns	
	VCC = 1.68 V or above		9.6	—	—		
EBCLK pin output rise time	VCC = 2.70 V or above	t_{Cr}	—	—	5.0	ns	
	VCC = 1.68 V or above		—	—	7.0		
EBCLK pin output fall time	VCC = 2.70 V or above	t_{Cf}	—	—	5.0	ns	
	VCC = 1.68 V or above		—	—	7.0		
SDCLK pin output cycle time		t_{SDcyc}	8.33	—	—	ns	
SDCLK pin output high pulse width		t_{CH}	1.0	—	—	ns	
SDCLK pin output low pulse width		t_{CL}	1.0	—	—	ns	
SDCLK pin output rise time		t_{Cr}	—	—	3.0	ns	
SDCLK pin output fall time		t_{Cf}	—	—	3.0	ns	

Table 60.31 Clock timing except for sub-clock oscillator (2 of 2)

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
EXTAL external clock input cycle time		t_{EXcyc}	20.80	—	—	ns	Figure 60.16
EXTAL external clock input high pulse width		t_{EXH}	5.30	—	—	ns	
EXTAL external clock input low pulse width		t_{EXL}	5.30	—	—	ns	
EXTAL external clock rise time		t_{EXr}	—	—	3.0	ns	
EXTAL external clock fall time		t_{EXf}	—	—	3.0	ns	
Main clock oscillator frequency		f_{MAIN}	8	—	48	MHz	—
Main clock oscillation stabilization wait time (crystal)*1		$t_{MAINOSCWT}$	—	—	—*1	ms	Figure 60.17
LOCO clock oscillation frequency		f_{LOCO}	29.4912	32.768	36.0448	kHz	—
LOCO clock oscillation stabilization wait time		t_{LOCOWT}	—	—	60.4	µs	Figure 60.18
MOCO clock oscillation frequency		f_{MOCO}	6.8	8.0	9.2	MHz	—
MOCO clock oscillation stabilization wait time		t_{MOCOWT}	—	—	15.0	µs	—
HOCO clock oscillator oscillation frequency	Without FLL	f_{HOCO16}	15.78	16.00	16.22	MHz	–20 ≤ Tj ≤ 125 °C
		f_{HOCO18}	17.75	18.00	18.25		
		f_{HOCO20}	19.72	20.00	20.28		
		f_{HOCO32}	31.55	32.00	32.45		
		f_{HOCO48}	47.33	48.00	48.67		
		f_{HOCO16}	15.71	16.00	16.29		–40 ≤ Tj ≤ –20 °C
		f_{HOCO18}	17.68	18.00	18.32		
		f_{HOCO20}	19.64	20.00	20.36		
		f_{HOCO32}	31.42	32.00	32.58		
		f_{HOCO48}	47.14	48.00	48.86		
	With FLL	f_{HOCO16}	15.960	16.000	16.040		–40 ≤ Tj ≤ 125 °C Sub-clock frequency accuracy is ±50 ppm.
		f_{HOCO18}	17.955	18.000	18.045		
		f_{HOCO20}	19.950	20.000	20.050		
		f_{HOCO32}	31.920	32.000	32.080		
		f_{HOCO48}	47.880	48.000	48.120		
HOCO clock oscillation stabilization wait time*2		t_{HOCOWT}	—	—	64.7	µs	—
HOCO stop width time		$t_{HOCOSTP}$	1	—	—	µs	Figure 60.21
HOCO period jitter		Jitter	-3	—	3	%	—
FLL stabilization wait time		t_{FLLWT}	—	—	1.8	ms	—
PLL1/PLL2 VCO frequency		f_{VCO}	640	—	1440.0	MHz	—
PLL1/PLL2 Output frequency for output clock P		t_{PLL}	40	—	480	MHz	—
PLL1/PLL2 Output frequency for output clock Q, R		t_{PLL}	71	—	480	MHz	—
PLL1/PLL2 clock oscillation stabilization wait time		t_{PLLWT}	—	—	40	µs	Figure 60.19
PLL1/PLL2 period jitter		—	—	—	±70	ps	—
PLL1/PLL2 long term jitter		—	—	—	±300	ps	Term: 1 µs, 10 µs

Note 1. When setting up the main clock oscillator, ask the oscillator manufacturer for an oscillation evaluation, and use the results as the recommended oscillation stabilization time. Set the MOSCWTCR register to a value equal to or greater than the recommended value.
After changing the setting in the MOSCCR.MOSTP bit to start main clock operation, read the OSCSF.MOSCSF flag to confirm that it is 1, and then start using the main clock oscillator.

Note 2. This is the time from release from reset state until the HOCO oscillation frequency (f_{HOCO}) reaches the range for guaranteed operation.

Table 60.32 Clock timing for the sub-clock oscillator

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Sub-clock frequency	f_{SUB}	—	32.768	—	kHz	—
Sub-clock oscillation stabilization wait time	$t_{SUBOSCWT}$	—	—	—*1	s	Figure 60.20

Note 1. When setting up the sub-clock oscillator, ask the oscillator manufacturer for an oscillation evaluation and use the results as the recommended oscillation stabilization time.
After changing the setting in the SOSCCR.SOSTP bit to start sub-clock operation, only start using the sub-clock oscillator after the sub-clock oscillation stabilization time elapses with an adequate margin. A value that is two times the value shown is recommended.

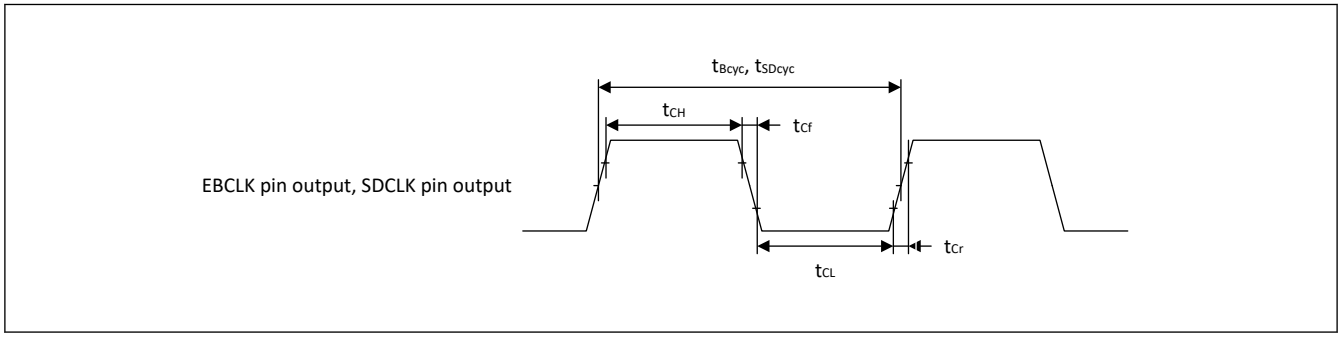


Figure 60.15 EBCLK and SDCLK output timing

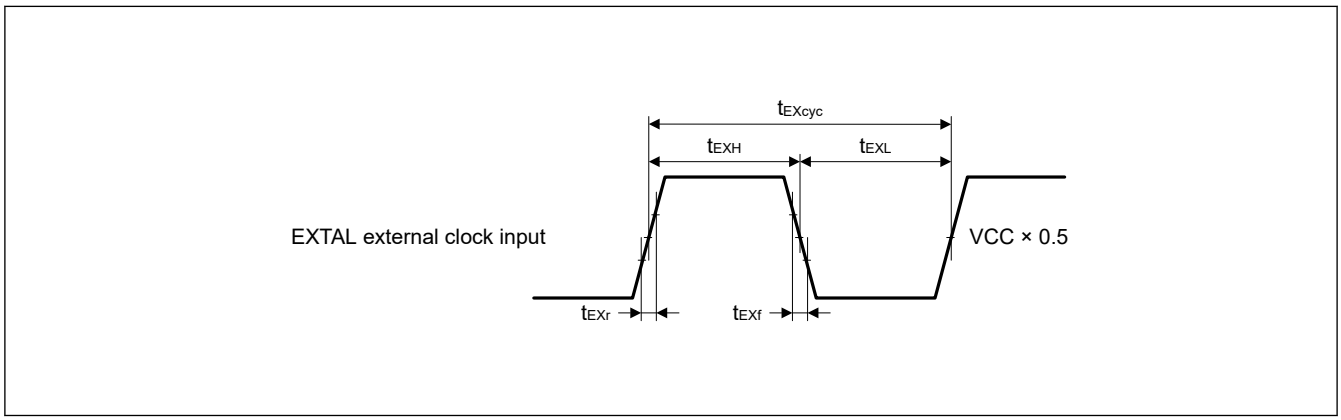


Figure 60.16 EXTAL external clock input timing

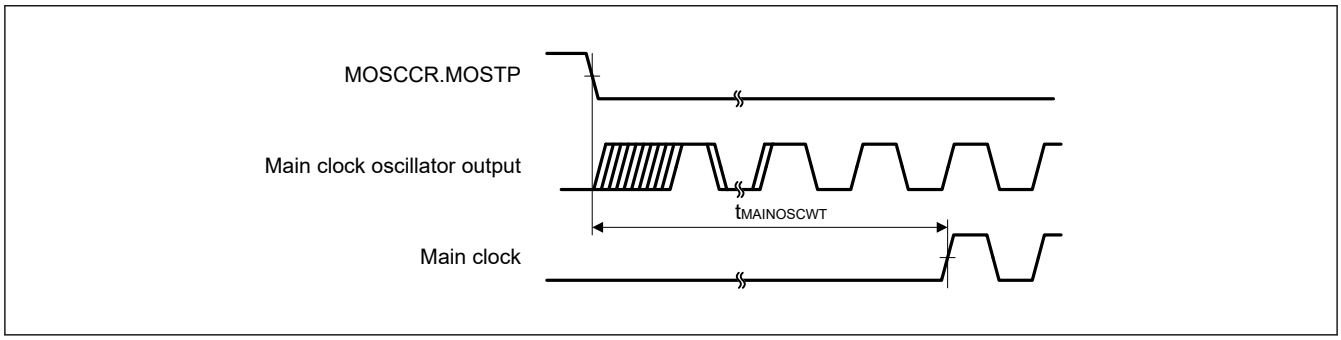


Figure 60.17 Main clock oscillation start timing

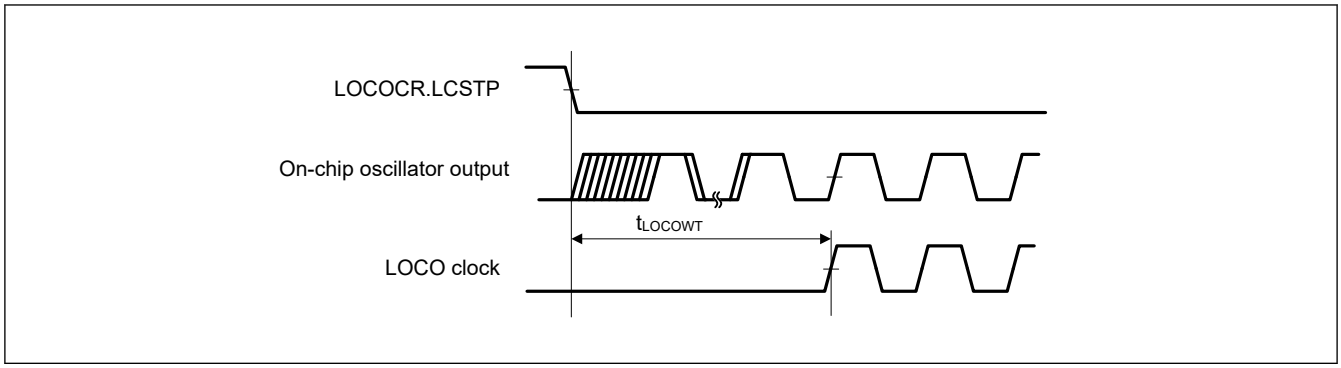
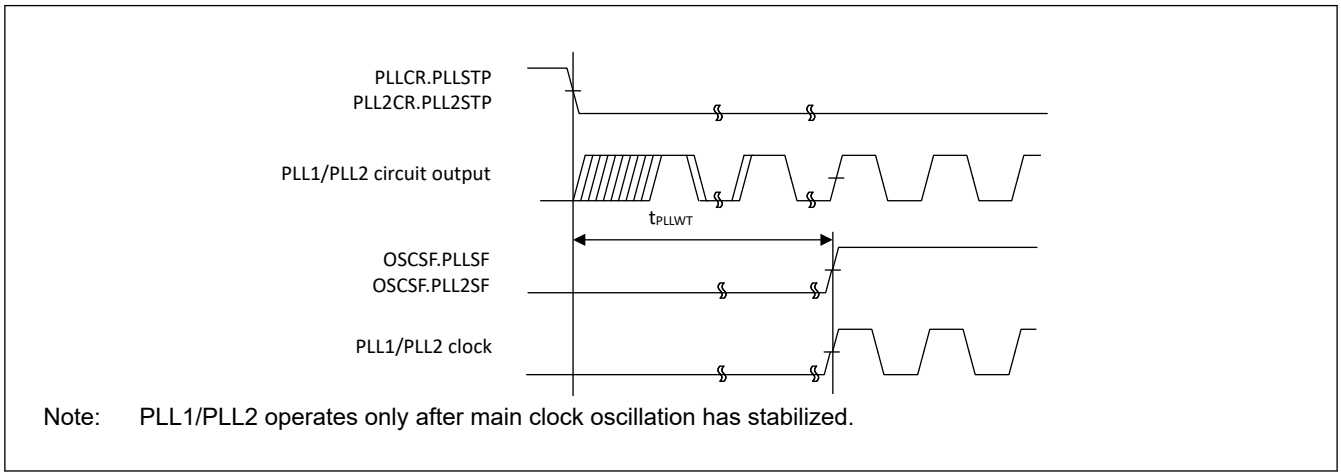


Figure 60.18 LOCO clock oscillation start timing



Note: PLL1/PLL2 operates only after main clock oscillation has stabilized.

Figure 60.19 PLL1/PLL2 clock oscillation start timing

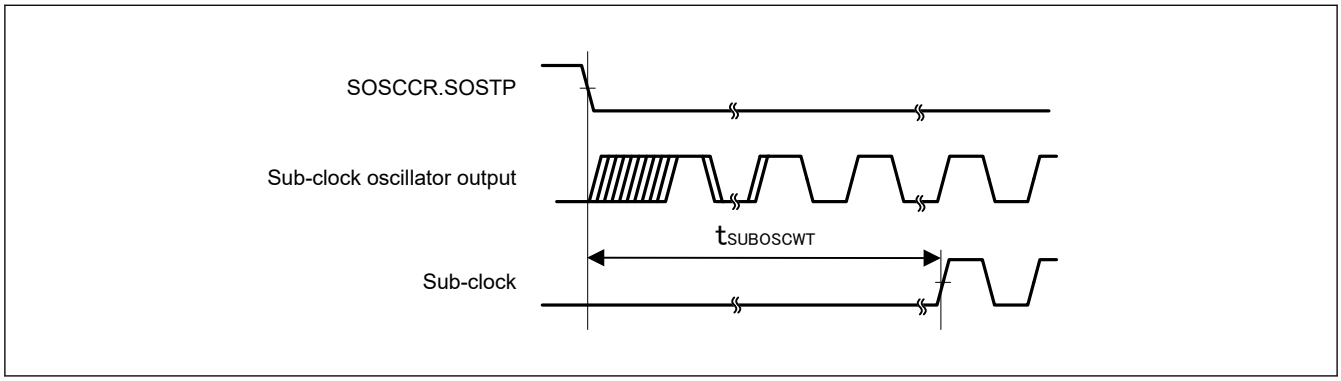


Figure 60.20 Sub-clock oscillation start timing

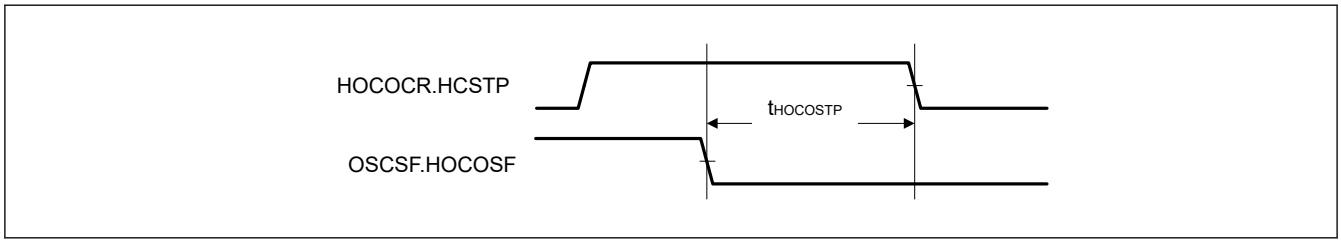


Figure 60.21 HOCO stop width time

60.3.3 Reset Timing

Table 60.33 Reset timing

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
RES pulse width	Power-on		t_{RESWP}	4.2	—	—	ms	Figure 60.22
	Deep Software Standby mode 1	DPSBYCR.DCSSMODE = 0	t_{RESWD}	1.30	—	—	ms	Figure 60.23
		DPSBYCR.DCSSMODE = 1		0.71	—	—		
	Deep Software Standby mode 2	DPSBYCR.DCSSMODE = 0		2.00	—	—		
		DPSBYCR.DCSSMODE = 1		1.50	—	—		
	Deep Software Standby mode 3	DPSBYCR.DCSSMODE = 0		3.50	—	—		
		DPSBYCR.DCSSMODE = 1		2.90	—	—		
	Software Standby mode		t_{RESWS}	0.66	—	—	ms	
	Low-speed Mode		t_{RESWLS}	0.46	—	—	ms	
	CPU Deep Sleep mode (SOSC operation)		$t_{RESWSODS}$	0.36	—	—	ms	
	CPU Deep Sleep mode (Other than SOSC operation)		t_{RESWDS}	0.24	—	—	ms	
	SOSC operation	PDCTRGD.PDDE = 1	t_{RESWSO}	0.19	—	—	ms	
		PDCTRGD.PDDE = 0		0.15	—	—		
	Other than above	PDCTRGD.PDDE = 1	t_{RESW}	62.0	—	—	µs	
		PDCTRGD.PDDE = 0		25.5	—	—		
Wait time after RES cancellation			t_{RESWT}	—	54.9	64.6	µs	Figure 60.22
Wait time after internal reset cancellation (IWDT reset, WDT reset, CPU Lockup reset, Bus Error reset, Common Memory Error reset, Software reset)			t_{RESW2}	—	54.9	64.6	µs	—

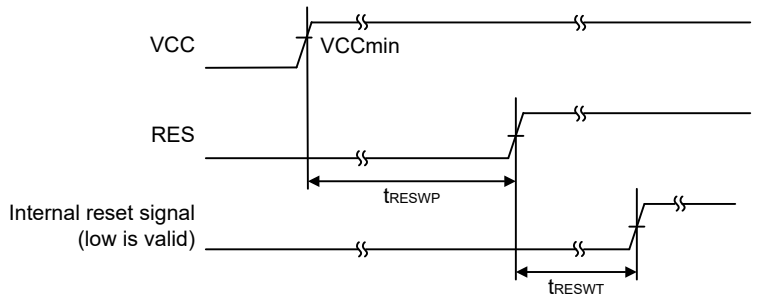


Figure 60.22 RES pin input timing under the condition that VCC exceeds V_{POR} voltage threshold

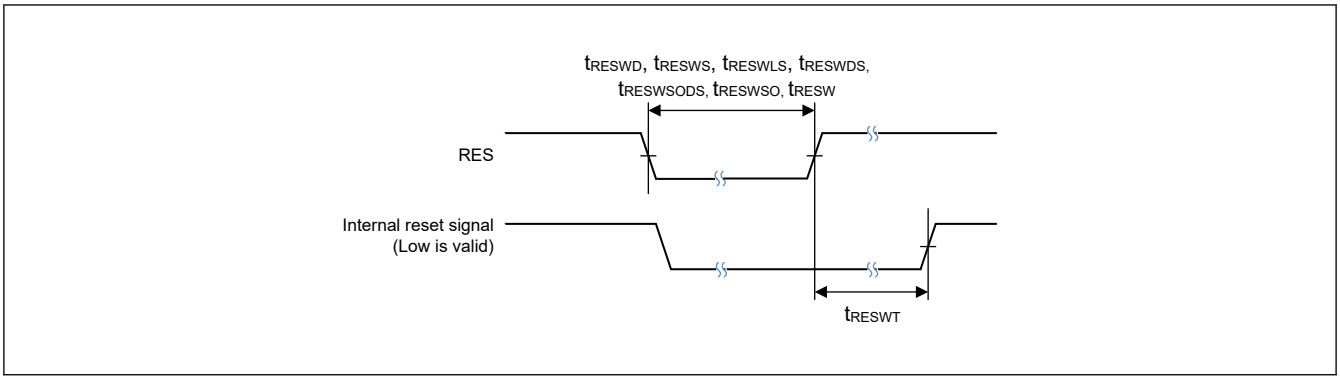


Figure 60.23 Reset input timing

60.3.4 Wakeup Timing

Table 60.34 Timing of recovery from low power modes (1 of 2)

Parameter			Fast return function*9	Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from CPU Deep Sleep mode			—	t_{DSLP}*11	—	182	214	µs	—
Recovery time from Software Standby mode	Crystal resonator connected to main clock oscillator	System clock source is main clock oscillator*1 MOSCSCR.MOSCSOKP = 0	Enabled	t_{SBYMC}*10	—	2.33	2.43	ms	Figure 60.24 The division ratio of all oscillators is 1.
		System clock source is main clock oscillator*1 MOSCSCR.MOSCSOKP = 1	Enabled		—	310	385	µs	
		System clock source is PLL1P with main clock oscillator*2 MOSCSCR.MOSCSOKP = 0	Enabled	t_{SBYPC}*10	—	2.47	2.59	ms	
		System clock source is PLL1P with main clock oscillator*2 MOSCSCR.MOSCSOKP = 1	Enabled		—	388	511	µs	
	External clock input to main clock oscillator	System clock source is main clock oscillator*3	Enabled	t_{SBYEX}*10	—	310	385	µs	
		System clock source is PLL1P with main clock oscillator*4	Enabled	t_{SBYPE}*10	—	388	511		
	System clock source is sub-clock oscillator*5		Enabled	t_{SBYSC}*10	—	0.81	0.87	ms	
	System clock source is HOCO clock oscillator*6		Enabled	t_{SBYHO}*10	—	310	385	µs	
	System clock source is PLL1P with HOCO*7		Enabled	t_{SBYPH}*10	—	398	522	µs	
	System clock source is MOCO clock oscillator*8		Enabled	t_{SBYMO}*10	—	312	387	µs	

Table 60.34 Timing of recovery from low power modes (2 of 2)

Parameter			Fast return function*9	Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from Deep Software Standby mode	Deep Software Standby mode 1	Any of PVD0(OFS1(_SEC).PVDLPSEL=1), PVD1, PVD2, or Battery power supply switch is enabled	Standard	t_{DSBY}	—	0.68	1.20	ms	Figure 60.25
			Fast		—	0.29	0.62	ms	
		All of PVD0(OFS1(_SEC).PVDLPSEL=1), PVD1, PVD2, and Battery power supply switch are disabled	Standard			0.73	1.30		
			Fast			0.33	0.71		
	Deep Software Standby mode 2	DPSWCR.WSTS = 0x0B	Standard		—	0.73	1.10	ms	
			Fast		—	0.33	0.50	ms	
		DPSWCR.WSTS = 0x9A	Standard		—	1.60	2.00	ms	
			Fast		—	1.20	1.50	ms	
	Deep Software Standby mode 3		Standard		—	2.10	3.50	ms	
			Fast		—	1.70	2.90	ms	
Wait time after cancellation of Deep Software Standby mode			—	t_{DSBYWT}	47.7	—	64.6	µs	

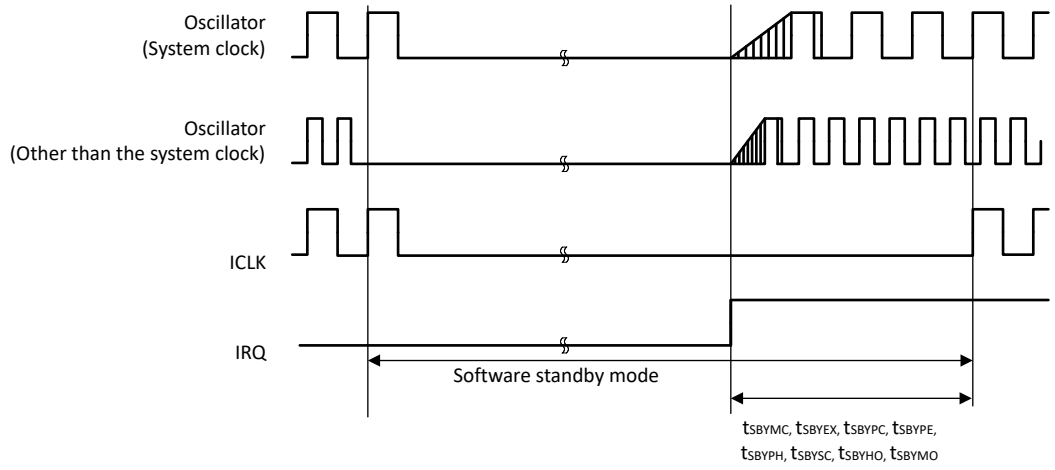
Note 1. When the frequency of the crystal is 48 MHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x05) and the greatest value of the internal clock division setting is 1.
Note 2. When the frequency of PLL1P is 480 MHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x05) and the greatest value of the internal clock division setting is 8.
Note 3. When the frequency of the external clock is 48 MHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x00) and the greatest value of the internal clock division setting is 1.
Note 4. When the frequency of PLL1P is 480 MHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x00) and the greatest value of the internal clock division setting is 8.
Note 5. The Sub-clock oscillator frequency is 32.768 kHz and the greatest value of the internal clock division setting is 1.
Note 6. The HOCO frequency is 20 MHz and the greatest value of the internal clock division setting is 1.
Note 7. The PLL frequency is 480 MHz and the greatest value of the internal clock division setting is 8.
Note 8. The MOCO frequency is 8 MHz and the greatest value of the internal clock division setting is 1.
Note 9. For details, see SSCR1.SS1FR, DPSBYCR.DCSSMODE bits in the section 10, Low Power Modes.
Note 10. The recovery time can be calculated with the equation of t_{Common} + max(t_{OSCSTB}, t_{PG1}, t_{PGCK}) + max(t_{PG2}, t_{LPW}). And they can be determined with the following values and equations. For n, the greatest value is selected from among the internal clock(CPUCLK, ICLK, PCLKm, FCLK, BCLK and EBCLK) division settings (m = A to E).
t_{OSCSTB} in the table below means the time when each oscillator is active. When multiple oscillators are active, t_{OSCSTB} is determined by the longest t_{OSCSTB} among the active oscillators.
Note 11. The ICLK frequency is 240 MHz. This recovery time corresponds to t_{PG2}.

Table 60.35 Each element of recovery time

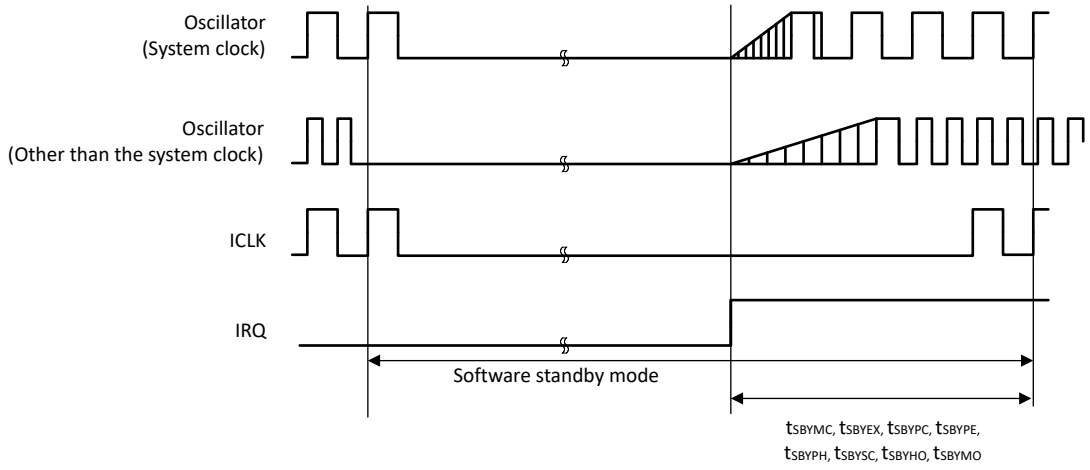
Wakeup time	Oscillation keep	Fast return function	Typ						Max						Unit
			t_{Common}	t_{OSCSTB} *1	t_{PG1}	t_{PGCK}	t_{PG2}	t_{LPW}	t_{Common}	t_{OSCSTB} *1	t_{PG1}	t_{PGCK}	t_{PG2}	t_{LPW}	
t_{SBYMC}	MOSC disabled	Enabled	52.667 + 4/ fICLK	$t_{MAINOSCWT}$	75.5	2.1 + 10.5/ fMOCO + 2.5n/ fMOCO + 2.5/ fSRCCLK + 2/ fICLK	1449/ fMOCO + 10/ fICLK	10 + 2/ fICLK + 2n/ fMAIN	82.369 + 4/ fICLK	$t_{MAINOSCWT}$ + 11/0.236	88.8	2.5 + 10.5/ fMOCO + 2.5n/ fMOCO + 2.5/ fSRCCLK + 2/ fICLK	1449/ fMOCO+ 10/ fICLK	10 + 2/ fICLK + 2n/ fMAIN	µs
	MOSC enabled	Enabled	52.667 + 4/ fICLK	3/0.262				10 + 2/ fICLK + 2n/ fMAIN	82.369 + 4/ fICLK	14/0.236				10 + 2/ fICLK + 2n/ fMAIN	µs
t_{SBYPC}	MOSC disabled	Enabled	52.667 + 4/ fICLK	24.125 +$t_{MAINOSCWT}$ + 31/0.262 *2				10 + 2/ fICLK + 2n/ fPLL	82.369 + 4/ fICLK	24.05 + $t_{MAINOSCWT}$ + 42/0.236 *3				10 + 2/ fICLK + 2n/ fPLL	µs
	MOSC enabled	Enabled	52.667 + 4/ fICLK	24.125 + 34/0.262 *2				10 + 2/ fICLK + 2n/ fPLL	82.369 + 4/ fICLK	24.05 + 45/0.236 *3				10 + 2/ fICLK + 2n/ fPLL	µs
t_{SBYEX}	—	Enabled	52.667 + 4/ fICLK	3/0.262				10 + 2/ fICLK + 2n/ fEXMAIN	82.369 + 4/ fICLK	14/0.236				10 + 2/ fICLK + 2n/ fEXMAIN	µs
t_{SBYPE}	—	Enabled	52.667 + 4/ fICLK	24.125 + 34/0.262 *2				10 + 2/ fICLK + 2n/ fPLL	82.369 + 4/ fICLK	24.05 +45/0.236*3				10 + 2/ fICLK + 2n/ fPLL	µs
t_{SBYSC}	—	Enabled	52.667 + 4/ fICLK	0				10 + 2/ fICLK + 2n/ fSOSC	82.369 + 4/ fICLK	0				10 + 2/ fICLK + 2n/ fSOSC	µs
t_{SBYHO}	—	Enabled	52.667 + 4/ fICLK	23.375				10 + 2/ fICLK + 2n/ fHOCO	82.369 + 4/ fICLK	70.234				10 + 2/ fICLK + 2n/ fHOCO	µs
t_{SBYPH}	—	Enabled	52.667 + 4/ fICLK	24.125 + 140*2				10 + 2/ fICLK + 2n/ fPLL	82.369 + 4/ fICLK	24.05 +202*3				10 + 2/ fICLK + 2n/ fPLL	µs
t_{SBYMO}	—	Enabled	52.667 + 4/ fICLK	0				10 + 2/ fICLK + 2n/ fMOCO	82.369 + 4/ fICLK	0				10 + 2/ fICLK + 2n/ fMOCO	µs

Note: The unit of frequency is MHz.
Note 1. If more than one oscillator is operating, the largest value of the operating oscillator in this column is applied.
Note 2. "24.125" can be reduced when both PLL1LDOCR.SKEEP and PLL2LDOCR.SKEEP are 1.

Note 3. "24.05" can be reduced when both PLL1LDOCR.SKEEP and PLL2LDOCR.SKEEP are 1.



When stabilization of the system clock oscillator is slower



When stabilization of an oscillator other than the system clock is slower

Figure 60.24 Software Standby mode cancellation timing

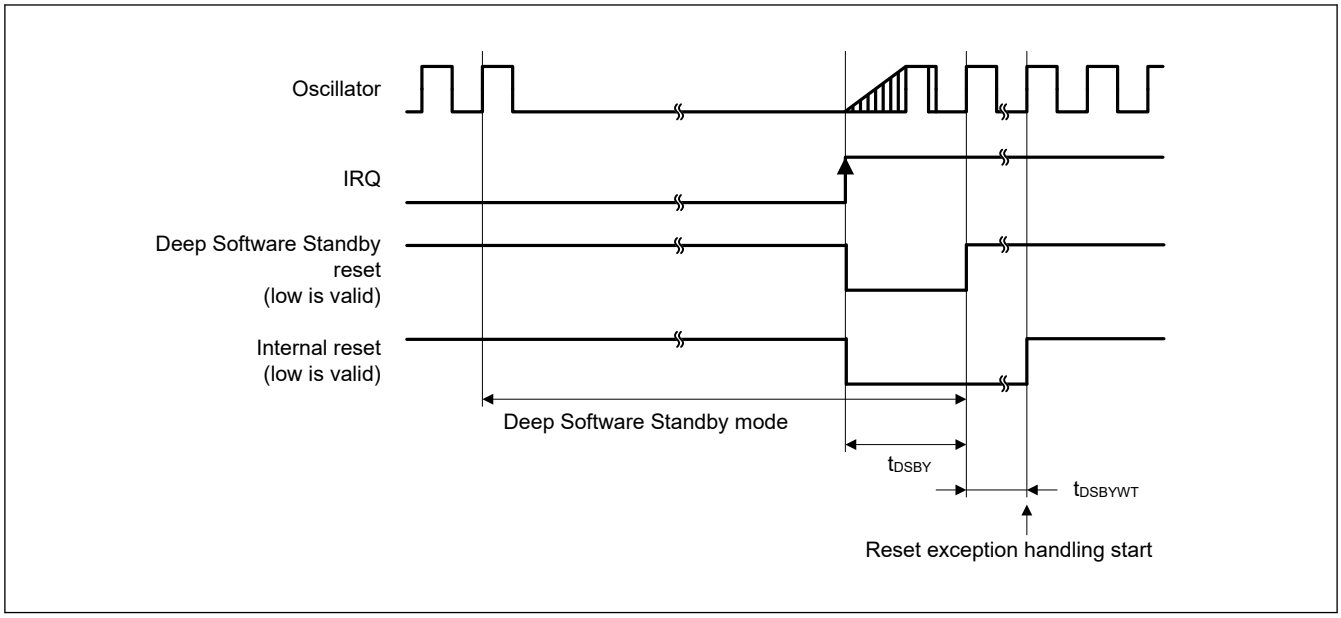


Figure 60.25 Deep Software Standby mode cancellation timing

60.3.5 NMI and IRQ Noise Filter

Table 60.36 NMI and IRQ noise filter

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions	
NMI pulse width	t_{NMIW}	200	—	—	ns	NMI digital filter disabled	$t_{Pcyc} \times 2 \leq 200$ ns
		$t_{Pcyc} \times 2$*1	—	—			$t_{Pcyc} \times 2 > 200$ ns
		200	—	—		NMI digital filter enabled	$t_{NMICK} \times 3 \leq 200$ ns
		$t_{NMICK} \times 3.5$*2	—	—			$t_{NMICK} \times 3 > 200$ ns
IRQ pulse width	t_{IRQW}	200	—	—	ns	IRQ digital filter disabled	$t_{Pcyc} \times 2 \leq 200$ ns
		$t_{Pcyc} \times 2$*1	—	—			$t_{Pcyc} \times 2 > 200$ ns
		200	—	—		IRQ digital filter enabled	$t_{IRQCK} \times 3 \leq 200$ ns
		$t_{IRQCK} \times 3.5$*3	—	—			$t_{IRQCK} \times 3 > 200$ ns

Note: 200 ns minimum in Software Standby mode.
Note: If the system clock source is switched, add 4 clock cycles of the switched source.
Note 1. t_{Pcyc} indicates the PCLKB cycle.
Note 2. t_{NMICK} indicates the cycle of the NMI digital filter sampling clock.
Note 3. t_{IRQCK} indicates the cycle of the IRQi digital filter sampling clock.

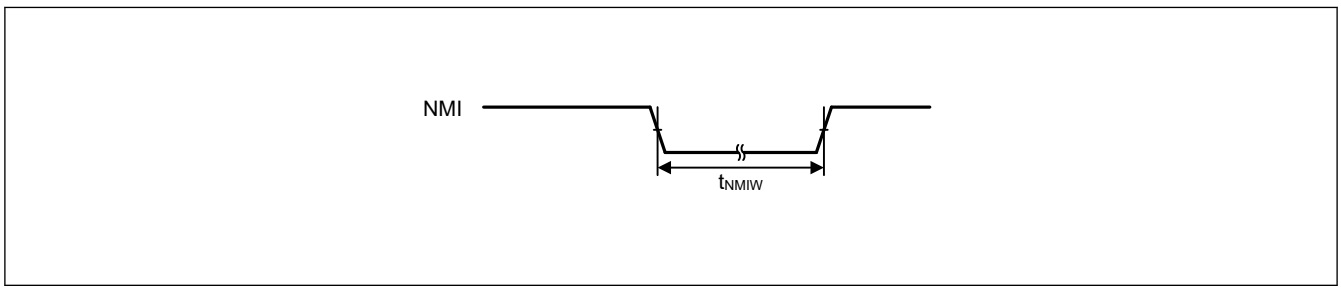


Figure 60.26 NMI interrupt input timing

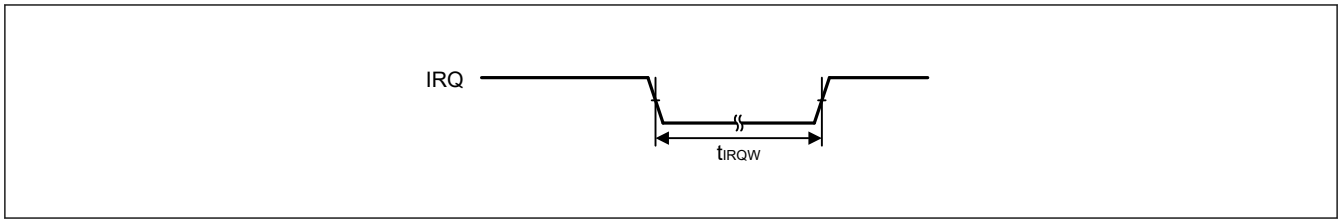


Figure 60.27 IRQ interrupt input timing

60.3.6 Bus Timing

Table 60.37 Bus timing (1 of 2)

Condition 1: When using the CS area controller (CSC).
VCC = VCC_DCDC = VCC_USB = VBATT = 1.68 V to 3.6 V, VCC2 = 1.65 V to 3.6 V
BCLK = 8 to 120 MHz, EBCLK = 8 to 60 MHz (When VCC = VCC_USB = VBATT = 2.70 to 3.6 V)
BCLK = 8 to 60 MHz, EBCLK = 8 to 30 MHz (When VCC = VCC_USB = VBATT = 1.68 to 3.6 V)
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 30 pF
EBCLK: High drive output is selected in the port drive capability bit in the PmnPFS register.
Others: Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Condition 2: When using the SDRAM area controller (SDRAMC).
BCLK = SDCLK = 8 to 120 MHz
VCC = VCC2 = VCC_DCDC = VCC_USB = VBATT = 3.0 to 3.6 V
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 15 pF
SDCLK: High-speed high drive output is selected in the port drive capability bit in the PmnPFS register.
Others: High drive output is selected in the port drive capability bit in the PmnPFS register.

Condition 3: When using the SDRAM area controller (SDRAMC) and CS area controller (CSC) simultaneously.
BCLK = SDCLK = 8 to 60 MHz
VCC = VCC2 = VCC_DCDC = VCC_USB = VBATT = 3.0 to 3.6 V
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 15 pF
EBCLK/SDCLK: High drive output is selected in the port drive capability bit in the PmnPFS register.
Others: Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Parameter		Symbol	Min	Max	Unit	Test conditions
Address delay	2.70V or above	t_{AD}	1.0	12.5	ns	Figure 60.28 to Figure 60.34
	1.68V or above		1.0	12.5	ns	
Byte control delay	2.70V or above	t_{BCD}	1.0	12.5	ns	
	1.68V or above		1.0	12.5	ns	
CS delay	2.70V or above	t_{CSD}	1.0	12.5	ns	
	1.68V or above		1.0	12.5	ns	
ALE delay time	2.70V or above	t_{ALED}	1.0	12.5	ns	
	1.68V or above		1.0	12.5	ns	
RD delay	2.70V or above	t_{RSD}	1.0	12.5	ns	
	1.68V or above		1.0	12.5	ns	
Read data setup time	2.70V or above	t_{RDS}	12.5	—	ns	
	1.68V or above		20.5	—	ns	
Read data hold time	2.70V or above	t_{RDH}	0	—	ns	
	1.68V or above		0	—	ns	
WR/WRn delay	2.70V or above	t_{WRD}	1.0	12.5	ns	
	1.68V or above		1.0	12.5	ns	
Write data delay	2.70V or above	t_{WDD}	—	12.5	ns	
	1.68V or above		—	12.5	ns	
Write data hold time	2.70V or above	t_{WDH}	1.0	—	ns	
	1.68V or above		1.0	—	ns	
WAIT setup time	2.70V or above	t_{WTS}	12.5	—	ns	
	1.68V or above		20.5	—	ns	
WAIT hold time	2.70V or above	t_{WTH}	0	—	ns	
	1.68V or above		0	—	ns	

Table 60.37 Bus timing (2 of 2)

Condition 1: When using the CS area controller (CSC).
VCC = VCC_DCDC = VCC_USB = VBATT = 1.68 V to 3.6 V, VCC2 = 1.65 V to 3.6 V
BCLK = 8 to 120 MHz, EBCLK = 8 to 60 MHz (When VCC = VCC_USB = VBATT = 2.70 to 3.6 V)
BCLK = 8 to 60 MHz, EBCLK = 8 to 30 MHz (When VCC = VCC_USB = VBATT = 1.68 to 3.6 V)
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 30 pF
EBCLK: High drive output is selected in the port drive capability bit in the PmnPFS register.
Others: Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Condition 2: When using the SDRAM area controller (SDRAMC).
BCLK = SDCLK = 8 to 120 MHz
VCC = VCC2 = VCC_DCDC = VCC_USB = VBATT = 3.0 to 3.6 V
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 15 pF
SDCLK: High-speed high drive output is selected in the port drive capability bit in the PmnPFS register.
Others: High drive output is selected in the port drive capability bit in the PmnPFS register.

Condition 3: When using the SDRAM area controller (SDRAMC) and CS area controller (CSC) simultaneously.
BCLK = SDCLK = 8 to 60 MHz
VCC = VCC2 = VCC_DCDC = VCC_USB = VBATT = 3.0 to 3.6 V
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 15 pF
EBCLK/SDCLK: High drive output is selected in the port drive capability bit in the PmnPFS register.
Others: Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Parameter		Symbol	Min	Max	Unit	Test conditions
Address delay 2 (SDRAM)	Condition 2	t_{AD2}	0.8	6.8	ns	Figure 60.35 to Figure 60.41
	Condition 3		0.8	10.8		
CS delay 2 (SDRAM)	Condition 2	t_{CSD2}	0.8	6.8	ns	
	Condition 3		0.8	10.8		
DQM delay (SDRAM)	Condition 2	t_{DQMD}	0.8	6.8	ns	
	Condition 3		0.8	10.8		
CKE delay (SDRAM)	Condition 2	t_{CKED}	0.8	6.8	ns	
	Condition 3		0.8	10.8		
Read data setup time 2 (SDRAM)	Condition 2	t_{RDS2}	2.9	—	ns	
	Condition 3		6.9	—		
Read data hold time 2 (SDRAM)	Condition 2	t_{RDH2}	1.5	—	ns	
	Condition 3		1.5	—		
Write data delay 2 (SDRAM)	Condition 2	t_{WDD2}	—	6.8	ns	
	Condition 3		—	10.8		
Write data hold time 2 (SDRAM)	Condition 2	t_{WDH2}	0.8	—	ns	
	Condition 3		0.8	—		
WE delay (SDRAM)	Condition 2	t_{WED}	0.8	6.8	ns	
	Condition 3		0.8	10.8		
RAS delay (SDRAM)	Condition 2	t_{RASD}	0.8	6.8	ns	
	Condition 3		0.8	10.8		
CAS delay (SDRAM)	Condition 2	t_{CASD}	0.8	6.8	ns	
	Condition 3		0.8	10.8		

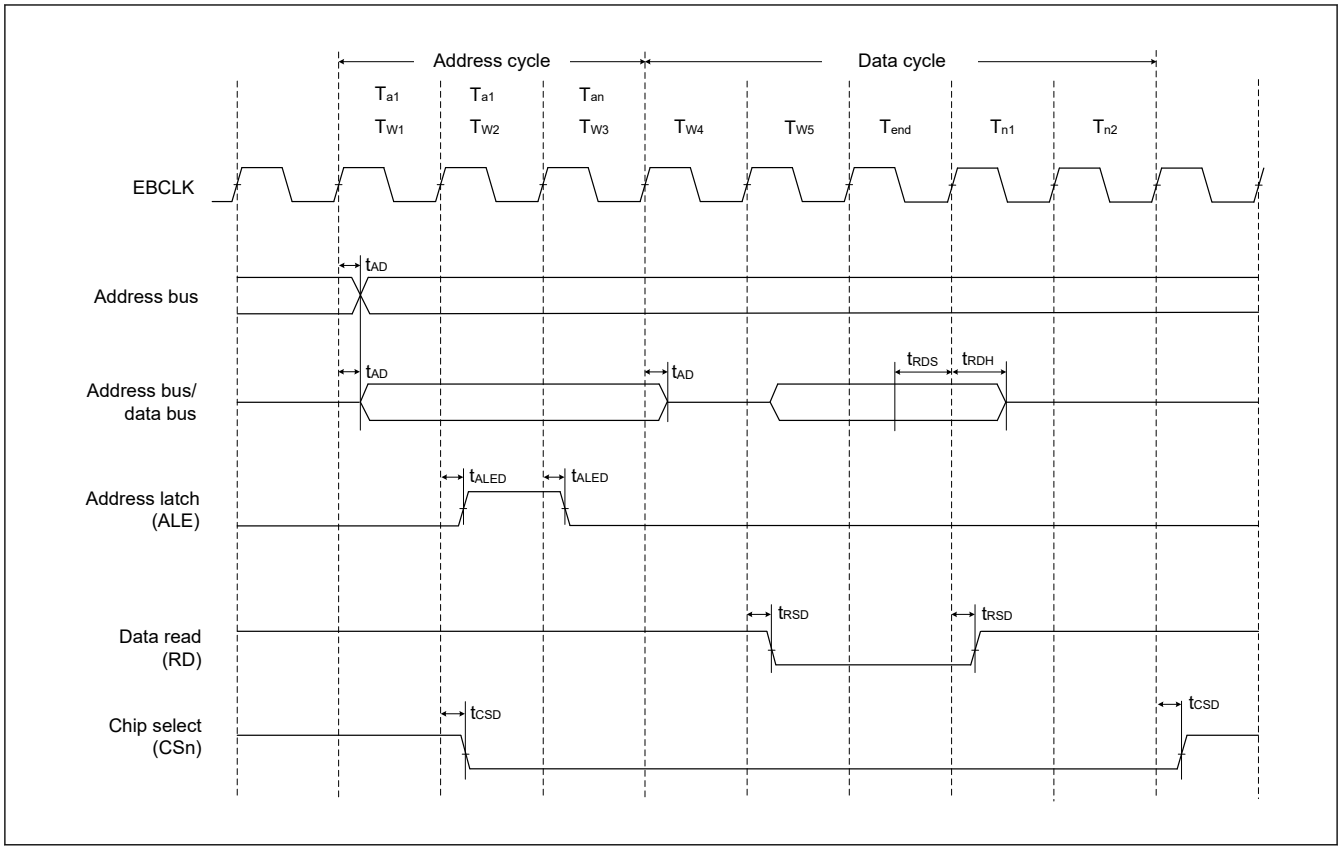


Figure 60.28 Address/data multiplexed bus read access timing

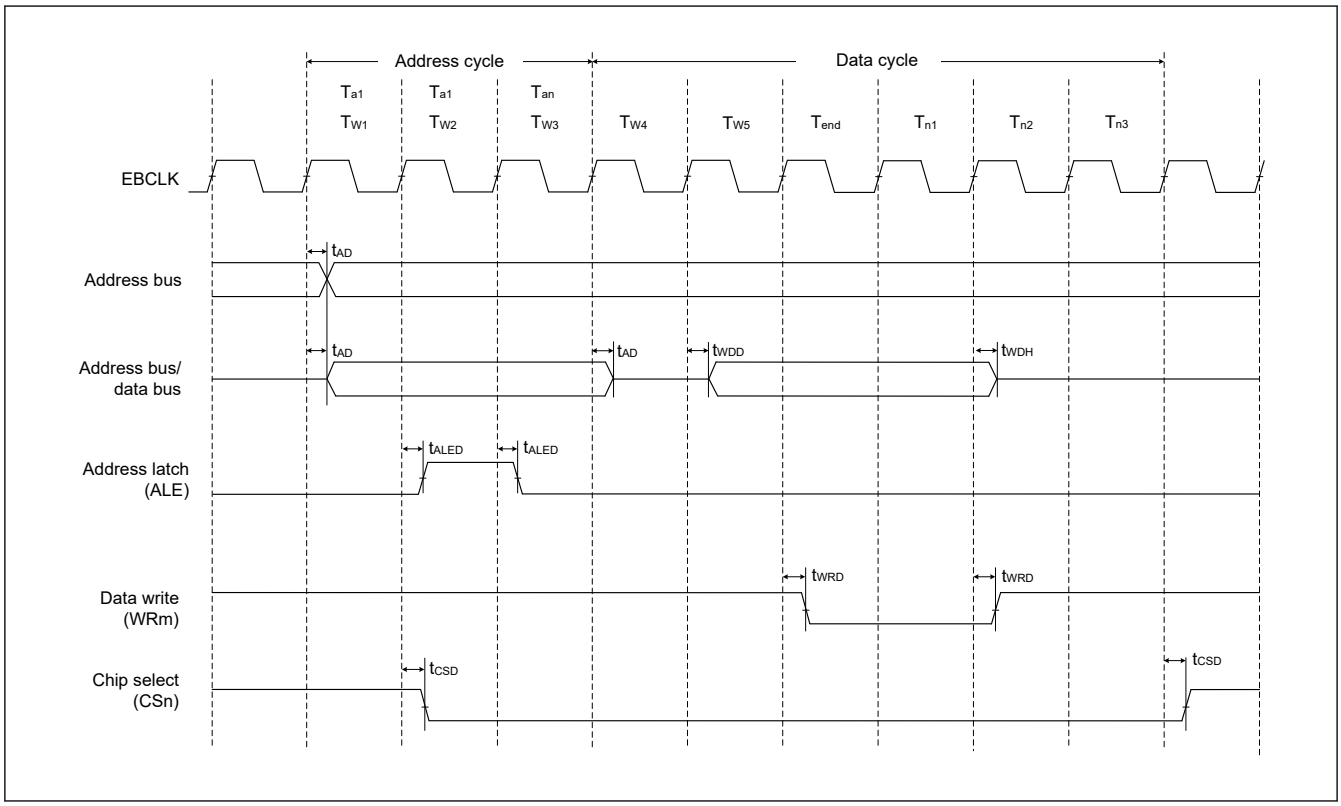


Figure 60.29 Address/data multiplexed bus write access timing

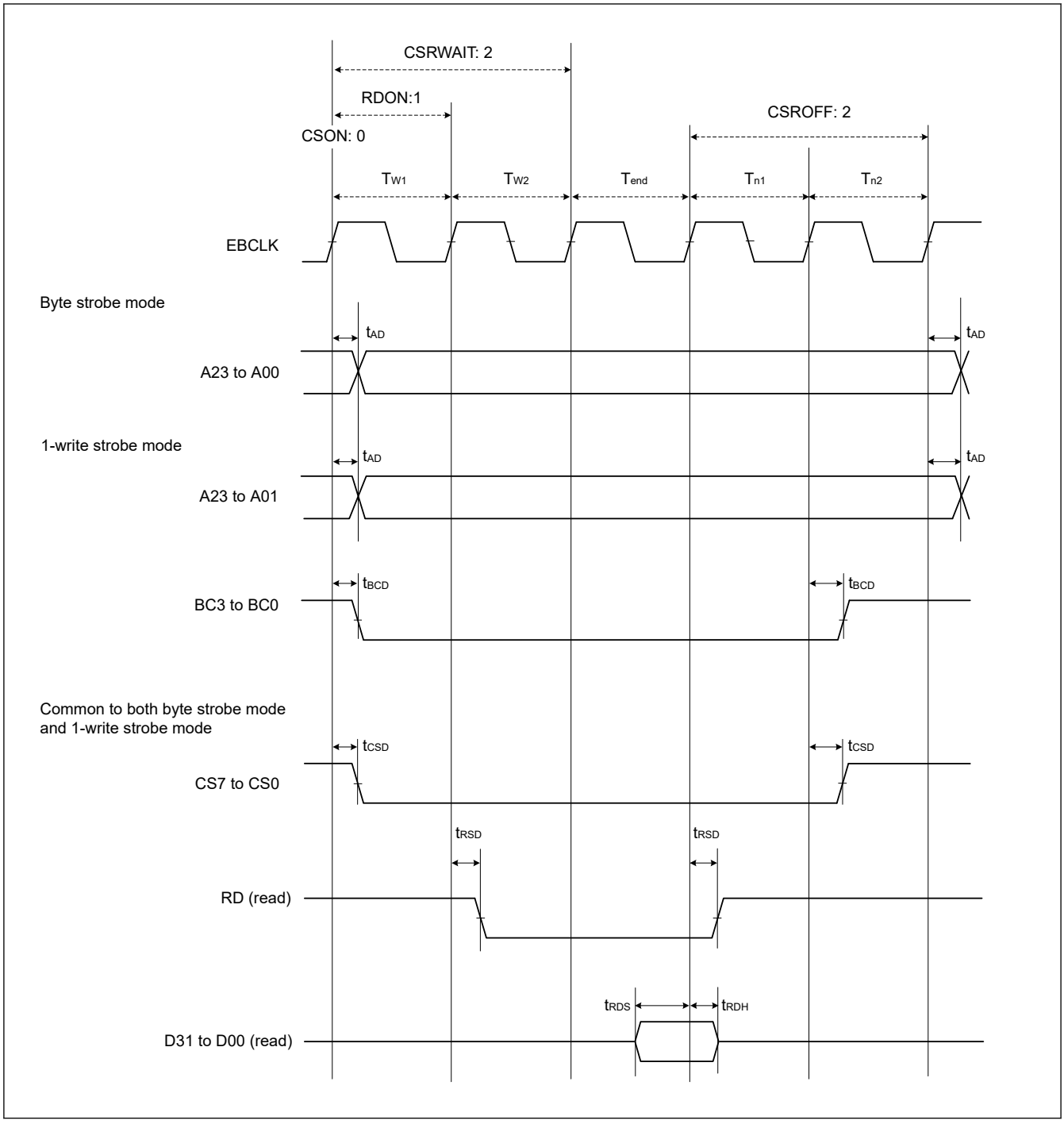
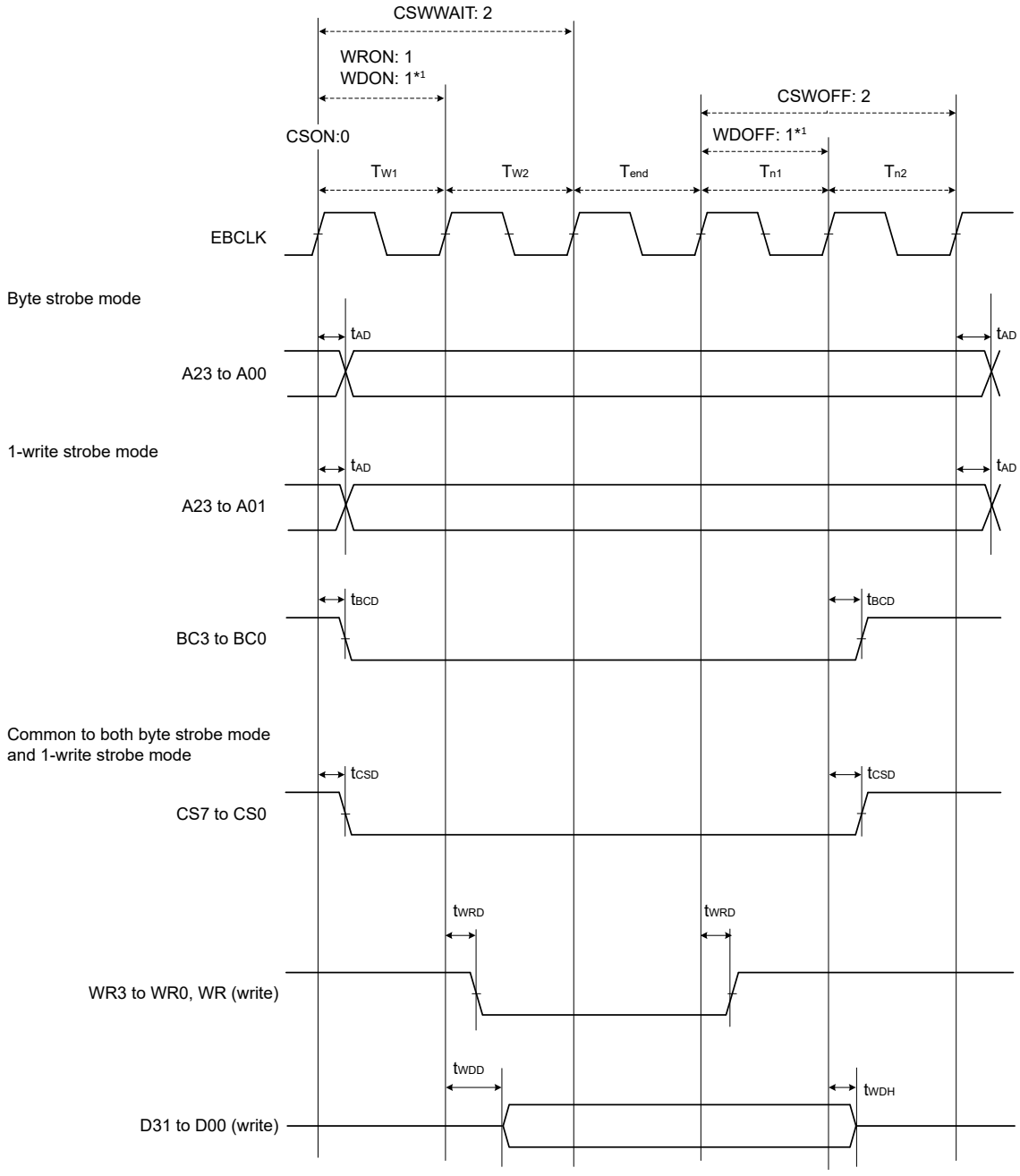


Figure 60.30 External bus timing for normal read cycle with bus clock synchronized



Note 1. Always specify WDON and WDOFF as at least one EBCLK cycle.

Figure 60.31 External bus timing for normal write cycle with bus clock synchronized

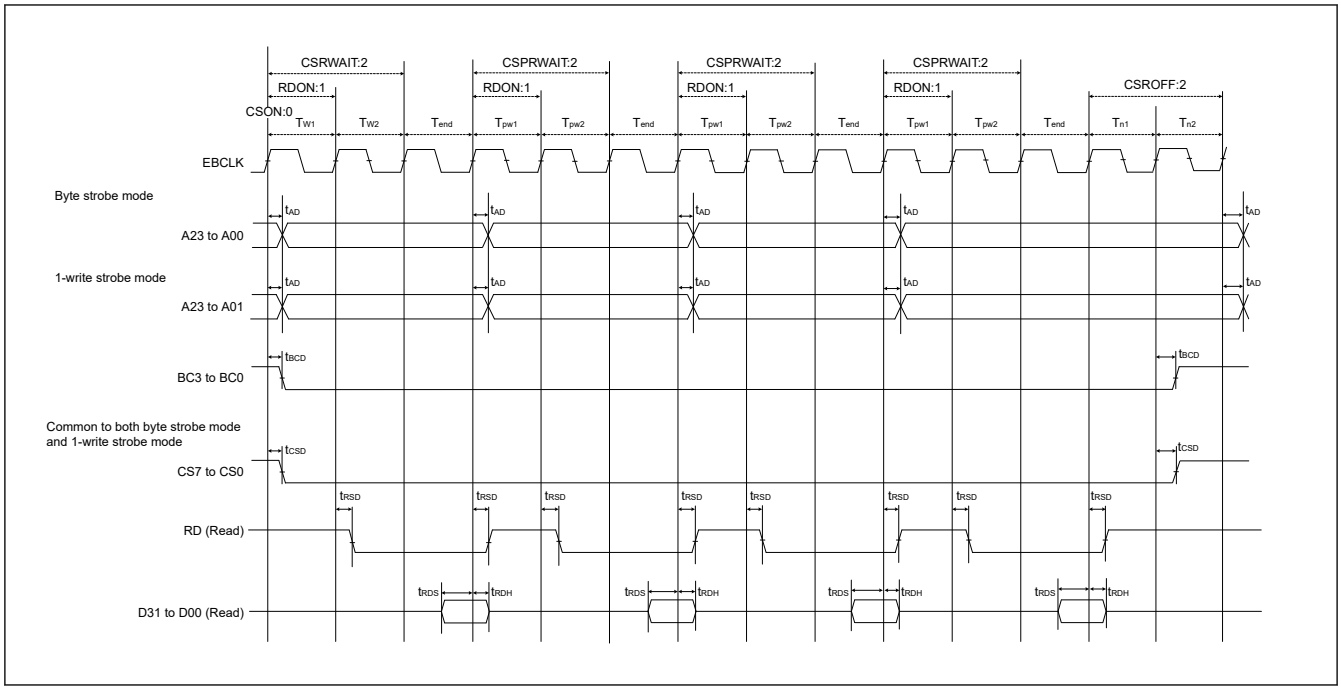
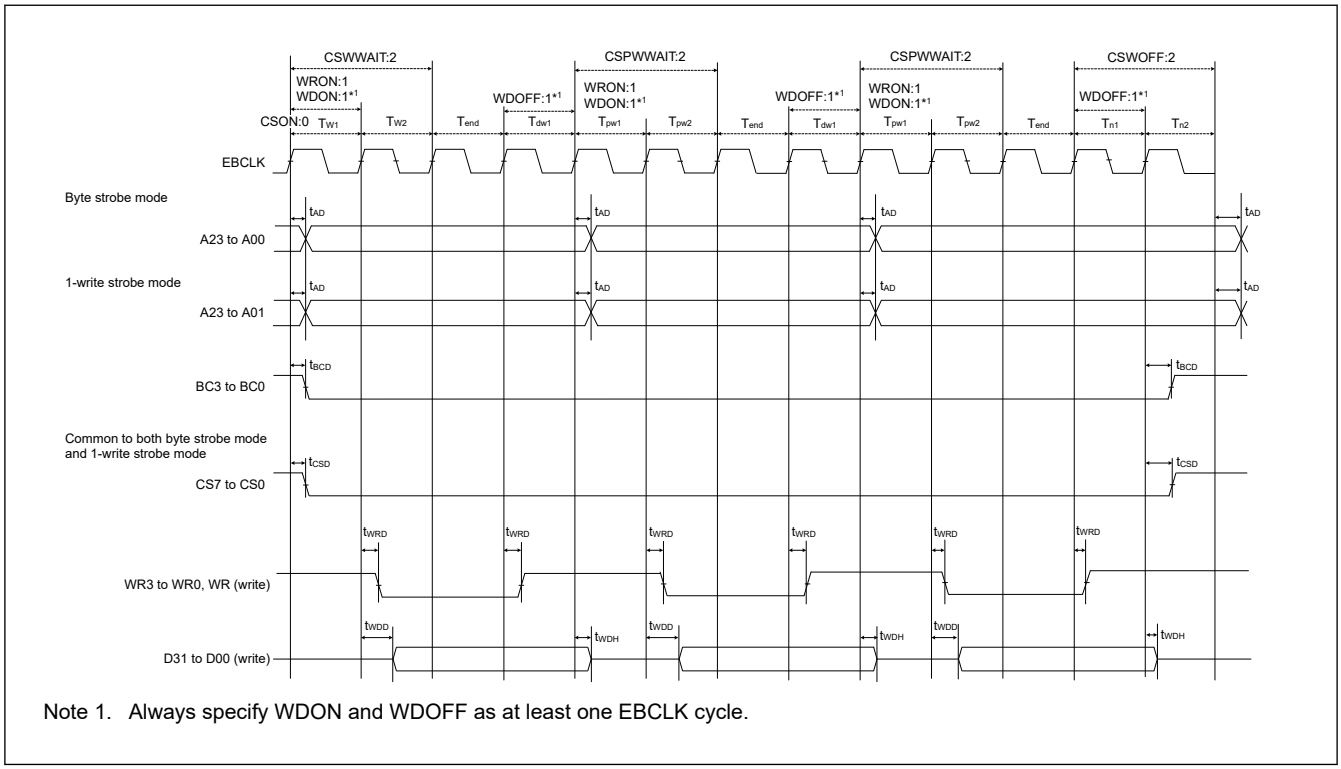


Figure 60.32 External bus timing for page read cycle with bus clock synchronized



Note 1. Always specify WDON and WDOFF as at least one EBCLK cycle.

Figure 60.33 External bus timing for page write cycle with bus clock synchronized

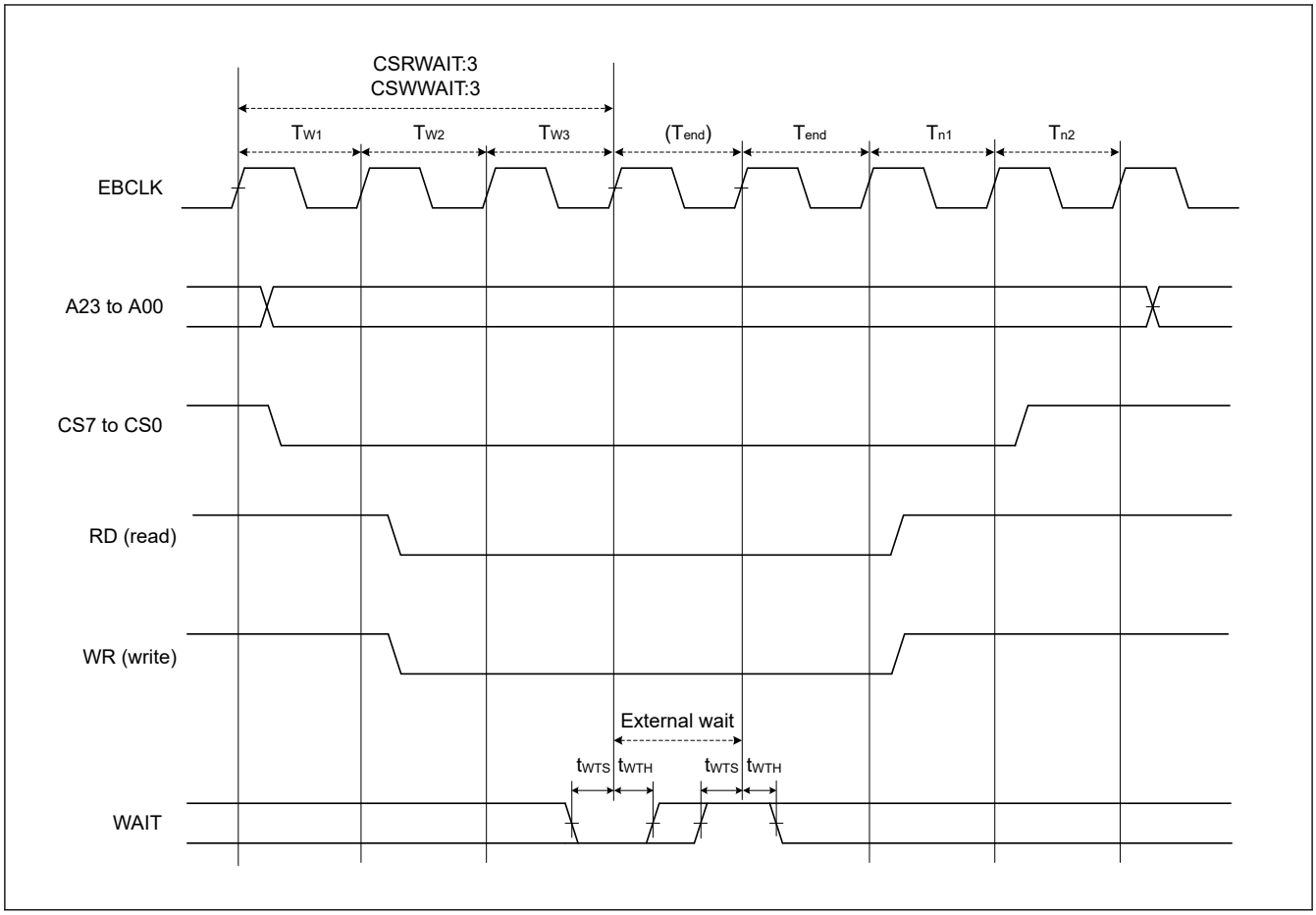
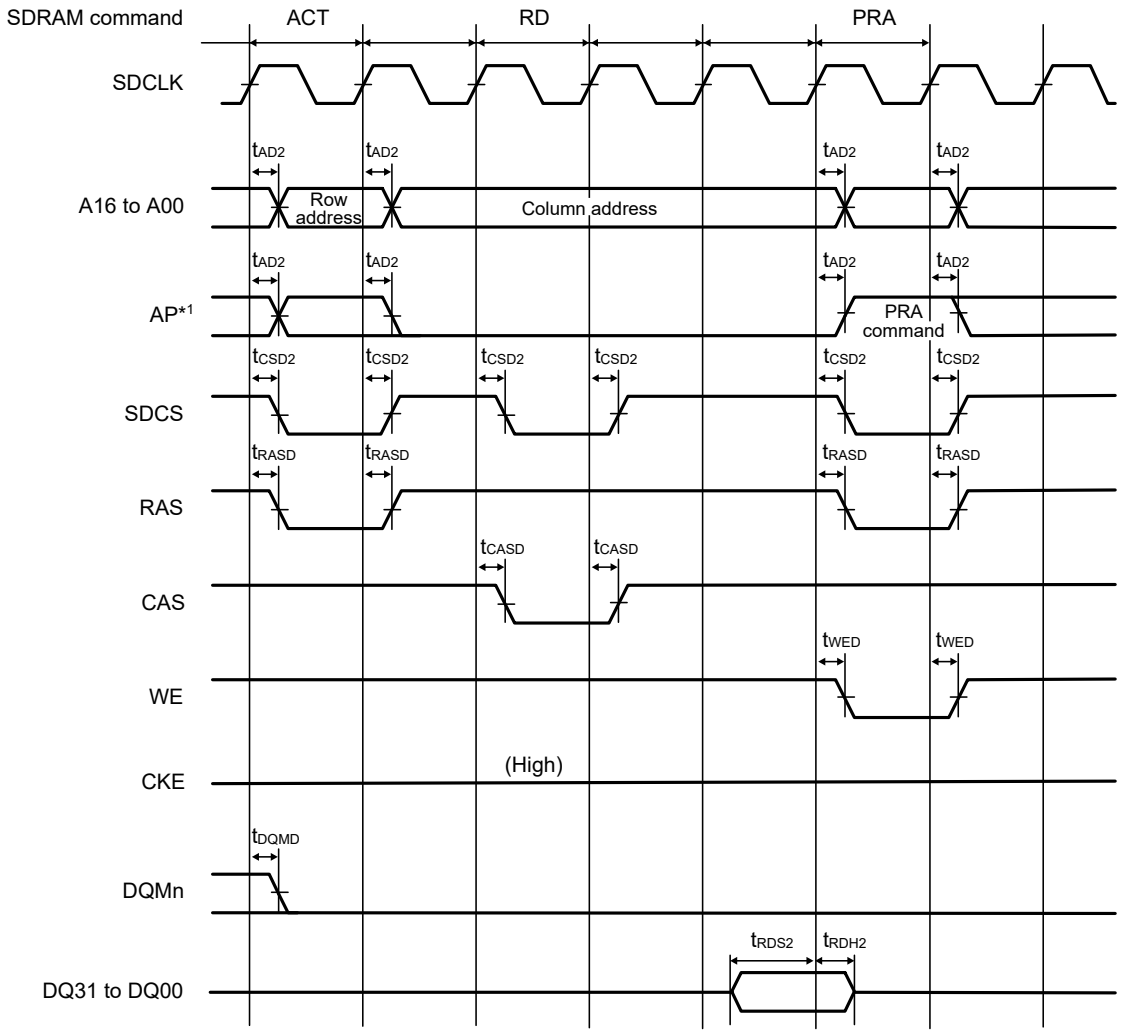
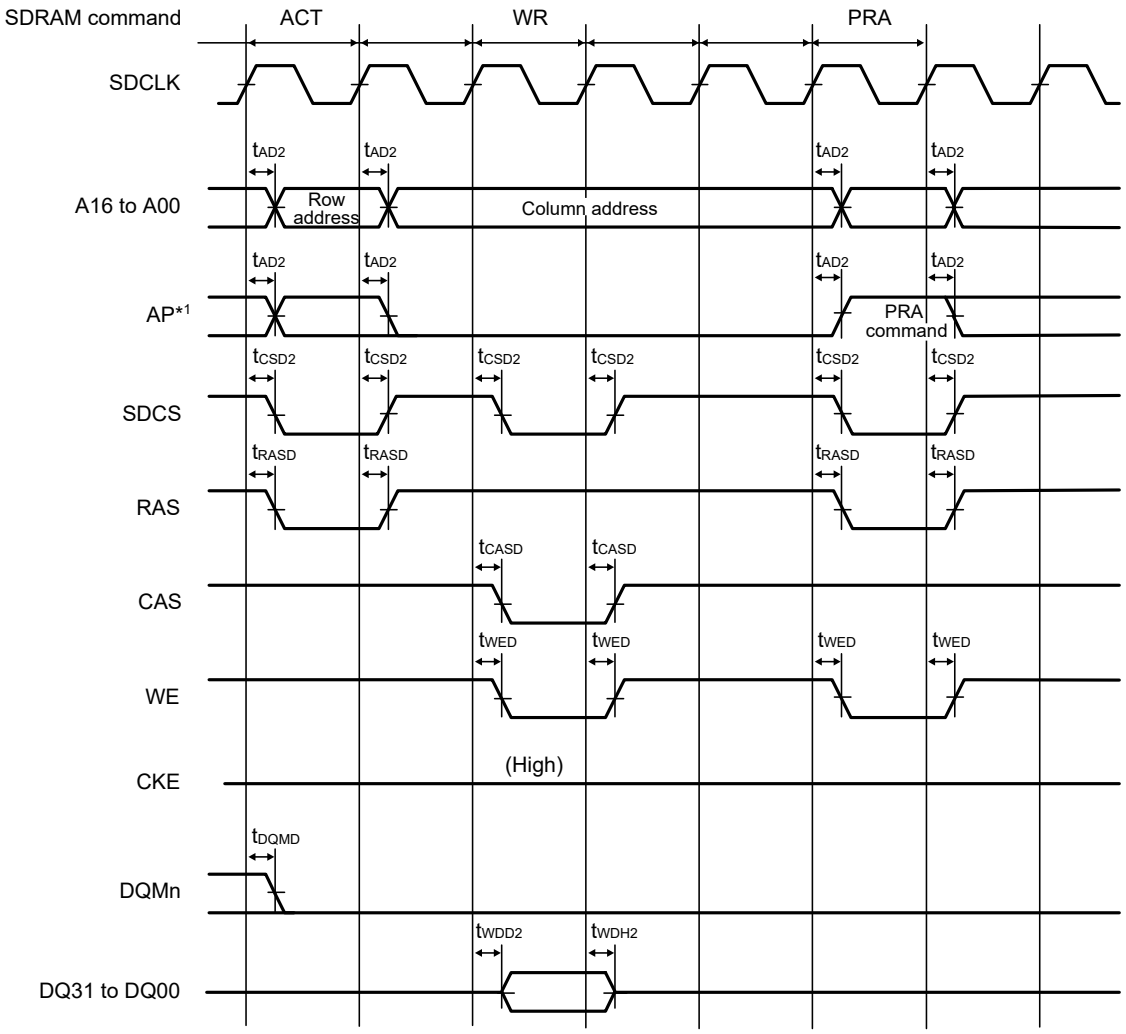


Figure 60.34 External bus timing for external wait control



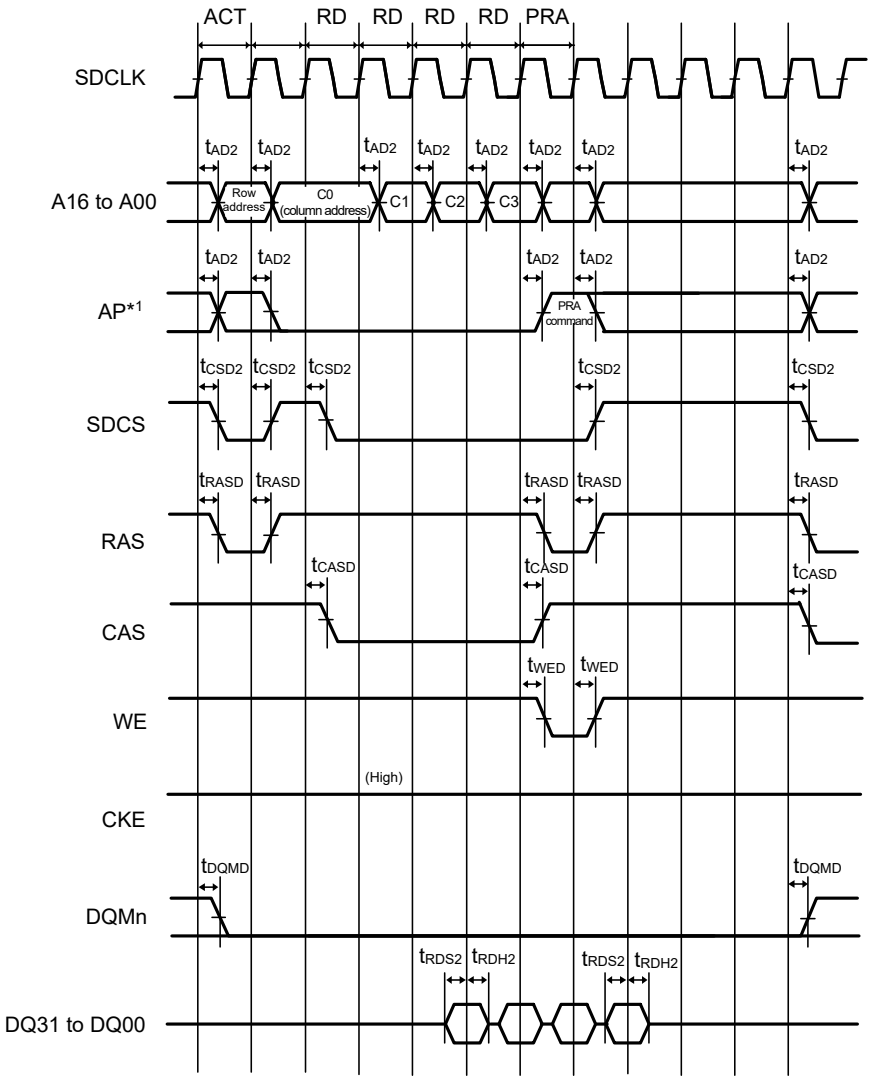
Note 1. Address pins are for output of the precharge-select command (Precharge-sel) for the SDRAM.

Figure 60.35 SDRAM single read timing



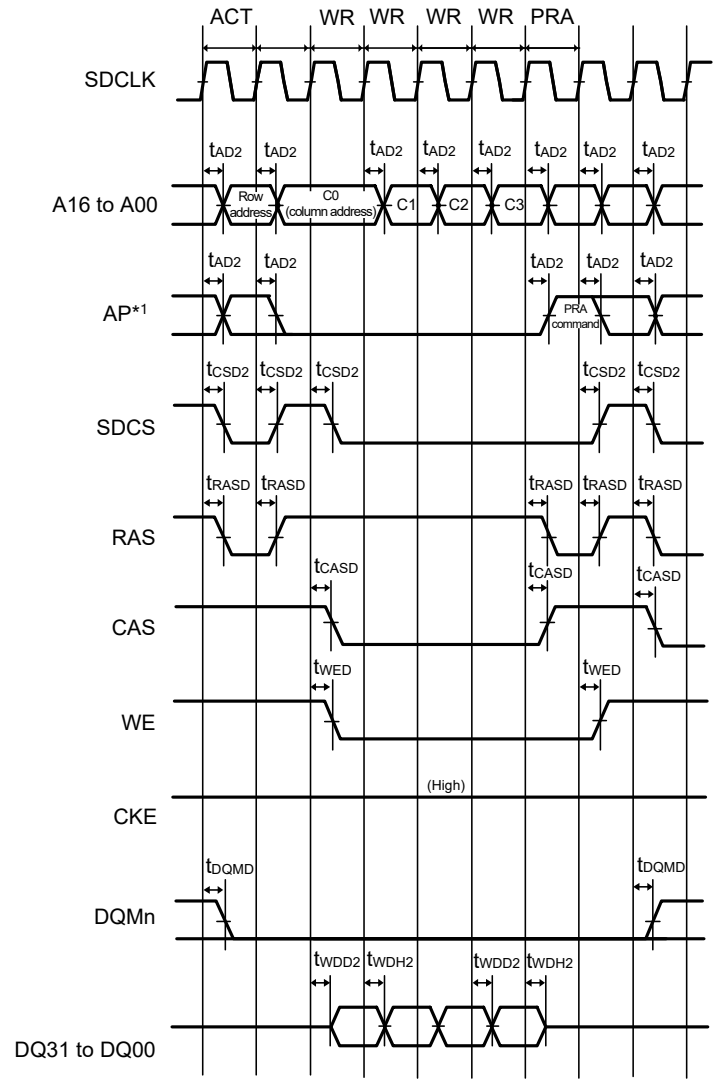
Note 1. Address pins are for output of the precharge-select command (Precharge-sel) for the SDRAM

Figure 60.36 SDRAM single write timing



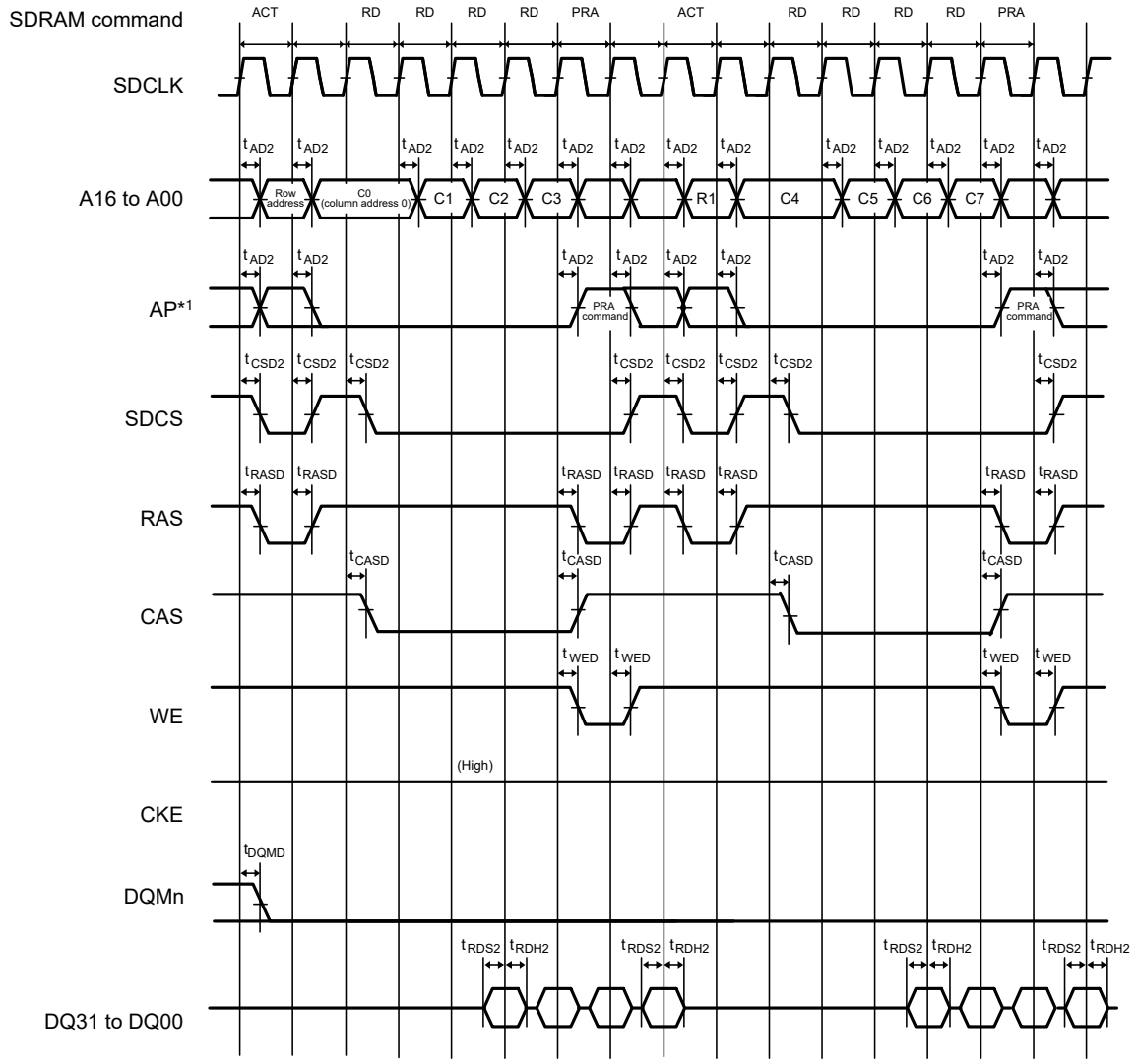
Note 1. Address pins are for output of the precharge-select command (Precharge-sel) for the SDRAM.

Figure 60.37 SDRAM multiple read timing



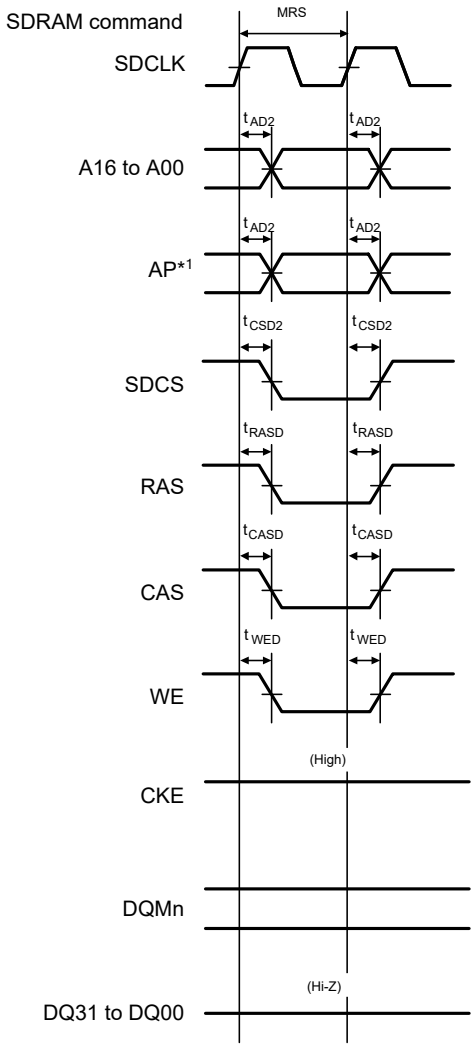
Note 1. Address pins are for output of the precharge-select command (Precharge-sel) for the SDRAM.

Figure 60.38 SDRAM multiple write timing



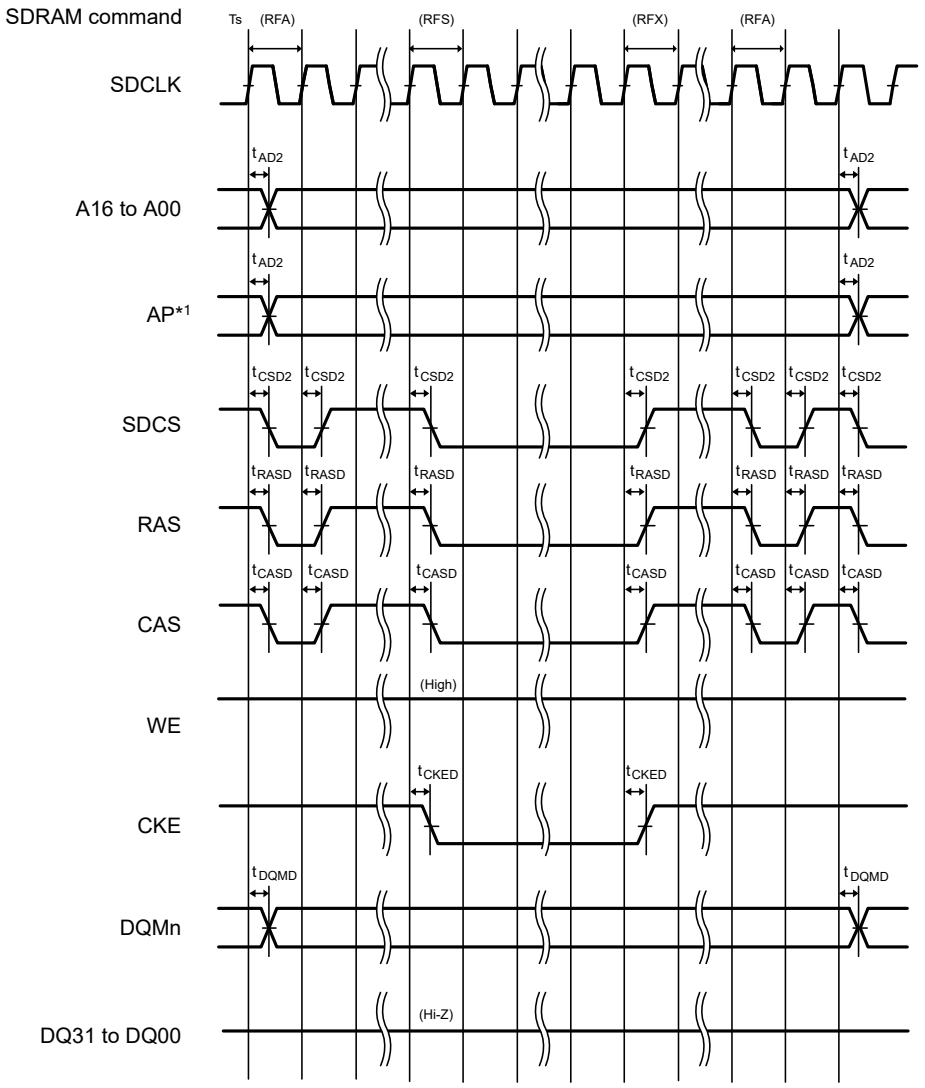
Note 1. Address pins are for output of the precharge-select command (Precharge-sel) for the SDRAM.

Figure 60.39 SDRAM multiple read line stride timing



Note 1. Address pins are for output of the precharge-select command (Precharge-sel) for the SDRAM.

Figure 60.40 SDRAM mode register set timing



Note 1. Address pins are for output of the precharge-select command (Precharge-sel) for the SDRAM.

Figure 60.41 SDRAM self-refresh timing

60.3.7 I/O Ports, POEG, GPT, AGT, ULPT and ADC12 Trigger Timing

Table 60.38 I/O ports, POEG, GPT, AGT, ULPT and ADC12 trigger timing (1 of 3)

GPT Conditions:
Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If GPT pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.
AGT Conditions:
Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	Min	Max	Unit	Test conditions
I/O ports	Input data pulse width	t_{PRW}	5.5	—	t_{Icyc}	Figure 60.42
	EXCIN input frequency	t_{EXCIN}	—	36	kHz	
	RTCICn (n = 0 to 2) input pulse width	t_{RTCICW}	13.89	—	µs	Figure 60.43
POEG	POEG input trigger pulse width	t_{POEW}	3	—	t_{Pcyc}	Figure 60.44

Table 60.38 I/O ports, POEG, GPT, AGT, ULPT and ADC12 trigger timing (2 of 3)

GPT Conditions:
Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If GPT pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.
AGT Conditions:
Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter			Symbol	Min	Max	Unit	Test conditions
GPT	Input capture pulse width (Cycle)	Single edge	t_{GTICW}*1	1.5	—	t_{PDcyc}	Figure 60.45
		Dual edge		2.5	—		
	Input capture pulse width (Time)	2.70V or above	t_{GTICW}*1	12.5	—	ns	
		1.68V or above (VCC) 1.65V or above (VCC2)		25.0	—		
	GTIOCxY output skew (x = 0 to 7, Y = A or B)	2.70V or above	t_{GTISK}	—	4	ns	Figure 60.46
		1.68V or above (VCC) 1.65V or above (VCC2)		—	5		
	GTIOCxY output skew (x = 8 to 13, Y = A or B)	2.70V or above		—	4		
		1.68V or above (VCC) 1.65V or above (VCC2)		—	5		
	GTIOCxY output skew (x = 0 to 13, Y = A or B)	2.70V or above		—	6		
		1.68V or above (VCC) 1.65V or above (VCC2)		—	7		
	OPS output skew GTOUUP, GTOULO, GTOVUP, GTOVLO, GTOWUP, GTOWLO	2.70V or above	t_{GTOSK}	—	5	ns	Figure 60.47
		1.68V or above (VCC) 1.65V or above (VCC2)		—	6		
AGT	AGTIO, AGTEE input cycle	2.70V or above	t_{ACYC}*2	100	—	ns	Figure 60.48
		1.68V or above (VCC) 1.65V or above (VCC2)		100	—		
	AGTIO, AGTEE input high width, low width	2.70V or above	t_{ACKWH}, t_{ACKWL}	40	—	ns	
		1.68V or above (VCC) 1.65V or above (VCC2)		40	—		
	AGTIO, AGTO, AGTOA, AGTOB output cycle	2.70V or above	t_{ACYC2}	62.5	—	ns	
		1.68V or above (VCC) 1.65V or above (VCC2)		62.5	—		

Table 60.38 I/O ports, POEG, GPT, AGT, ULPT and ADC12 trigger timing (3 of 3)

GPT Conditions:
Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If GPT pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.
AGT Conditions:
Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter			Symbol	Min	Max	Unit	Test conditions
ULPT	ULPTEE, ULPTEVI input cycle	2.70V or above	t_{ULCYC}*3	32	-	μs	Figure 60.49
		1.68V or above (VCC) 1.65V or above (VCC2)		32	-		
	ULPTEE, ULPTVI input high width, low width	2.70V or above	t_{ULCKWH}, t_{ULCKWL}	12	-	μs	
		1.68V or above (VCC) 1.65V or above (VCC2)		12	-		
	ULPTO, ULPTOA, ULPTOB output cycle	2.70V or above	t_{ULCYC2}	64	-	μs	
		1.68V or above (VCC) 1.65V or above (VCC2)		64	-		
ADC12	ADC12 trigger input pulse width	2.70V or above	t_{TRGW}	1.5	—	t_{Pcyc}	Figure 60.50
		1.68V or above (VCC) 1.65V or above (VCC2)		3.0	—		

Note: t_{Icyc}: ICLK cycle, t_{Pcyc}: PCLKB cycle, t_{PDcyc}: PCLKD cycle, $t_{ULPTLCLK}$: ULPTLCLK cycle..
Note 1. For Cycle and Time, the longer time characteristics is applied.
Note 2. Constraints on input cycle:
When not switching the source clock: $t_{Pcyc} \times 2 < t_{ACYC}$ should be satisfied.
When switching the source clock: $t_{Pcyc} \times 6 < t_{ACYC}$ should be satisfied.
Note 3. Constraints on input cycle:
ULPTEVI : $t_{Pcyc} \times 2 < t_{ULCYC}$ should be satisfied.
ULPTEE: $t_{ULPTLCLK} \times 2 < t_{ULCYC}$ should be satisfied.

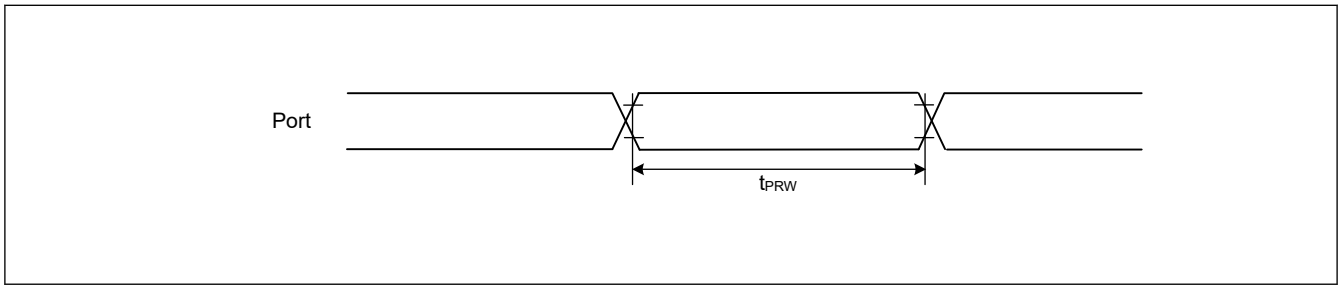


Figure 60.42 I/O ports input timing

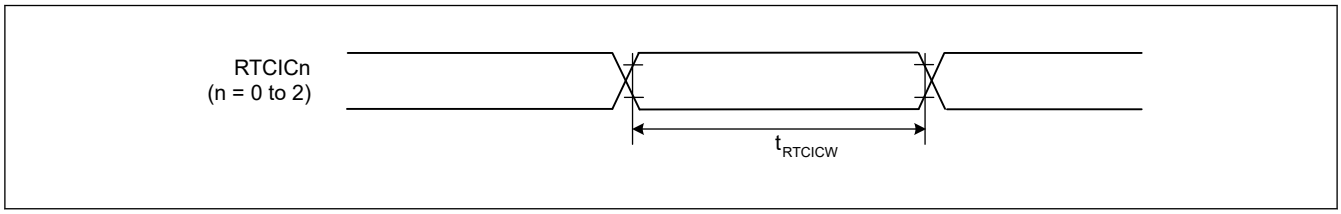


Figure 60.43 RTCICn input timing

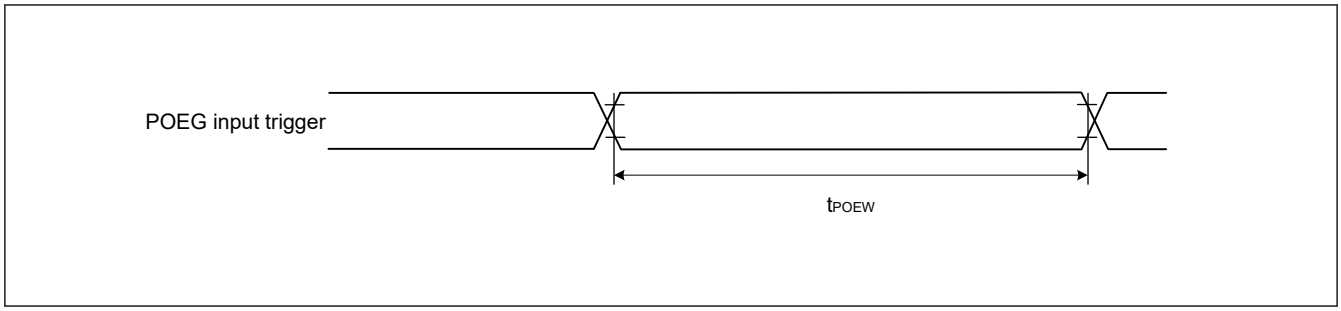


Figure 60.44 POEG input trigger timing

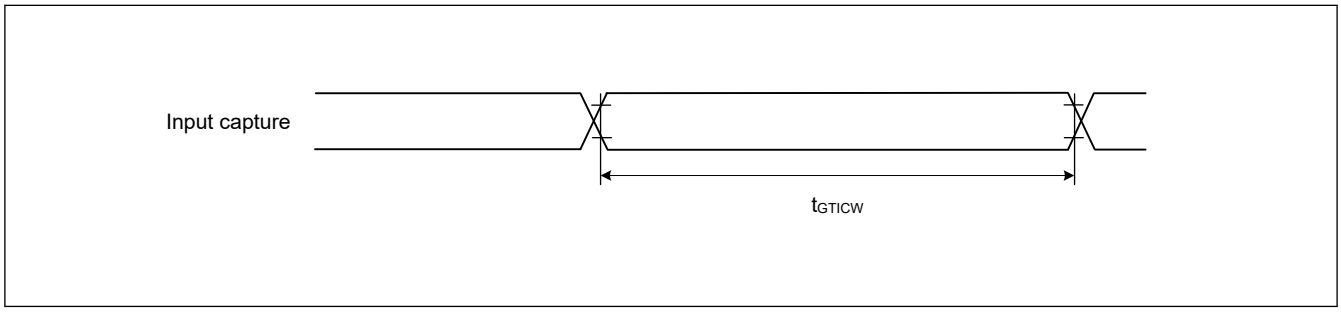


Figure 60.45 GPT input capture timing

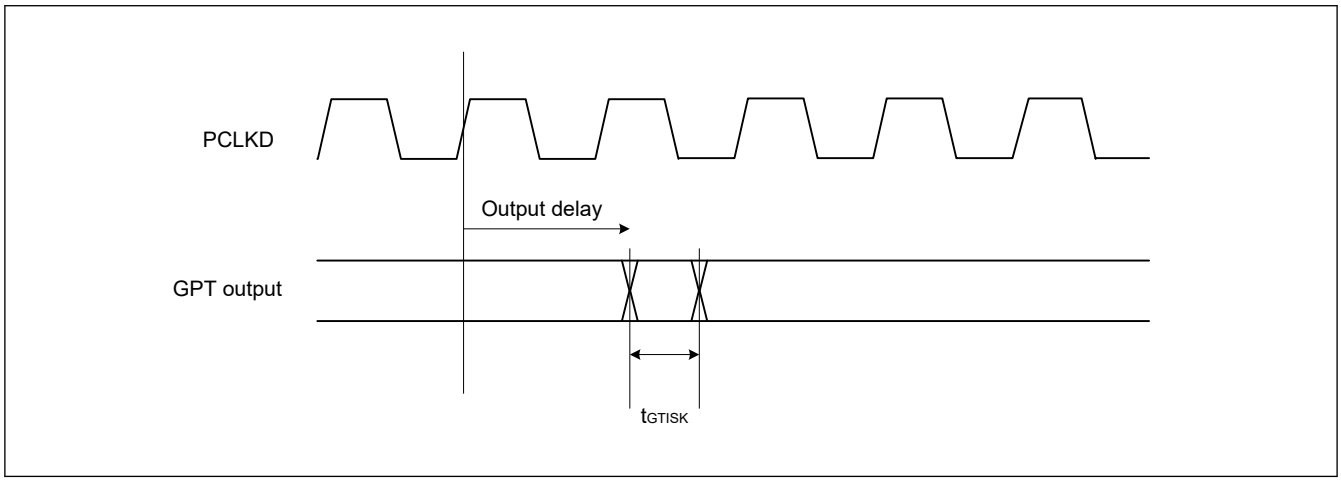


Figure 60.46 GPT output delay skew

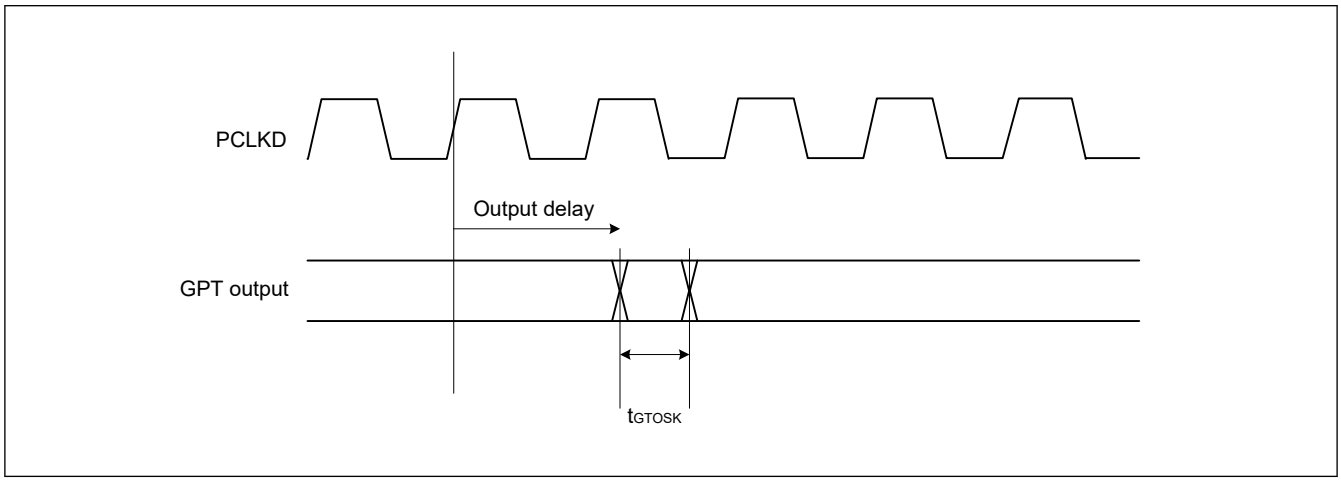


Figure 60.47 GPT output delay skew for OPS

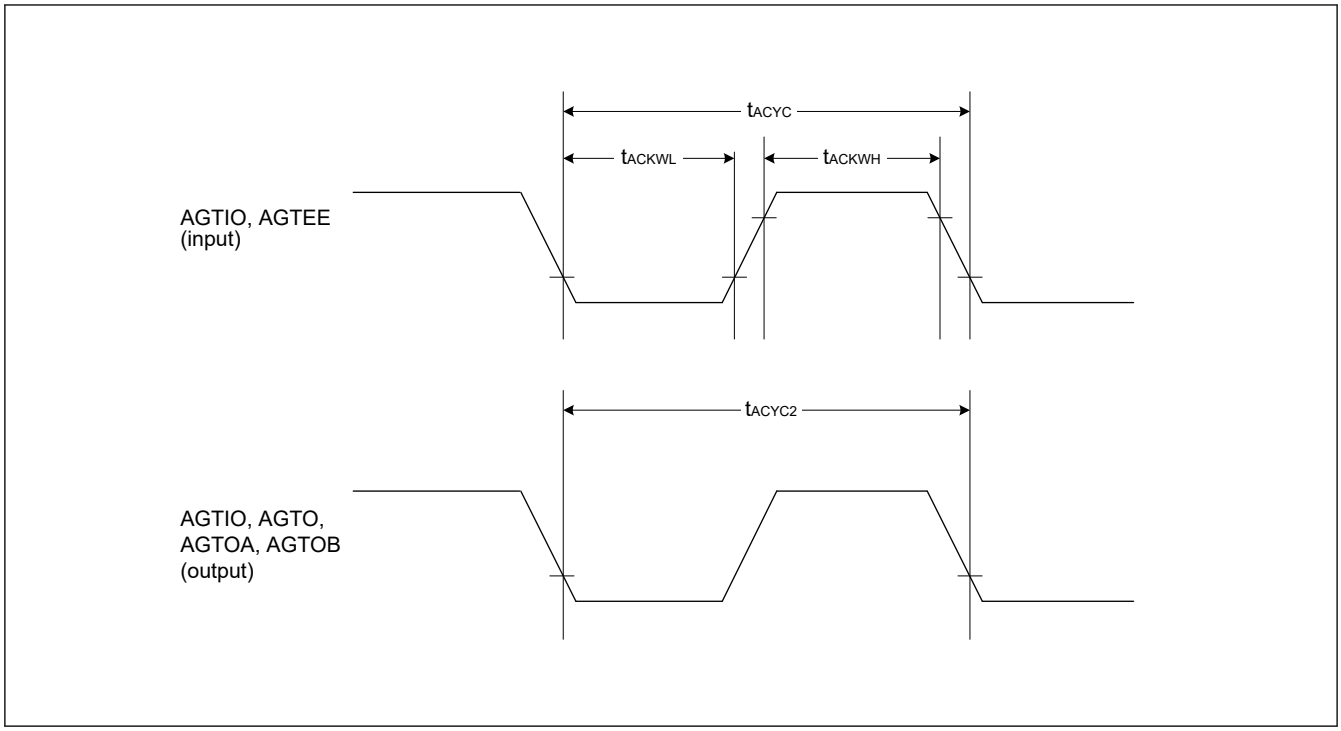


Figure 60.48 AGT input/output timing

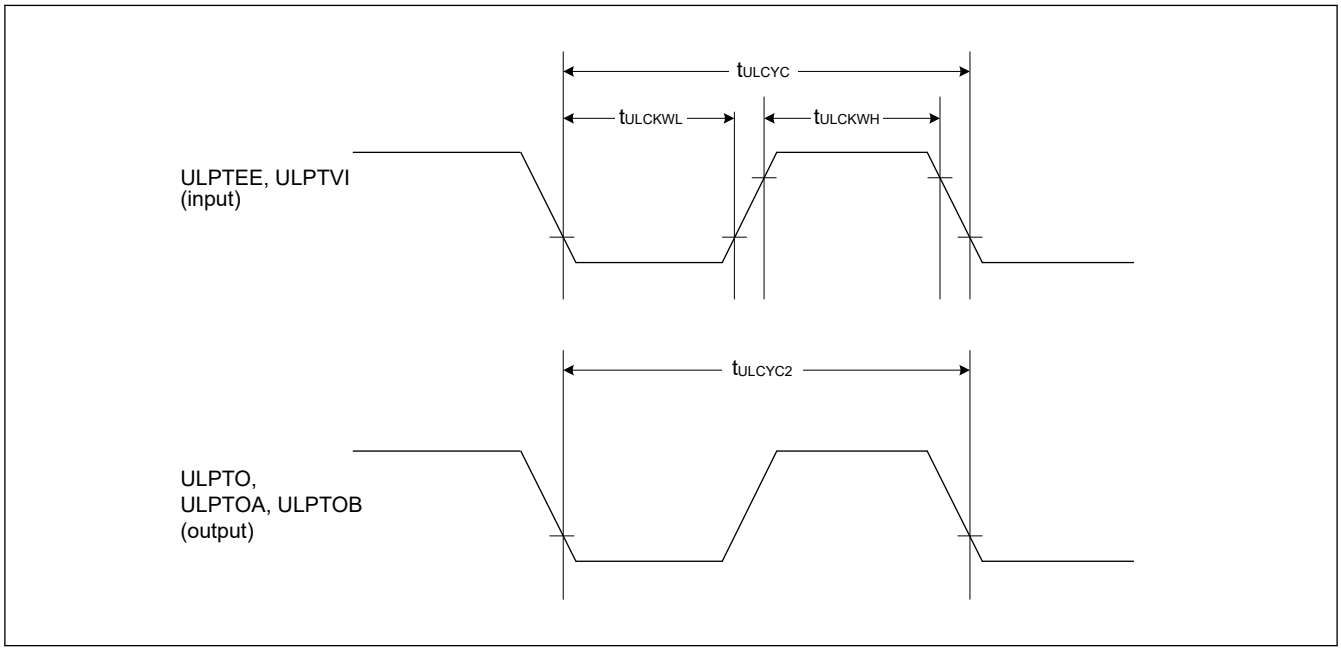


Figure 60.49 ULPT input/output timing

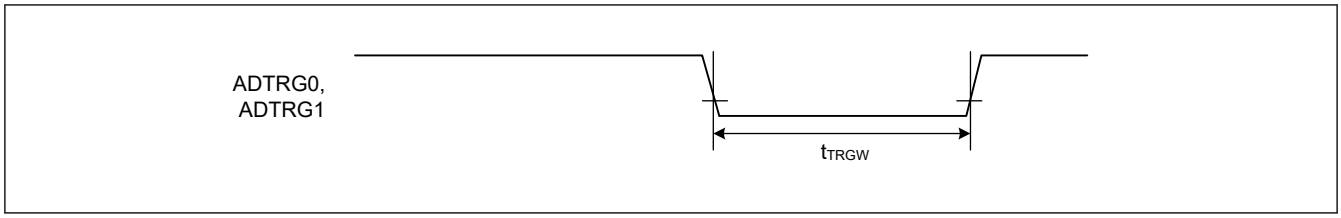


Figure 60.50 ADC12 trigger input timing

60.3.8 CAC Timing

Table 60.39 CAC timing

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
CAC	CACREF input pulse width	$t_{PBcyc} \leq t_{cac}$ *1	t_{CACREF}	$4.5 \times t_{cac} + 3 \times t_{PBcyc}$	—	—	ns	—
		$t_{PBcyc} > t_{cac}$ *1		$5 \times t_{cac} + 6.5 \times t_{PBcyc}$	—	—	ns	

Note: t_{PBcyc}: PCLKB cycle.
Note 1. t_{cac}: CAC count clock source cycle.

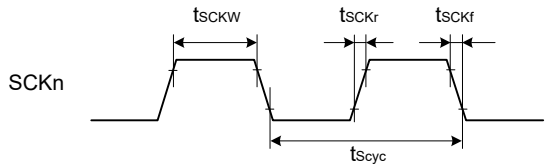
60.3.9 SCI Timing

Table 60.40 SCI timing (Asynchronous mode)

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

Parameter	VCC/VCC2	Symbol	Min	Max	Unit	Note
Input clock cycle	1.68 V or above (VCC) 1.65 V or above (VCC2)	t_{Scyc}	4.0	—	t_{Tcyc}	Figure 60.51
Input clock pulse width	1.68 V or above (VCC) 1.65 V or above (VCC2)	t_{SCKW}	0.4	—	t_{Scyc}	
Input clock rise time	1.68 V or above (VCC) 1.65 V or above (VCC2)	t_{SCKr}	—	0.1*1	t_{Scyc}	
Input clock fall time	1.68 V or above (VCC) 1.65 V or above (VCC2)	t_{SCKf}	—	0.1*1	t_{Scyc}	
Output clock cycle	1.68 V or above (VCC) 1.65 V or above (VCC2)	t_{Scyc}	6.0	—	t_{Tcyc}	
Output clock pulse width	1.68 V or above (VCC) 1.65 V or above (VCC2)	t_{SCKW}	0.4	—	t_{Scyc}	
Output clock rise time	2.70 V or above	t_{SCKr}	—	3.3	ns	
	1.68 V or above (VCC) 1.65 V or above (VCC2)		—	6.6		
Output clock fall time	2.70 V or above	t_{SCKf}	—	3.3	ns	
	1.68 V or above (VCC) 1.65 V or above (VCC2)		—	6.6		

Note: t_{Tcyc}: TCLK cycle.
Note 1. 1 µs at the longest



Note: n = 0 to 4, 9

Figure 60.51 SCK clock input/output timing

Table 60.41 SCI timing (Simple SPI) (1 of 3)

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

Parameter		High Speed/ Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
SCK clock cycle output	Master	—	2.70 V or above	t_{SPcyc}	2.0	65536	t_{Tcyc}	Figure 60.52
			1.68 V or above (VCC) 1.65 V or above (VCC2)		4.0	65536		
SCK clock cycle input	Slave	—	2.70 V or above		2.0	65536		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		4.0	65536		
SCK clock high pulse width	Master	—	1.68 V or above (VCC) 1.65 V or above (VCC2)	t_{SPCKWH}	0.4	—	t_{SPcyc}	
	Slave	—						
SCK clock low pulse width	Master	—	1.68 V or above (VCC) 1.65 V or above (VCC2)	t_{SPCKWL}	0.4	—	t_{SPcyc}	
	Slave	—						
SCK clock rise and fall time	Output	—	2.70 V or above	t_{SPCKr}, t_{SPCKf}	—	3.3	ns	
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	6.6		
	Input	—	2.70 V or above		—	0.1*3	t_{SPcyc}	
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	0.1*3		
Data input setup time	Master	High Speed*1	2.70 V or above	t_{SU}	14.9 - (AST[2:0] settings)	—	ns	Figure 60.53, Figure 60.54
			1.68 V or above (VCC) 1.65 V or above (VCC2)		23.1 - (AST[2:0] settings)	—		
		Default*2	2.70 V or above		16.2 - (AST[2:0] settings)	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		23.8 - (AST[2:0] settings)	—		
	Slave	Default*2	2.70 V or above		2.5	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		4.5	—		

Table 60.41 SCI timing (Simple SPI) (2 of 3)

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

Parameter		High Speed/ Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
Data input hold time	Master	High Speed*1	2.70 V or above	t_H	-3.2 + (AST[2:0] settings)	—	ns	Figure 60.53, Figure 60.54
			1.68 V or above (VCC) 1.65 V or above (VCC2)		-3.2 + (AST[2:0] settings)	—		
		Default*2	2.70 V or above		-3.2 + (AST[2:0] settings)	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		-3.2 + (AST[2:0] settings)	—		
	Slave	Default*2	2.70 V or above		2.5	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		4.5	—		
Data output delay	Master	High Speed*1	2.70 V or above	t_{OD}	—	3.0	ns	Figure 60.53, Figure 60.54
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	4.5		
		Default*2	2.70 V or above		—	3.5		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	5.5		
	Slave	High Speed*1	2.70 V or above		—	15.0		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	23.0		
		Default*2	2.70 V or above		—	21.0		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	29.0		
Data output hold time	Master	High Speed*1	2.70 V or above	t_{OH}	-3.0	—	ns	Figure 60.53, Figure 60.54
			1.68 V or above (VCC) 1.65 V or above (VCC2)		-4.5	—		
		Default*2	2.70 V or above		-3.5	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		-5.5	—		
	Slave	Default*2	2.70 V or above		0.0	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		0.0	—		
Data rise and fall time	Output	—	2.70 V or above	t_{Dr}, t_{Df}	—	3.3	ns	
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	6.6		
	Input	—	2.70 V or above		—	1.0	µs	
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	1.0		
SS input setup time		—	1.68 V or above (VCC) 1.65 V or above (VCC2)	t_{LEAD}	1.0	—	t_{SPcyc}	Figure 60.55, Figure 60.56
SS input hold time		—	1.68 V or above (VCC) 1.65 V or above (VCC2)	t_{LAG}	1.0	—	t_{SPcyc}	
SS input rise and fall time		—	1.68 V or above (VCC) 1.65 V or above (VCC2)	t_{SSLr}, t_{SSLf}	—	1.0	µs	—

Table 60.41 SCI timing (Simple SPI) (3 of 3)

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

Parameter	High Speed/ Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
Slave access time	—	2.70 V or above	t_{SA}	—	3 × t_{Tcyc} + 25	ns	Figure 60.55, Figure 60.56
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	3 × t_{Tcyc} + 32		
Slave output release time	—	2.70 V or above	t_{REL}	—	3 × t_{Tcyc} + 25	ns	
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	3 × t_{Tcyc} + 32		

Note: t_{Tcyc}: TCLK cycle.
Note 1. Must use pins that have a letter appended to their name, for instance _A, _B, _C, to indicate group membership. SCI0, SCI1, SCI2, SCI3 and SCI9 are instance _A, SCI4 is instance _B.
Note 2. All pins of group membership can be used.
Note 3. 1 µs at the longest

Table 60.42 SCI timing (Clock synchronous mode) (1 of 4)

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

Parameter		High Speed/ Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
SCK clock cycle output	Master	—	2.70 V or above	t_{Scyc}	2.0	—	t_{Tcyc}	
			1.68 V or above (VCC) 1.65 V or above (VCC2)		4.0	—		
SCK clock cycle input	Slave	—	2.70 V or above		2.0	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		4.0	—		
SCK clock high pulse width	Master	—	2.70 V or above	t_{SCKWH}	0.4	—	t_{Scyc}	
	Slave		1.68 V or above (VCC) 1.65 V or above (VCC2)					
SCK clock low pulse width	Master	—	2.70 V or above	t_{SCKWL}	0.4	—	t_{Scyc}	
	Slave		1.68 V or above (VCC) 1.65 V or above (VCC2)					

Table 60.42 SCI timing (Clock synchronous mode) (2 of 4)

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

Parameter		High Speed/ Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
SCK clock rise and fall time	Output	—	2.70 V or above	t_{SCKr}, t_{SCKf}	—	3.3	ns	
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	6.6		
	Input	—	1.68 V or above (VCC) 1.65 V or above (VCC2)		—	0.1*3	t_{Scyc}	
Data input setup time	Master	High Speed*1	2.70 V or above	t_{SU}	15.1 - (AST[2:0] settings)	—	ns	
			1.68 V or above (VCC) 1.65 V or above (VCC2)		23.2 - (AST[2:0] settings)	—		
		Default*2	2.70 V or above		16.5 - (AST[2:0] settings)	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		24.2 - (AST[2:0] settings)	—		
	Slave	Default*2	2.70 V or above		3.3	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		5.3	—		
Data input hold time	Master	High Speed*1	2.70 V or above	t_H	-3.3 + (AST[2:0] settings)	—	ns	
			1.68 V or above (VCC) 1.65 V or above (VCC2)		-3.3 + (AST[2:0] settings)	—		
		Default*2	2.70 V or above		-3.2 + (AST[2:0] settings)	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		-3.2 + (AST[2:0] settings)	—		
	Slave	Default*2	2.70 V or above		3.0	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		5.0	—		

Table 60.42 SCI timing (Clock synchronous mode) (3 of 4)

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

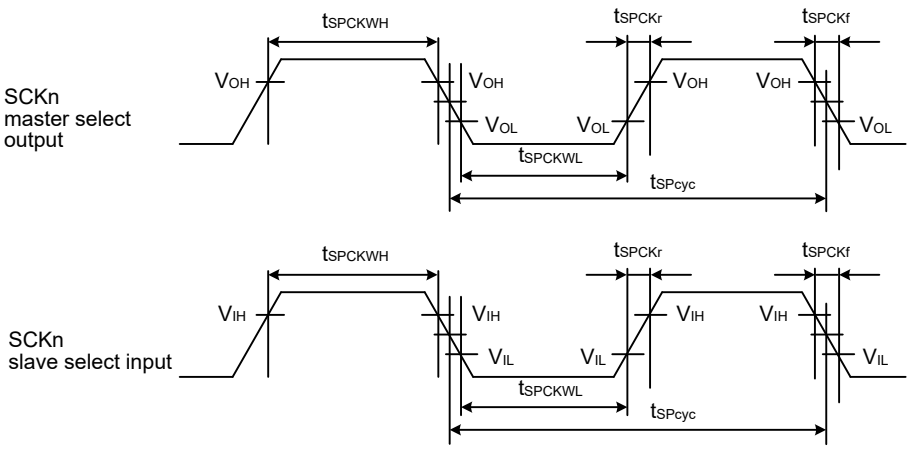
Parameter		High Speed/ Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
Data output delay	Master	High Speed*1	2.70 V or above	t_{OD}	—	5.0	ns	
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	5.0		
		Default*2	2.70 V or above		—	7.3		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	7.3		
	Slave	High Speed*1	2.70 V or above		—	15.0		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	23.0		
		Default*2	2.70 V or above		—	21.0		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	29.0		
Data output hold time	Master	High Speed*1	2.70 V or above	t_{OH}	-5.0	—	ns	
			1.68 V or above (VCC) 1.65 V or above (VCC2)		-5.0	—		
		Default*2	2.70 V or above		-7.3	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		-7.3	—		
	Slave	High Speed*1	2.70 V or above		0	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		0	—		
		Default*2	2.70 V or above		0	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		0	—		

Table 60.42 SCI timing (Clock synchronous mode) (4 of 4)

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

Parameter		High Speed/ Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
Data rise and fall time	Output	—	2.70 V or above	t_{Dr}, t_{Df}	—	3.3	ns	
		—	1.68 V or above (VCC) 1.65 V or above (VCC2)		—	6.6		
	Input	—	1.68 V or above (VCC) 1.65 V or above (VCC2)		—	1.0	µs	

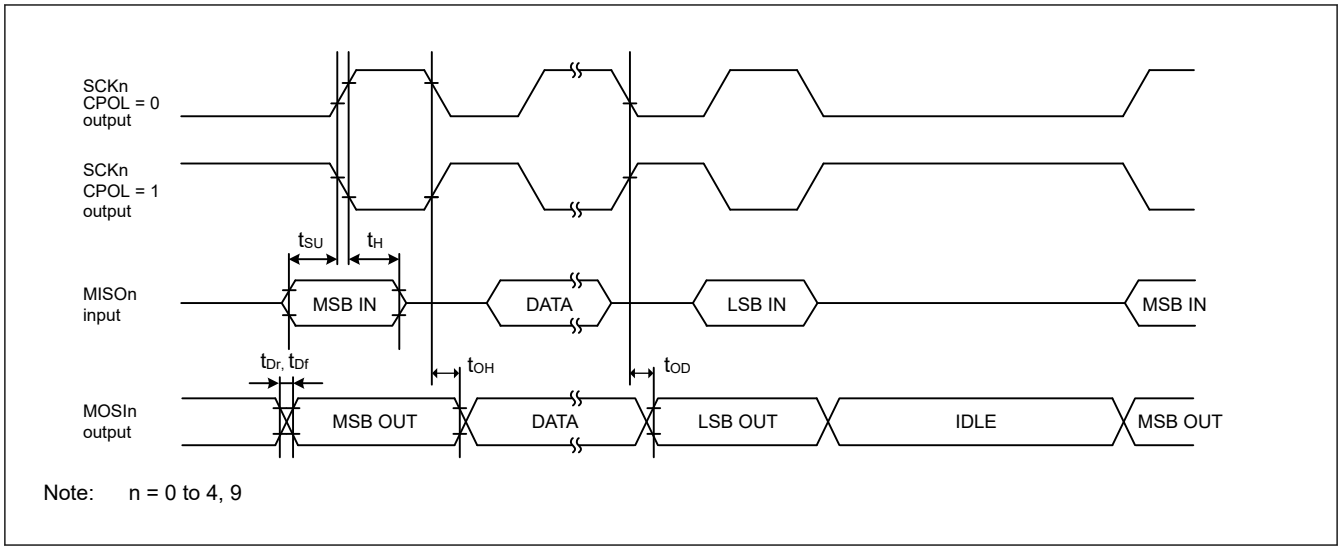
Note: t_{Tcyc}: TCLK cycle.
Note 1. Must use pins that have a letter appended to their name, for instance _A, _B, _C, to indicate group membership. SCI0, SCI1, SCI2, SCI3 and SCI9 are instance _A, SCI4 is instance _B.
Note 2. All pins of group membership can be used.
Note 3. 1 µs at the longest



V_{OH} = 0.7 × VCC or 0.7 × VCC2, V_{OL} = 0.3 × VCC or 0.3 × VCC2,
V_{IH} = 0.7 × VCC or 0.7 × VCC2, V_{IL} = 0.3 × VCC or 0.3 × VCC2

Note: n = 0 to 4, 9

Figure 60.52 SCI simple SPI mode clock timing



Note: n = 0 to 4, 9

Figure 60.53 SCI simple SPI mode timing for master when CPHA = 0

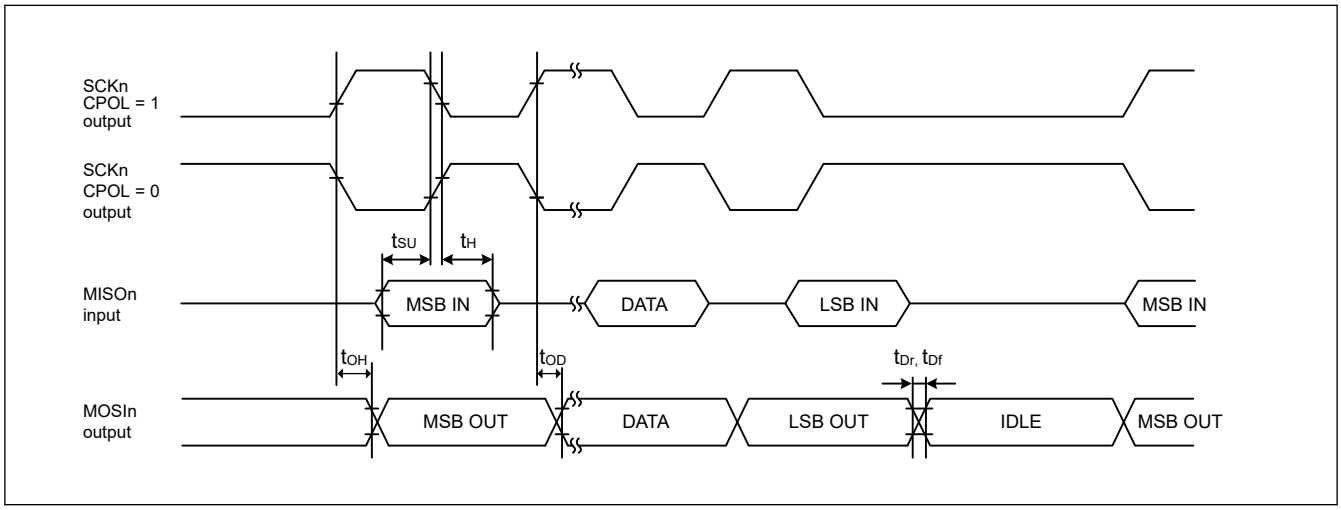
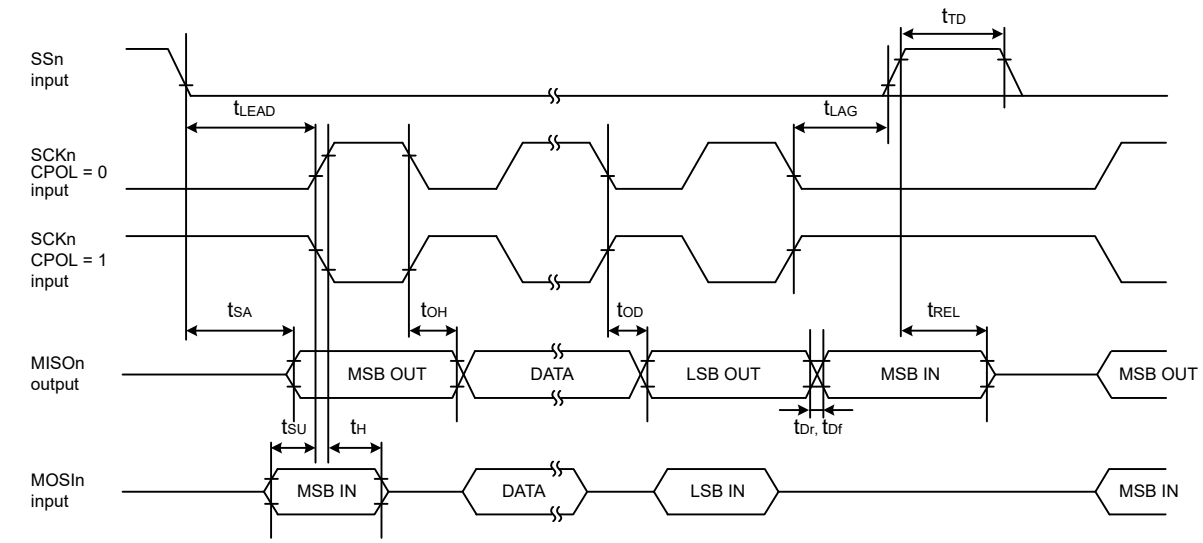
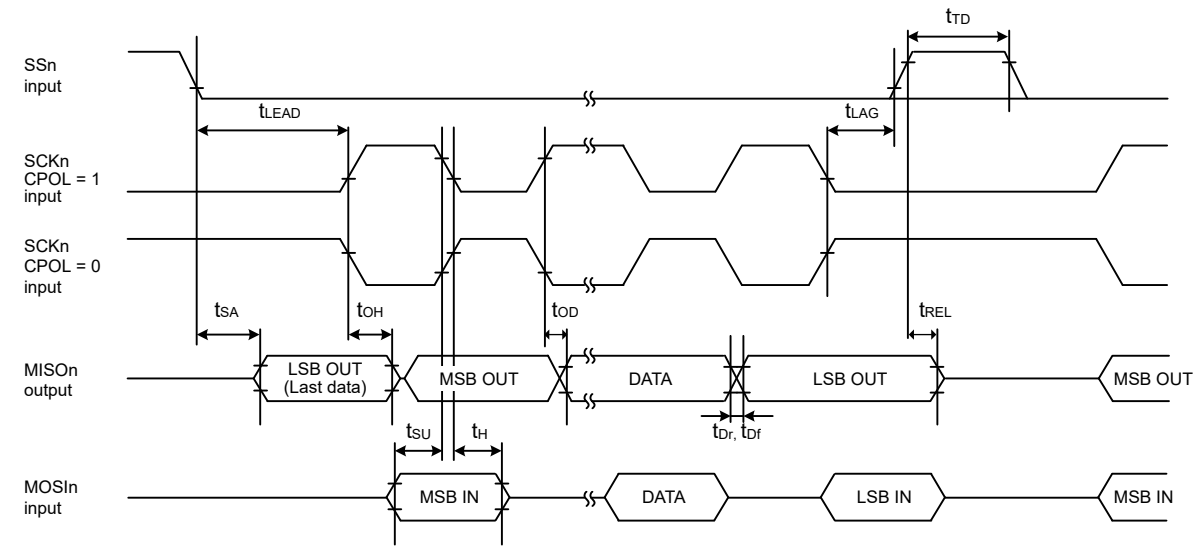


Figure 60.54 SCI simple SPI mode timing for master when CPHA = 1



Note: n = 0 to 4, 9

Figure 60.55 SCI simple SPI mode timing for slave when CPHA = 0



Note: n = 0 to 4, 9

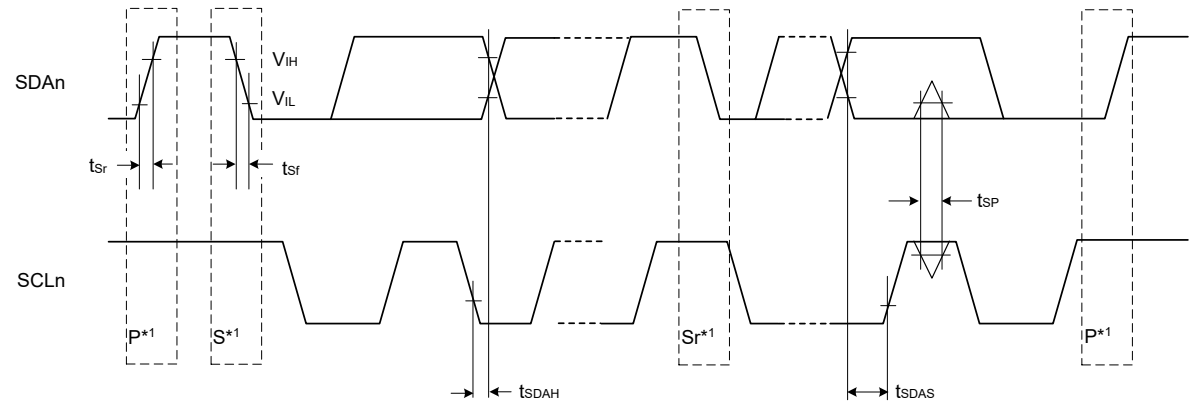
Figure 60.56 SCI simple SPI mode timing for slave when CPHA = 1

Table 60.43 SCI timing (Simple IIC mode)

Conditions: Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.
VCC: 1.68V or above, VCC2: 1.65V or above
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

Parameter		Symbol	Min	Max	Unit	Note
Simple IIC (Standard mode)	SCL, SDA input rise time	t_{Sr}	—	1000	ns	
	SCL, SDA input fall time	t_{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	4 × t_{Tcyc}	ns	
	Data input setup time	t_{SDAS}	250	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b *1	—	400	pF	
Simple IIC (Fast mode)	SCL, SDA input rise time	t_{Sr}	—	300	ns	
	SCL, SDA input fall time	t_{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	4 × t_{Tcyc}	ns	
	Data input setup time	t_{SDAS}	100	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b *1	—	400	pF	

Note: t_{Tcyc}: TCLK cycle.
Note 1. C_b indicates the total capacity of the bus line.



Test conditions:
V_{IH} = VCC × 0.7 or VCC2 × 0.7
V_{IL} = VCC × 0.3 or VCC2 × 0.3
V_{OL} = 0.6 V, I_{OL} = 6 mA

Note: n = 0 to 4, 9
Note 1. S, P, and Sr indicate the following conditions:
S: Start condition
P: Stop condition
Sr: Restart condition

Figure 60.57 SCI simple IIC mode timing

60.3.10 SPI Timing

Table 60.44 SPI timing (1 of 4)

Conditions:

1. High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
2. Use pins that have a letter appended to their names, for instance “_A” or “_B” to indicate group membership.
3. Load capacitance C = 15pF is applied to the VCC/VCC2 condition "3.00 V or above".

Parameter		VCC/VCC2	Symbol	Min	Max	Unit	Note
RSPCK clock cycle	Master	3.00 V or above	t_{SPcyc}	2.0	4096	t_{Tcyc}	Figure 60.58
		2.70 V or above		2.0	4096		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		4.0	4096		
	Slave	3.00 V or above		2.0	—		
		2.70 V or above		2.0	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		4.0	—		
RSPCK clock high pulse width	Master	1.68 V or above (VCC) 1.65 V or above (VCC2)	t_{SPCKWH}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	—	ns	
	Slave	1.68 V or above (VCC) 1.65 V or above (VCC2)		0.4	—	t_{SPcyc}	
RSPCK clock low pulse width	Master	1.68 V or above (VCC) 1.65 V or above (VCC2)	t_{SPCKWL}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	—	ns	
	Slave	1.68 V or above (VCC) 1.65 V or above (VCC2)		0.4	—	t_{SPcyc}	
RSPCK clock rise and fall time	Output	3.00 V or above	t_{SPCKr}, t_{SPCKf}	—	1.66	ns	
		2.70 V or above		—	3.30		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	6.60		
	Input	3.00 V or above		—	0.1*1	t_{SPcyc}	
		2.70 V or above		—	0.1*1		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	0.1*1		

Table 60.44 SPI timing (2 of 4)

Conditions:

1. High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
2. Use pins that have a letter appended to their names, for instance “_A” or “_B” to indicate group membership.
3. Load capacitance C = 15pF is applied to the VCC/VCC2 condition "3.00 V or above".

Parameter		VCC/VCC2	Symbol	Min	Max	Unit	Note
Data input setup time	Master	3.00 V or above	t_{SU}	-2.5	—	ns	Figure 60.59, Figure 60.60
		2.70 V or above		0.0	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		0.0	—		
	Slave	3.00 V or above		2.5	—		
		2.70 V or above		2.5	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		2.5	—		
Data input hold time	Master	3.00 V or above	t_{H}	7.5	—	ns	
		2.70 V or above		7.5	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		9.5	—		
	Slave	3.00 V or above		2.5	—		
		2.70 V or above		2.5	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		5.5	—		
SSL setup time	Master	3.00 V or above	t_{LEAD}	1 × t_{SPcyc} - 10	8 × t_{SPcyc} + 10	ns	
		2.70 V or above		1 × t_{SPcyc} - 10	8 × t_{SPcyc} + 10		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		1 × t_{SPcyc} - 10	8 × t_{SPcyc} + 10		
	Slave	3.00 V or above		4.0	—	t_{Tcyc}	
		2.70 V or above		4.0	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		4.0	—		
SSL hold time	Master	3.00 V or above	t_{LAG}	1 × t_{SPcyc} - 10	8 × t_{SPcyc} + 10	ns	
		2.70 V or above		1 × t_{SPcyc} - 10	8 × t_{SPcyc} + 10		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		1 × t_{SPcyc} - 10	8 × t_{SPcyc} + 10		
	Slave	3.00 V or above		4.0	—	t_{Tcyc}	
		2.70 V or above		4.0	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		4.0	—		

Table 60.44 SPI timing (3 of 4)

Conditions:

1. High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
2. Use pins that have a letter appended to their names, for instance “_A” or “_B” to indicate group membership.
3. Load capacitance C = 15pF is applied to the VCC/VCC2 condition "3.00 V or above".

Parameter		VCC/VCC2	Symbol	Min	Max	Unit	Note
TI SSP SS input setup time	Slave	3.00 V or above	t_{TISS}	2.5	—	ns	Figure 60.64
		2.70 V or above		2.5	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		2.5	—		
TI SSP SS input hold time	Slave	3.00 V or above	t_{TISH}	2.5	—	ns	
		2.70 V or above		2.5	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		5.5	—		
TI SSP next-access time	Slave	3.00 V or above	t_{TIND}	2 × t_{Tcyc} + SLNDL × t_{Tcyc}	—	ns	
		2.70 V or above		2 × t_{Tcyc} + SLNDL × t_{Tcyc}	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		2 × t_{Tcyc} + SLNDL × t_{Tcyc}	—		
TI SSP master SS output delay	Master	3.00 V or above	t_{TISSOD}	—	4.0	ns	Figure 60.61
		2.70 V or above		—	8.0		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	8.0		
Data output delay time	Master	3.00 V or above	t_{OD1}	—	2.0	ns	Figure 60.59, Figure 60.60
		2.70 V or above		—	3.0		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	6.0		
		3.00 V or above	t_{OD2}	—	2.5		
		2.70 V or above		—	2.5		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	4.5		
	Slave	3.00 V or above	t_{OD}	—	12.5		
		2.70 V or above		—	16.0		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	24.0		
Data output hold time	Master	3.00 V or above	t_{OH}	-2.5	—	ns	Figure 60.59, Figure 60.60
		2.70 V or above		-2.5	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		-4.5	—		
	Slave	3.00 V or above		0.0	—		
		2.70 V or above		0.0	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		0.0	—		

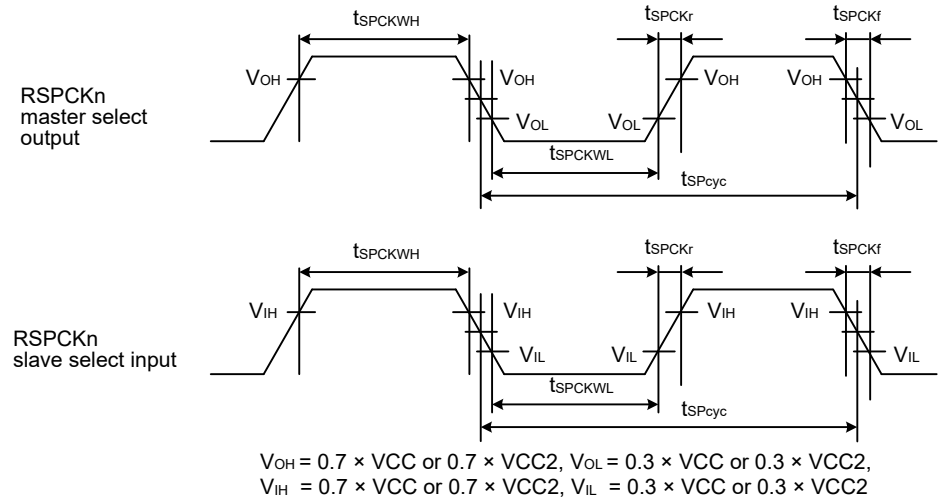
Table 60.44 SPI timing (4 of 4)

Conditions:

1. High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
2. Use pins that have a letter appended to their names, for instance “_A” or “_B” to indicate group membership.
3. Load capacitance C = 15pF is applied to the VCC/VCC2 condition "3.00 V or above".

Parameter		VCC/VCC2	Symbol	Min	Max	Unit	Note
Successive transmission delay time	Master	3.00 V or above	t_{TD}	$t_{SPcyc} + 2 \times t_{Tcyc}$	$8 \times t_{SPcyc} + 2 \times t_{Tcyc}$	ns	Figure 60.59, Figure 60.60
		2.70 V or above		$t_{SPcyc} + 2 \times t_{Tcyc}$	$8 \times t_{SPcyc} + 2 \times t_{Tcyc}$		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		$t_{SPcyc} + 2 \times t_{Tcyc}$	$8 \times t_{SPcyc} + 2 \times t_{Tcyc}$		
	Slave	3.00 V or above		t_{Tcyc}	—	ns	
		2.70 V or above		t_{Tcyc}	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		t_{Tcyc}	—		
MOSI and MISO rise and fall time	Output	3.00 V or above	t_{Dr}, t_{Df}	—	1.66	ns	Figure 60.59, Figure 60.60
		2.70 V or above		—	3.30		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	6.60		
	Input	3.00 V or above		—	1.0	µs	
		2.70 V or above		—	1.0		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	1.0		
SSL rise and fall time	Output	3.00- V or above	t_{SSLr}, t_{SSLf}	—	1.66	ns	Figure 60.59, Figure 60.60
		2.70 V or above		—	3.30		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	6.60		
	Input	3.00 V or above		—	1.0	µs	
		2.70 V or above		—	1.0		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	1.0		
Slave access time	Slave	3.00 V or above	t_{SA}	—	20.0	ns	Figure 60.62, Figure 60.63
		2.70 V or above		—	20.0		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	25.0		
Slave output release time	Slave	3.00 V or above	t_{REL}	—	20.0	ns	
		2.70 V or above		—	20.0		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	25.0		

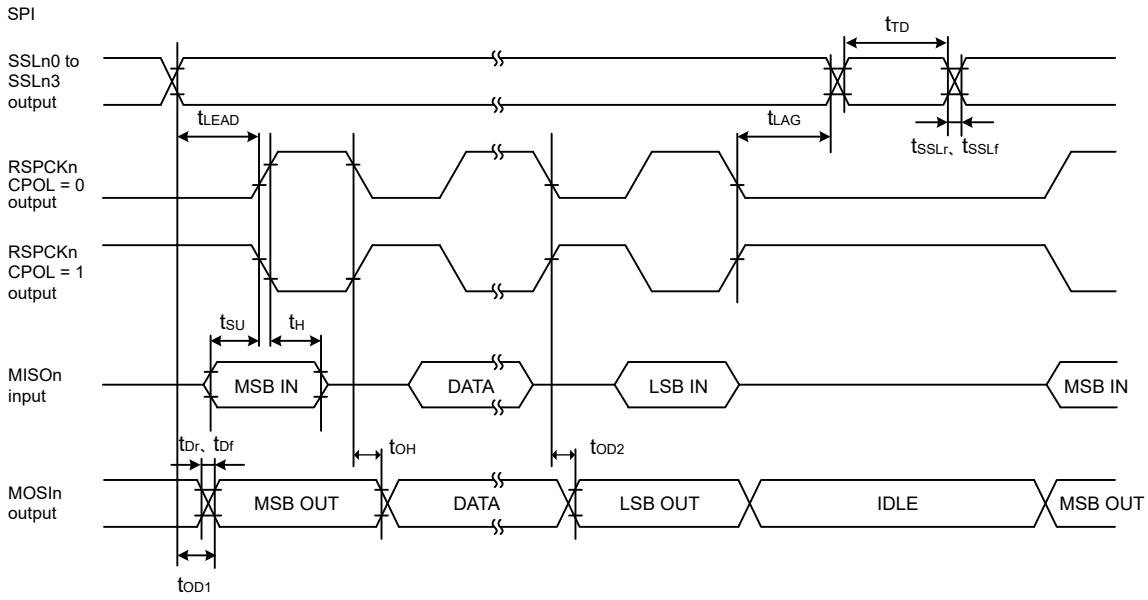
Note: t_{Tcyc}: TCLK cycle.
Note 1. 1 µs at the longest



V_{OH} = 0.7 × VCC or 0.7 × VCC2, V_{OL} = 0.3 × VCC or 0.3 × VCC2,
V_{IH} = 0.7 × VCC or 0.7 × VCC2, V_{IL} = 0.3 × VCC or 0.3 × VCC2

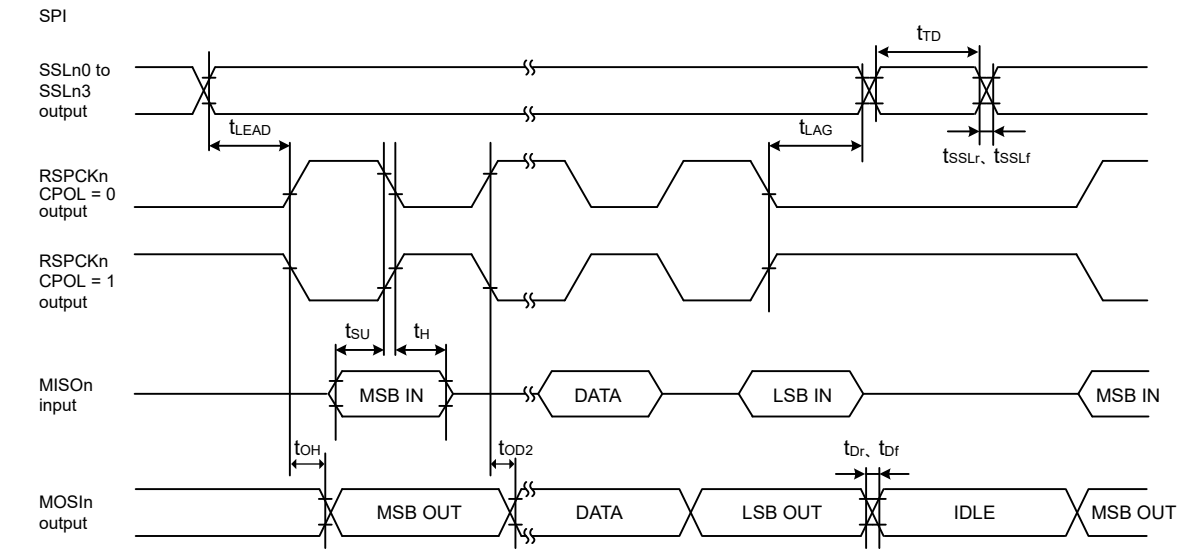
Note: n = A or B

Figure 60.58 SPI clock timing



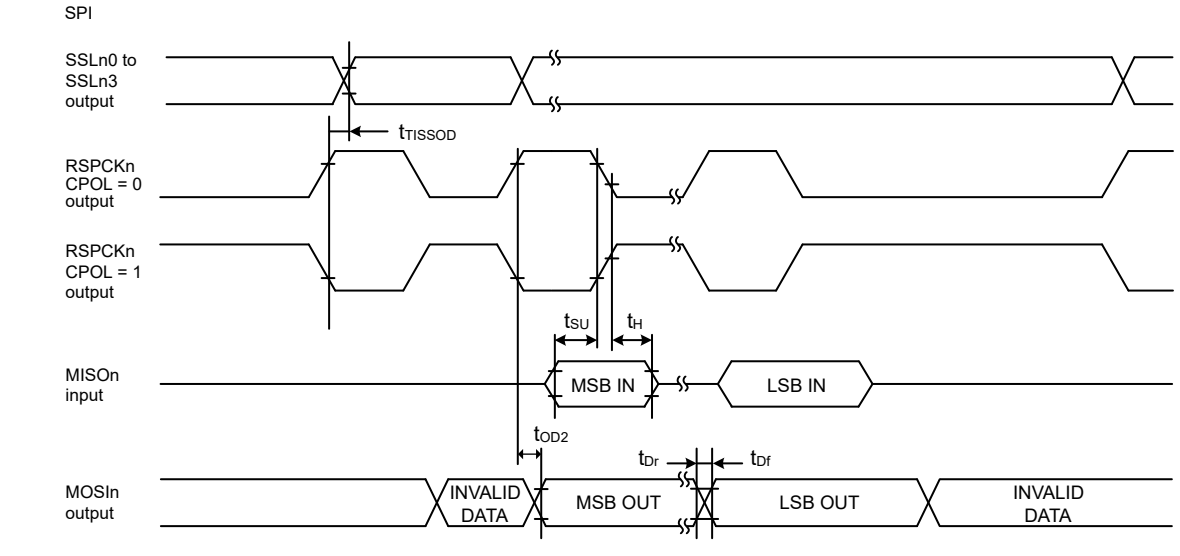
Note: n = A or B

Figure 60.59 SPI timing for Motorola SPI master when CPHA = 0



Note: n = A or B

Figure 60.60 SPI timing for Motorola SPI master when CPHA = 1



Note: n = A or B

Figure 60.61 SPI timing for TI SSP master

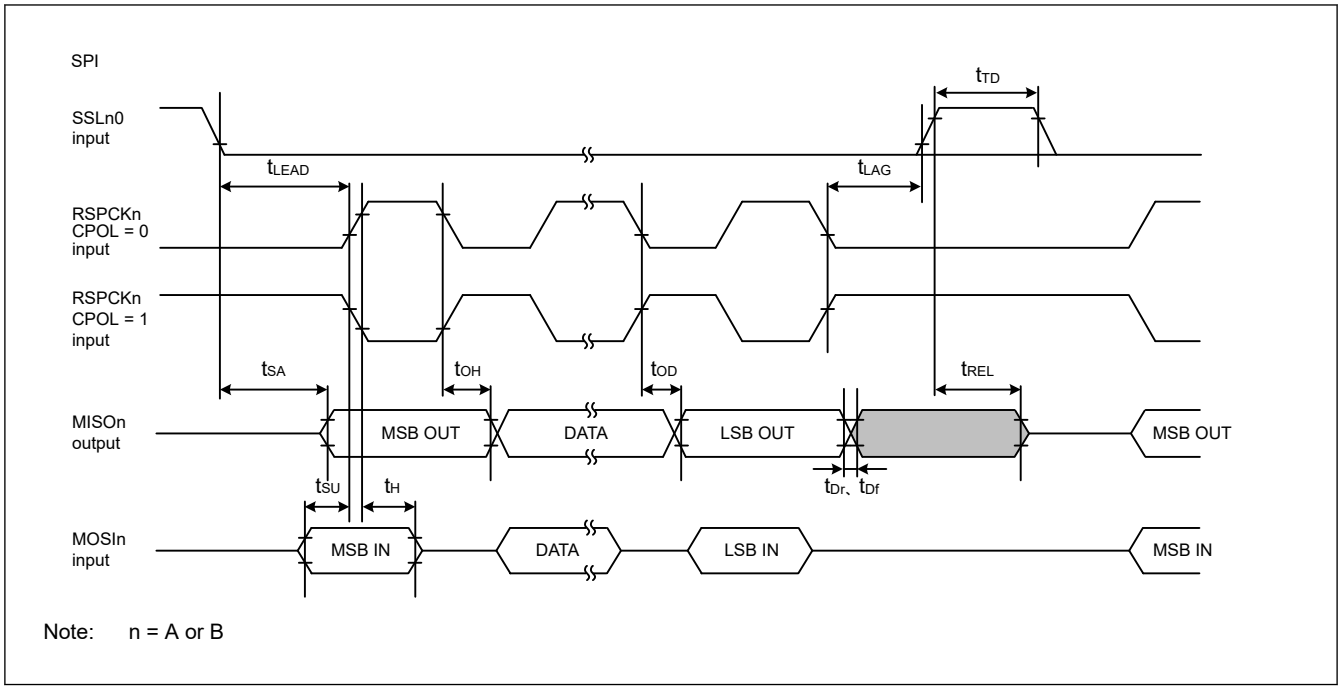


Figure 60.62 SPI timing for Motorola SPI slave when CPHA = 0

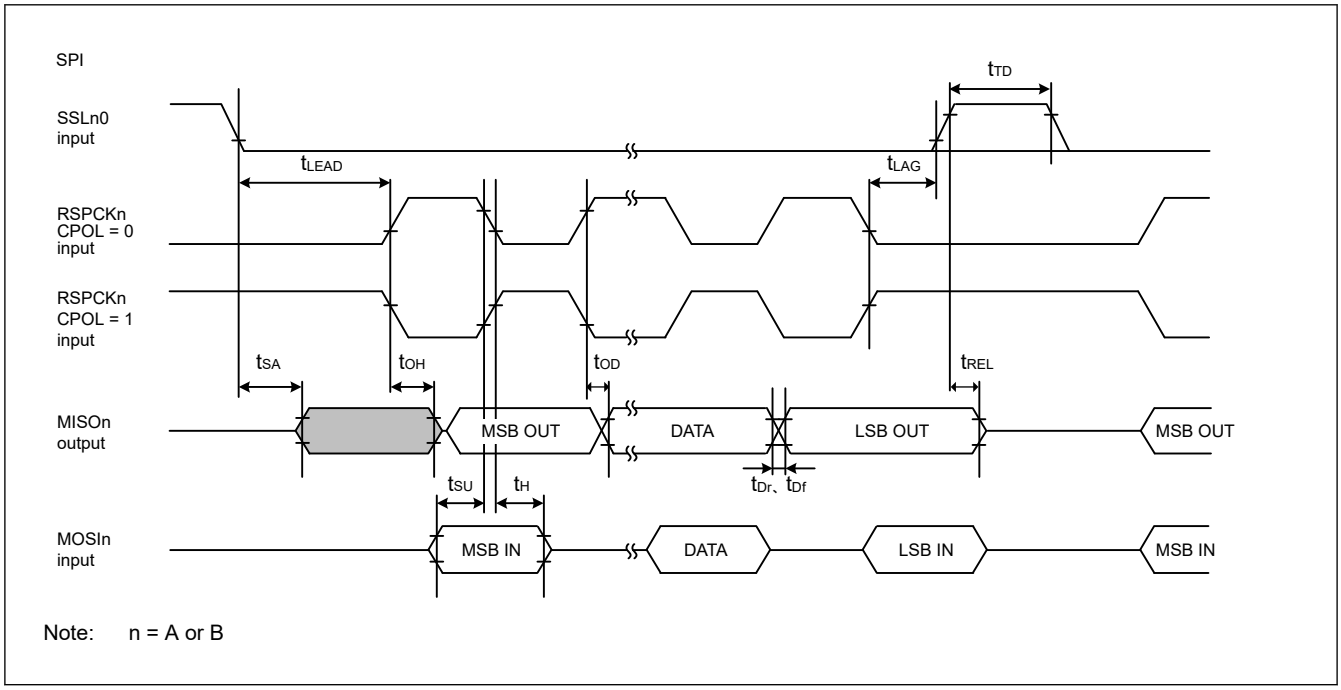


Figure 60.63 SPI timing for Motorola SPI slave when CPHA = 1

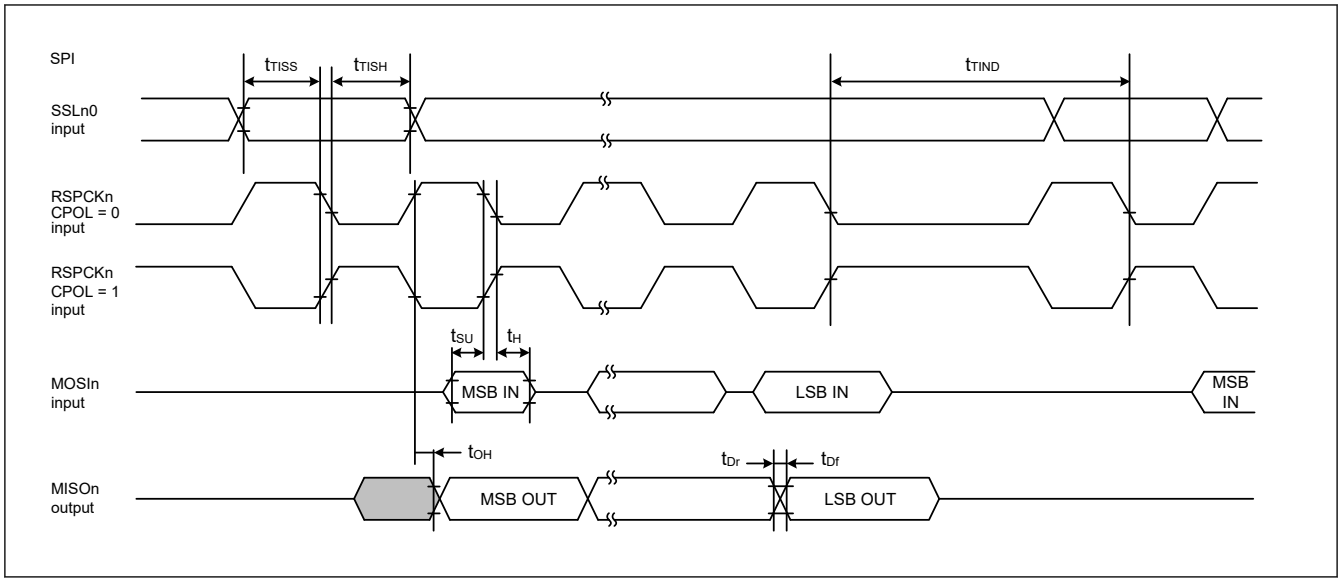


Figure 60.64 SPI timing for TI SSP slave when transmit with delay between frames

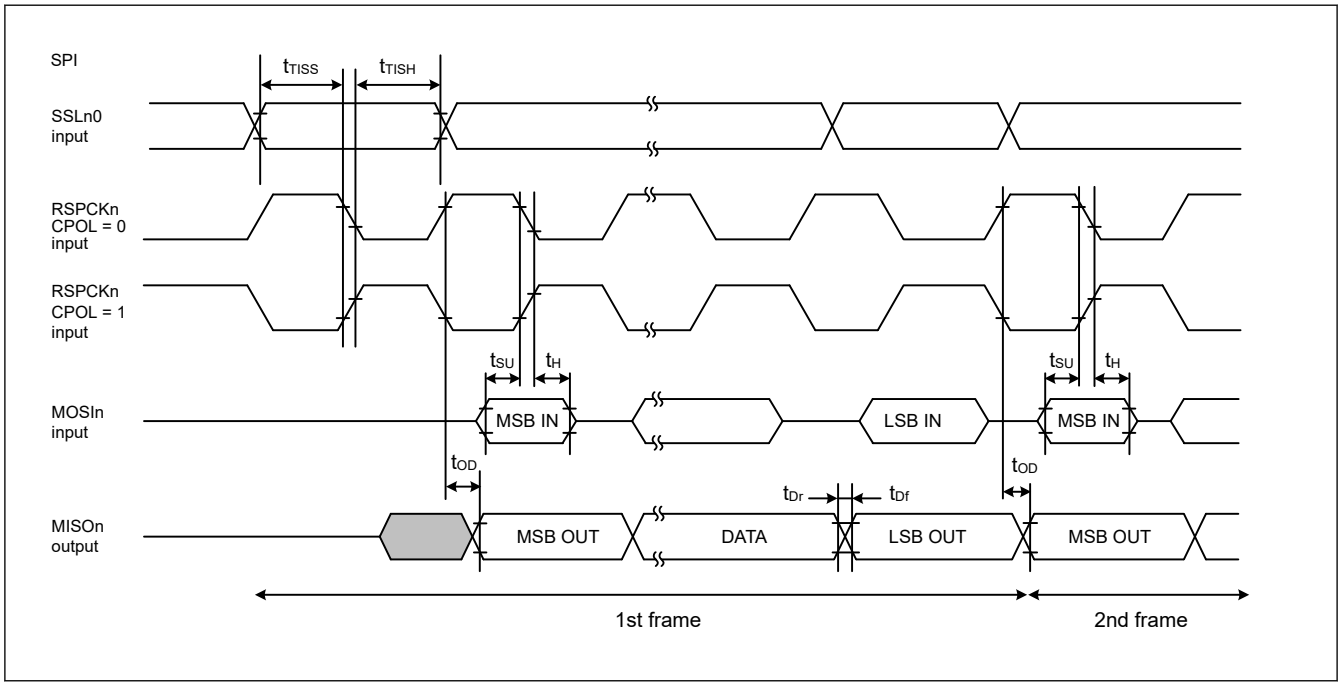


Figure 60.65 SPI timing for TI SSP slave when transmit with no delay between frames

60.3.11 OSPI Timing

Table 60.45 OSPI timing (1 of 2)

Conditions:
High-speed high drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_SCLK, OM_SCLKN, OM_SIO7-0, OM_DQS.
High drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_CS0, OM_CS1.
Load capacitance C = 20 pF

Item		Symbol	VCC/VCC2	Min	Max	Unit	Note
Cycle time	SDR without OM_DQS	t_{PERIOD}	2.70 V or above	20	—	ns	Figure 60.66
			1.65 V to 2.00 V	20	—		
	SDR with OM_DQS/DDR	t_{PERIOD}	2.70 V or above	10	—	ns	
			1.65 V to 2.00 V	10	—		
Clock output slew rate		t_{SRck}	2.70 V or above	0.56	—	V/ns	
			1.65 V to 2.00 V	0.56	—		
Clock Duty cycle-distortion		t_{CKDCD}	2.70 V or above	0	0.5	ns	
			1.65 V to 2.00 V	0	0.5		
Clock Minimum Pulse width		t_{CKMPW}	2.70 V or above	4.5	—	ns	
			1.65 V to 2.00 V	4.5	—		
Differential clock crossing voltage		V_{ox}(AC)	2.70 V or above	0.4 × VCC2	0.6 × VCC2	V	
			1.65 V to 2.00 V	0.4 × VCC2	0.6 × VCC2		
DS Duty cycle distortion		t_{DSDCD}	2.70 V or above	0	0.4	ns	
			1.65 V to 2.00 V	0	0.4		
DS Minimum Pulse width		t_{DSMPW}	2.70 V or above	4.1	—	ns	
			1.65 V to 2.00 V	4.1	—		
Data input/output slew rate		t_{SR}	2.70 V or above	1.03	—	ns	
			1.65 V to 2.00 V	0.56	—		
Data input setup time (to OM_SCLK/ OM_SCLKN)	SDR without OM_DQS	t_{SU}	2.70 V or above	8.0	—	ns	Figure 60.67
			1.65 V to 2.00 V	12.5	—		
Data input hold time (to OM_SCLK/ OM_SCLKN)		t_H	2.70 V or above	0.5	—	ns	
			1.65 V to 2.00 V	0.5	—		
Data output valid time		t_{OV}*2	2.70 V or above	—	4.0	ns	
			1.65 V to 2.00 V	—	4.0		
Data output hold time		t_{OH}	2.70 V or above	−2.0	—	ns	
			1.65 V to 2.00 V	−2.0	—		
Data output buffer off time		t_{BOFF}	2.70 V or above	−2.0	—	ns	
			1.65 V to 2.00 V	−2.0	—		

Table 60.45 OSPI timing (2 of 2)

Conditions:
High-speed high drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_SCLK, OM_SCLKN, OM_SIO7-0, OM_DQS.
High drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_CS0, OM_CS1.
Load capacitance C = 20 pF

Item		Symbol	VCC/VCC2	Min	Max	Unit	Note
Data input setup time (to OM_DQS)	SDR with OM_DQS/DDR	t_{SU}	2.70 V or above	−0.9	—	ns	Figure 60.68, Figure 60.69
			1.65 V to 2.00 V	−0.9	—		
Data input hold time (to OM_DQS)		t_{H}	2.70 V or above	3.2	—	ns	
			1.65 V to 2.00 V	3.2	—		
Data output valid time		t_{OV}*2	2.70 V or above	—	$t_{PERIOD}/4$ + 0.9	ns	
			1.65 V to 2.00 V	—	$t_{PERIOD}/4$ + 0.9		
Data output hold time		t_{HO}	2.70 V or above	1.1	—	ns	
			1.65 V to 2.00 V	1.1	—		
Data output buffer off time		t_{BOFF}	2.70 V or above	1.1	—	ns	
			1.65 V to 2.00 V	1.1	—		
Clock Low to CS Low		t_{CKLCSL}	2.70 V or above	8	—	ns	Figure 60.67, Figure 60.68, Figure 60.69
			1.65 V to 2.00 V	8	—		
CS Low to Clock High		t_{CSLCKH}*3	2.70 V or above	8	—	ns	
			1.65 V to 2.00 V	8	—		
Cock Low to CS High		t_{CKLCSH}	2.70 V or above	8	—	ns	
			1.65 V to 2.00 V	8	—		
CS High to Clock High		t_{CSHCKH}	2.70 V or above	8	—	ns	
			1.65 V to 2.00 V	8	—		
DS Low output to CS High		t_{DSLCSH}	2.70 V or above	0.8 × t_{PERIOD}	—	ns	Figure 60.70
			1.65 V to 2.00 V	0.8 × t_{PERIOD}	—		
CS High to DS Tri-State		t_{CSHDST}	2.70 V or above	—	t_{PERIOD}	ns	
			1.65 V to 2.00V	—	t_{PERIOD}		
CS Low to DS Low input*1		t_{CSLDSL}	2.70 V or above	0	18.5	ns	
			1.65 V to 2.00 V	0	12.5		
DS Tri-State to CS Low		t_{DSTCSL}	2.70 V or above	0	—	ns	
			1.65 V to 2.00 V	0	—		

Note 1. This restriction does not need to be met when using the JESD251 Profile 1.0 memory with an external pull-down attached to the OM_DQS pin.
Note 2. Condition: COMCFG.OEASTEX = 1
Note 3. Condition: LIOCFGCSx.CSASTEX = 1

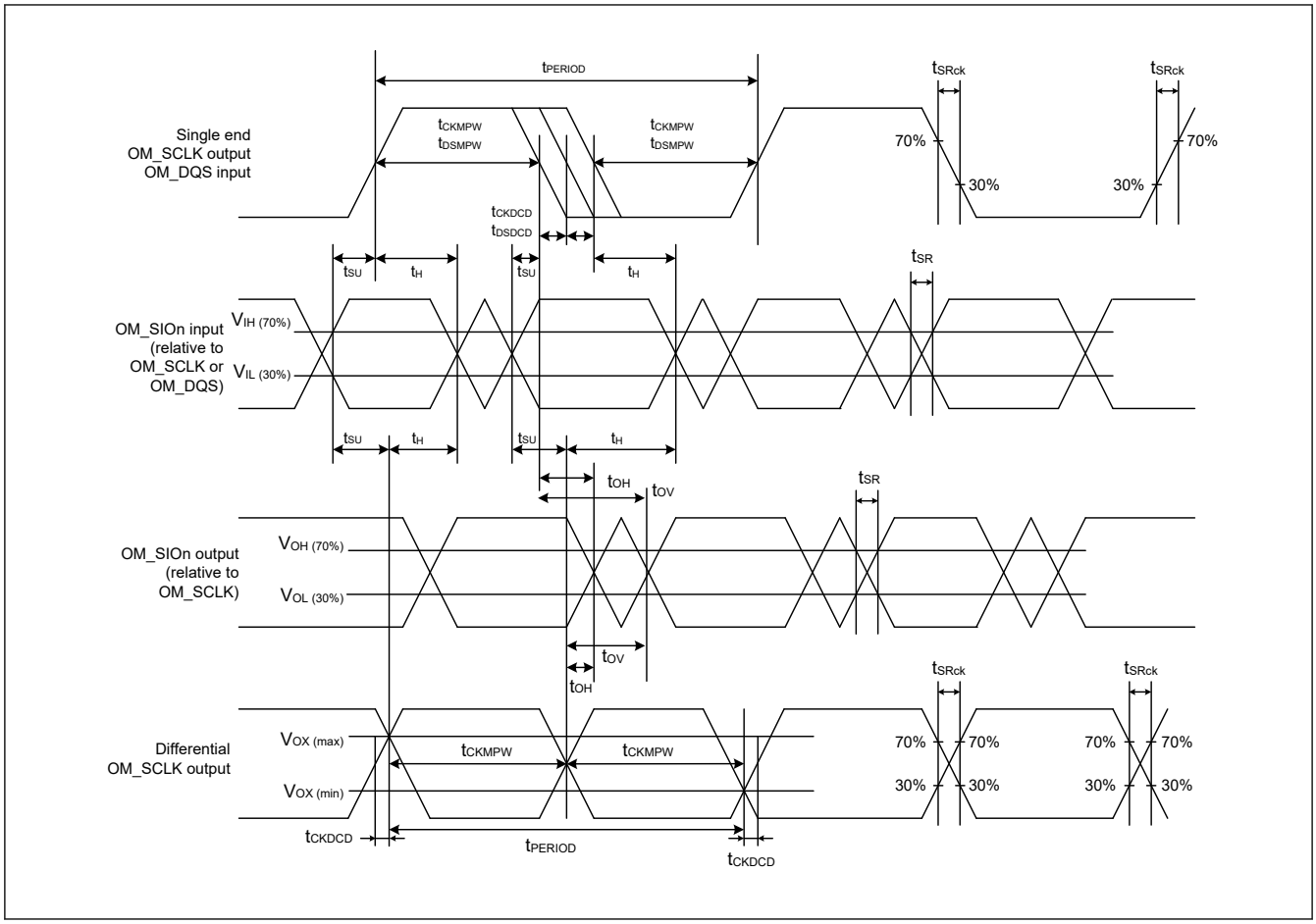


Figure 60.66 OSPI clock / DS timing

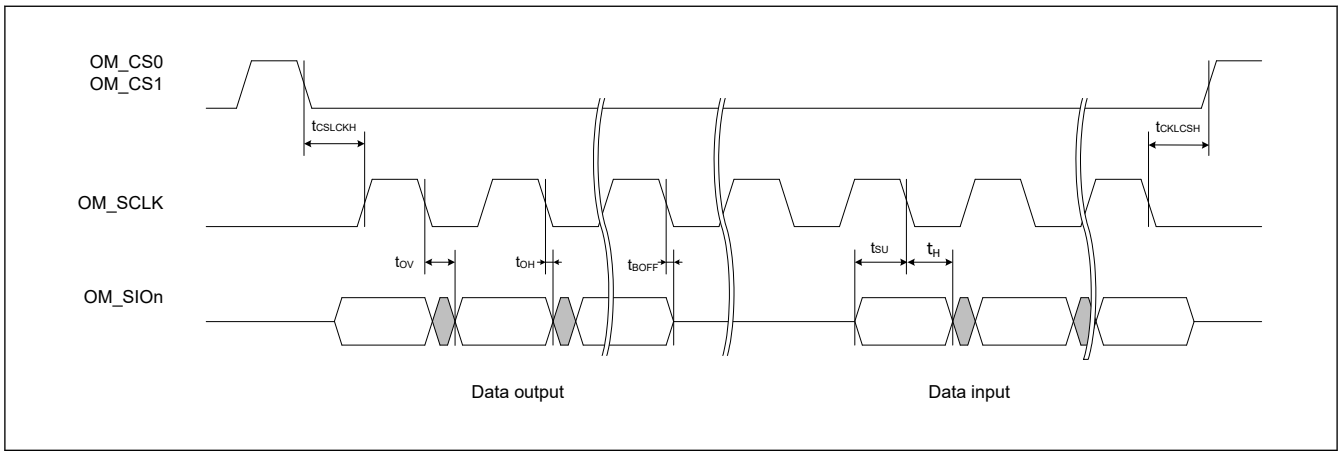


Figure 60.67 SDR transmit/receive timing (1S-1S-1S, 1S-2S-2S, 2S-2S-2S, 1S-4S-4S, 4S-4S-4S)

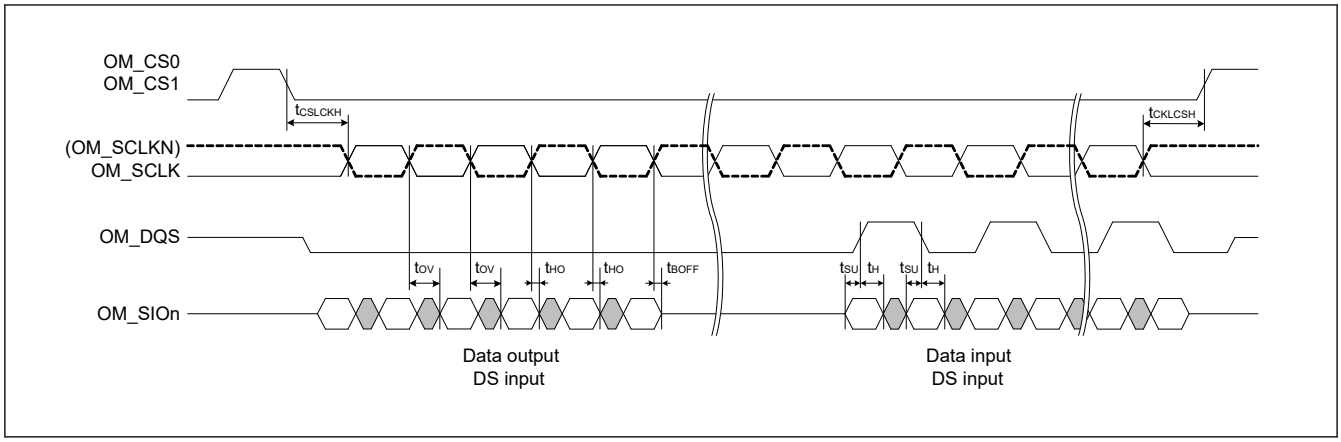


Figure 60.68 DDR transmit/receive timing (4S-4D-4D, 8D-8D-8D)

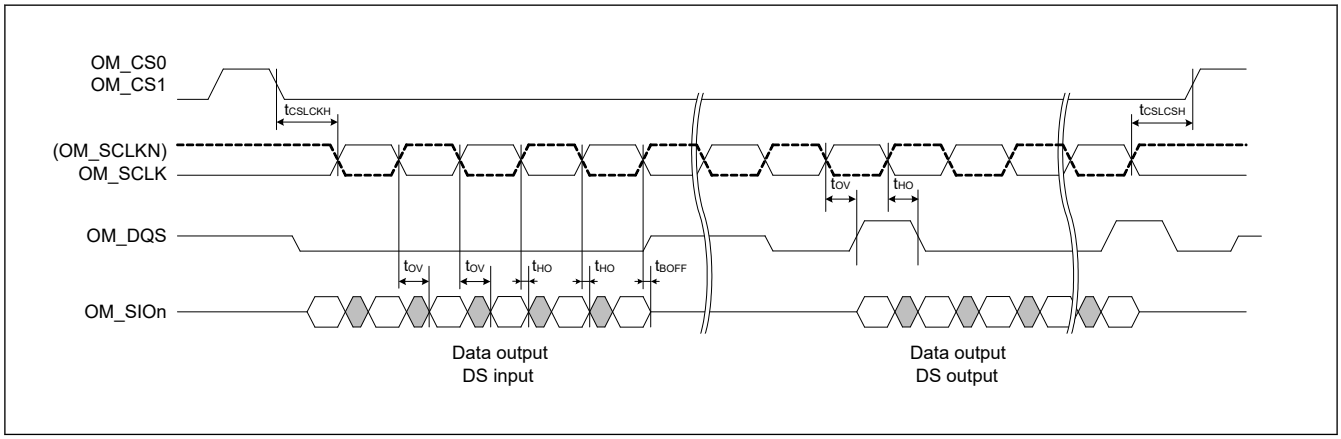


Figure 60.69 DDR transmit/receive timing (HyperRAM write)

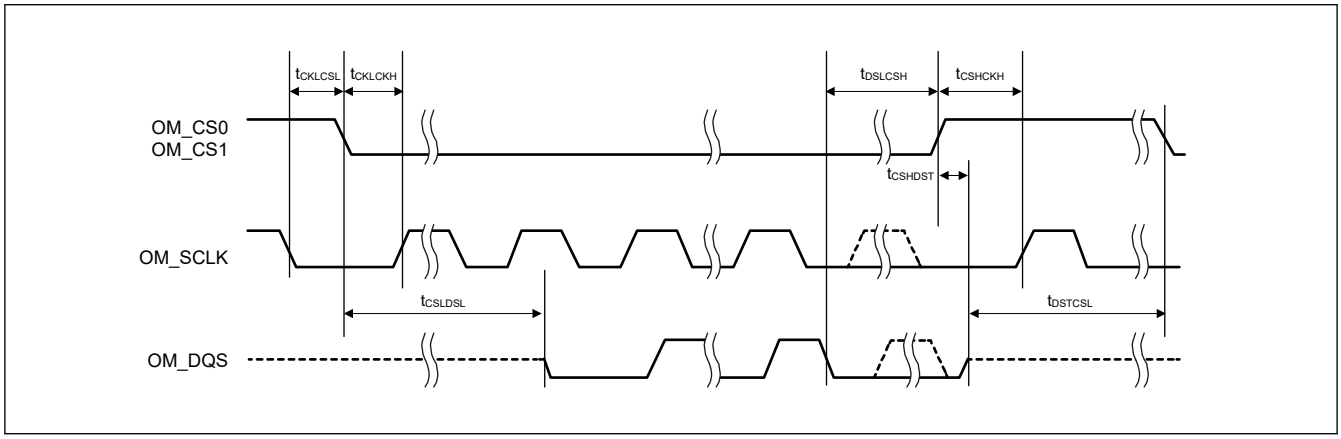


Figure 60.70 DS to CS signal timing

60.3.12 IIC Timing

Table 60.46 IIC timing (1) (1 of 2)

(1) Conditions: Middle drive output is selected when VCC is 2.70 V or above, High drive output is selected when VCC is 1.68 to 1.95 V in the port drive capability bit in the PmnPFS register for the following pins: SDA0_B, SCL0_B, SDA1_B, SCL1_B
(2) The following pins do not require setting: SCL0_A, SDA0_A. SCL1_A, SDA1_A
(3) Use pins that have a letter appended to their names, for instance "_A" or "_B", to indicate group membership.
For the IIC interface, the AC portion of the electrical characteristics is measured for each group.

Parameter		Symbol	VCC	Min	Max	Unit	Test conditions
IIC (Standard mode, SMBus) ICFER.FMPE = 0 when VCC is 2.70 V or above, ICFER.FMPE = 1 when VCC is 1.68 to 1.95 V	SCL input cycle time	t_{SCL}	2.70 V or above	6 (12) × t_{IICcyc} + 1300	—	ns	Figure 60.71
			1.68 to 1.95 V				
	SCL input high pulse width	t_{SCLH}	2.70 V or above	3 (6) × t_{IICcyc} + 300	—	ns	
			1.68 to 1.95 V				
	SCL input low pulse width	t_{SCLL}	2.70 V or above	3 (6) × t_{IICcyc} + 300	—	ns	
			1.68 to 1.95 V				
	SCL, SDA rise time	t_{Sr}	2.70 V or above	—	1000	ns	
			1.68 to 1.95 V				
	SCL, SDA fall time	t_{Sf}	2.70 V or above	—	300	ns	
			1.68 to 1.95 V				
	SCL, SDA input spike pulse removal time	t_{SP}	2.70 V or above	0	1 (4) × t_{IICcyc}	ns	
			1.68 to 1.95 V				
	SDA input bus free time when wakeup function is disabled	t_{BUF}	2.70 V or above	3 (6) × t_{IICcyc} + 300	—	ns	
			1.68 to 1.95 V				
	SDA input bus free time when wakeup function is enabled	t_{BUF}	2.70 V or above	3 (6) × t_{IICcyc} + 4 × t_{Pcyc} + 300	—	ns	
			1.68 to 1.95 V				
	START condition input hold time when wakeup function is disabled	t_{STAH}	2.70 V or above	t_{IICcyc} + 300	—	ns	
			1.68 to 1.95 V				
	START condition input hold time when wakeup function is enabled	t_{STAH}	2.70 V or above	1 (5) × t_{IICcyc} + t_{Pcyc} + 300	—	ns	
			1.68 to 1.95 V				
	Repeated START condition input setup time	t_{STAS}	2.70 V or above	1000	—	ns	
			1.68 to 1.95 V				
	STOP condition input setup time	t_{STOS}	2.70 V or above	1000	—	ns	
			1.68 to 1.95 V				
	Data input setup time	t_{SDAS}	2.70 V or above	t_{IICcyc} + 50	—	ns	
			1.68 to 1.95 V				
	Data input hold time	t_{SDAH}	2.70 V or above	0	—	ns	
			1.68 to 1.95 V				
	SCL, SDA capacitive load	C_b	2.70 V or above	—	400	pF	
			1.68 to 1.95 V				

Table 60.46 IIC timing (1) (2 of 2)

(1) Conditions: Middle drive output is selected when VCC is 2.70 V or above, High drive output is selected when VCC is 1.68 to 1.95 V in the port drive capability bit in the PmnPFS register for the following pins: SDA0_B, SCL0_B, SDA1_B, SCL1_B
(2) The following pins do not require setting: SCL0_A, SDA0_A. SCL1_A, SDA1_A
(3) Use pins that have a letter appended to their names, for instance "_A" or "_B", to indicate group membership.
For the IIC interface, the AC portion of the electrical characteristics is measured for each group.

Parameter		Symbol	VCC	Min	Max	Unit	Test conditions
IIC (Fast mode) ICFER.FMPE = 0 when VCC is 2.70 V or above, ICFER.FMPE = 1 when VCC is 1.68 to 1.95 V	SCL input cycle time	t_{SCL}	2.70 V or above	6 (12) × t_{IICcyc} + 600	—	ns	Figure 60.71
			1.68 to 1.95 V				
	SCL input high pulse width	t_{SCLH}	2.70 V or above	3 (6) × t_{IICcyc} + 300	—	ns	
			1.68 to 1.95 V				
	SCL input low pulse width	t_{SCLL}	2.70 V or above	3 (6) × t_{IICcyc} + 300	—	ns	
			1.68 to 1.95 V				
	SCL, SDA rise time	t_{Sr}	2.70 V or above	20	300	ns	
			1.68 to 1.95 V				
	SCL, SDA fall time	t_{Sf}	2.70 V or above	20 × (external pullup voltage/5.5 V)*1	300	ns	
			1.68 to 1.95 V				
	SCL, SDA input spike pulse removal time	t_{SP}	2.70 V or above	0	1 (4) × t_{IICcyc}	ns	
			1.68 to 1.95 V				
	SDA input bus free time when wakeup function is disabled	t_{BUF}	2.70 V or above	3 (6) × t_{IICcyc} + 300	—	ns	
			1.68 to 1.95 V				
	SDA input bus free time when wakeup function is enabled	t_{BUF}	2.70 V or above	3 (6) × t_{IICcyc} + 4 × t_{Pcyc} + 300	—	ns	
			1.68 to 1.95 V				
	START condition input hold time when wakeup function is disabled	t_{STAH}	2.70 V or above	t_{IICcyc} + 300	—	ns	
			1.68 to 1.95 V				
	START condition input hold time when wakeup function is enabled	t_{STAH}	2.70 V or above	1 (5) × t_{IICcyc} + t_{Pcyc} + 300	—	ns	
			1.68 to 1.95 V				
	Repeated START condition input setup time	t_{STAS}	2.70 V or above	300	—	ns	
			1.68 to 1.95 V				
	STOP condition input setup time	t_{STOS}	2.70 V or above	300	—	ns	
			1.68 to 1.95 V				
	Data input setup time	t_{SDAS}	2.70 V or above	t_{IICcyc} + 50	—	ns	
			1.68 to 1.95 V				
	Data input hold time	t_{SDAH}	2.70 V or above	0	—	ns	
			1.68 to 1.95 V				
	SCL, SDA capacitive load	C_b	2.70 V or above	—	400	pF	
			1.68 to 1.95 V				

Note: t_{IICcyc}: IIC internal reference clock (IICφ) cycle, t_{Pcyc}: PCLKB cycle.
Note: Values in parentheses apply when ICMR3.NF[1:0] is set to 11b while the digital filter is enabled with ICFER.NFE set to 1.
Note: Must use pins that have a letter appended to their name, for instance "_A", "_B", to indicate group membership. For the IIC interface, the AC portion of the electrical characteristics is measured for each group.
Note 1. Only supported for SCL0_A, SDA0_A, SCL1_A, and SDA1_A.

Table 60.47 IIC timing (2)

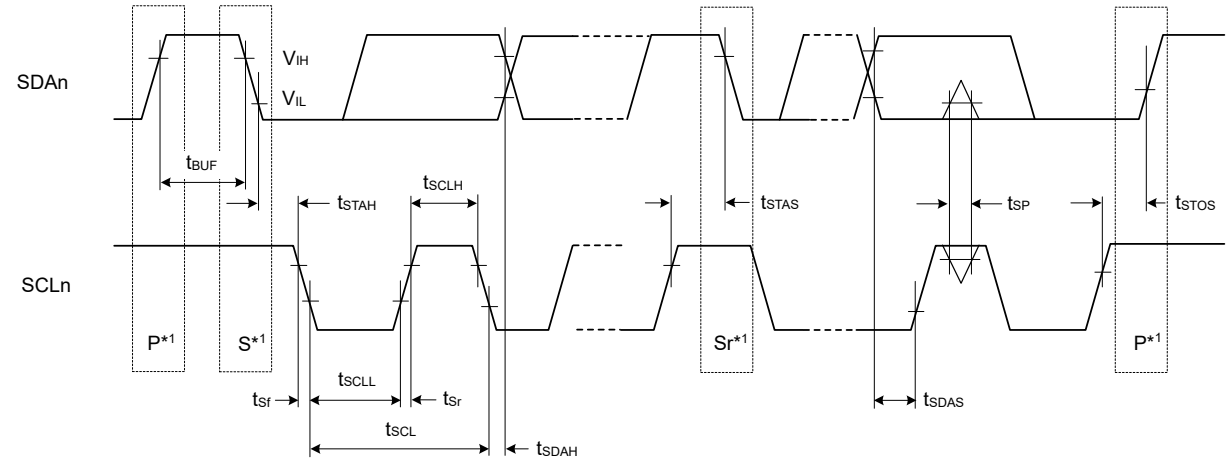
Setting of the SCL0_A, SDA0_A, SCL1_A, SDA1_A pins are not required with the port drive capability bit in the PmnPFS register.

Parameter		Symbol	VCC	Min	Max	Unit	Test conditions
IIC (Fast-mode+) ICFER.FMPE = 1	SCL input cycle time	t_{SCL}	2.70 V or above	6 (12) × t_{IICcyc} + 240	—	ns	Figure 60.71
	SCL input high pulse width	t_{SCLH}	2.70 V or above	3 (6) × t_{IICcyc} + 120	—	ns	
	SCL input low pulse width	t_{SCLL}	2.70 V or above	3 (6) × t_{IICcyc} + 120	—	ns	
	SCL, SDA rise time	t_{Sr}	2.70 V or above	—	120	ns	
	SCL, SDA fall time	t_{Sf}	2.70 V or above	20 × (external pullup voltage/ 5.5V)	120	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	2.70 V or above	0	1 (4) × t_{IICcyc}	ns	
	SDA input bus free time when wakeup function is disabled	t_{BUF}	2.70 V or above	3 (6) × t_{IICcyc} + 120	—	ns	
	SDA input bus free time when wakeup function is enabled	t_{BUF}	2.70 V or above	3 (6) × t_{IICcyc} + 4 × t_{Pcyc} + 120	—	ns	
	Start condition input hold time when wakeup function is disabled	t_{STAH}	2.70 V or above	t_{IICcyc} + 120	—	ns	
	START condition input hold time when wakeup function is enabled	t_{STAH}	2.70 V or above	1 (5) × t_{IICcyc} + t_{Pcyc} + 120	—	ns	
	Restart condition input setup time	t_{STAS}	2.70 V or above	120	—	ns	
	Stop condition input setup time	t_{STOS}	2.70 V or above	120	—	ns	
	Data input setup time	t_{SDAS}	2.70 V or above	t_{IICcyc} + 30	—	ns	
	Data input hold time	t_{SDAH}	2.70 V or above	0	—	ns	
	SCL, SDA capacitive load	C_b*1	2.70 V or above	—	550	pF	

Note: t_{IICcyc}: IIC internal reference clock (IICφ) cycle, tPcyc: PCLKB cycle.

Note: Values in parentheses apply when ICMR3.NF[1:0] is set to 11b while the digital filter is enabled with ICFER.NFE set to 1.

Note 1. Cb indicates the total capacity of the bus line.



Note 1. S, P, and Sr indicate the following conditions:
S: Start condition
P: Stop condition
Sr: Restart condition

Figure 60.71 I^2C bus interface input/output timing

60.3.13 I3C Timing

Table 60.48 IIC timing(1)-1

Setting of the I3C_SCL0, I3C_SDA0 pins are not required with the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	VCC	Min	Max	Unit
IIC (Standard mode, SMBus) BFCTL.FMPE = 0	SCL input cycle time	t_{SCL}	2.70 V or above, 1.68 to 1.95 V	10 (18) × t_{I3Ccyc} + 1300	—	ns
	SCL input high pulse width	t_{SCLH}	2.70 V or above, 1.68 to 1.95 V	5 (9) × t_{I3Ccyc} + 300	—	ns
	SCL input low pulse width	t_{SCLL}	2.70 V or above, 1.68 to 1.95 V	5 (9) × t_{I3Ccyc} + 300	—	ns
	SCL, SDA rise time	t_{Sr}	2.70 V or above, 1.68 to 1.95 V	—	1000	ns
	SCL, SDA fall time	t_{Sf}	2.70 V or above, 1.68 to 1.95 V	—	300	ns
	SCL, SDA input spike pulse removal time	t_{SP}	2.70 V or above, 1.68 to 1.95 V	0	1 (4) × t_{I3Ccyc}	ns
	SDA input bus free time when wakeup function is disabled	t_{BUF}	2.70 V or above, 1.68 to 1.95 V	5(9) × t_{I3Ccyc} + 300	—	ns
	SDA input bus free time when wakeup function is enabled	t_{BUF}	2.70 V or above, 1.68 to 1.95 V	5(9) × t_{I3Ccyc} + 4 × t_{Tcyc} + 300	—	ns
	START condition input hold time when wakeup function is disabled	t_{STAH}	2.70 V or above, 1.68 to 1.95 V	t_{I3Ccyc} + 300	—	ns
	START condition input hold time when wakeup function is enabled	t_{STAH}	2.70 V or above, 1.68 to 1.95 V	1(5) × t_{I3Ccyc} + t_{Tcyc} + 300	—	ns
	Repeated START condition input setup time	t_{STAS}	2.70 V or above, 1.68 to 1.95 V	1000	—	ns
	STOP condition input setup time	t_{STOS}	2.70 V or above, 1.68 to 1.95 V	1000	—	ns
	Data input setup time	t_{SDAS}	2.70 V or above, 1.68 to 1.95 V	t_{I3Ccyc} + 50	—	ns
	Data input hold time	t_{SDAH}	2.70 V or above, 1.68 to 1.95 V	0	—	ns
	SCL, SDA capacitive load	C_b*1	2.70 V or above, 1.68 to 1.95 V	—	400	pF

Note: t_{I3Ccyc}: I3C internal reference clock (I3Cφ) cycle, t_{Tcyc}: TCLK cycle.

Note: Values in parentheses apply when INCTL.DNFS[3:0] is set to 0011b while the digital filter is enabled with INCTL.DNFE set to 1.

Note 1. C_b indicates the total capacity of the bus line.

Table 60.49 IIC timing(1)-2

Setting of the I3C_SCL0, I3C_SDA0 pins are not required with the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	VCC	Min	Max	Unit
IIC (Fast-mode)	SCL input cycle time	t_{SCL}	2.70 V or above, 1.68 to 1.95 V	10 (18) × t_{I3Ccyc} + 600	—	ns
	SCL input high pulse width	t_{SCLH}	2.70 V or above, 1.68 to 1.95 V	5 (9) × t_{I3Ccyc} + 300	—	ns
	SCL input low pulse width	t_{SCLL}	2.70 V or above, 1.68 to 1.95 V	5 (9) × t_{I3Ccyc} + 300	—	ns
	SCL, SDA rise time	t_{Sr}	2.70 V or above, 1.68 to 1.95 V	20	300	ns
	SCL, SDA fall time	t_{Sf}	2.70 V or above, 1.68 to 1.95 V	20 × (external pull-up voltage/3.6 V)	300	ns
	SCL, SDA input spike pulse removal time	t_{SP}	2.70 V or above, 1.68 to 1.95 V	0	1 (4) × t_{I3Ccyc}	ns
	SDA input bus free time when wakeup function is disabled	t_{BUF}	2.70 V or above, 1.68 to 1.95 V	5 (9) × t_{I3Ccyc} + 300	—	ns
	SDA input bus free time when wakeup function is enabled		2.70 V or above, 1.68 to 1.95 V	5(9) × t_{I3Ccyc} + 4 × t_{Tcyc} + 300	—	ns
	START condition input hold time when wakeup function is disabled	t_{STAH}	2.70 V or above, 1.68 to 1.95 V	t_{I3Ccyc} + 300	—	ns
	START condition input hold time when wakeup function is enabled		2.70 V or above, 1.68 to 1.95 V	1(5) × t_{I3Ccyc} + t_{Tcyc} + 300	—	ns
	Repeated START condition input setup time	t_{STAS}	2.70 V or above, 1.68 to 1.95 V	300	—	ns
	STOP condition input setup time	t_{STOS}	2.70 V or above, 1.68 to 1.95 V	300	—	ns
	Data input setup time	t_{SDAS}	2.70 V or above, 1.68 to 1.95 V	t_{I3Ccyc} + 50	—	ns
	Data input hold time	t_{SDAH}	2.70 V or above, 1.68 to 1.95 V	0	—	ns
	SCL, SDA capacitive load	C_b*1	2.70 V or above, 1.68 to 1.95 V	—	400	pF

Note: t_{I3Ccyc}: I3C internal reference clock (I3Cφ) cycle, t_{Tcyc}: TCLK cycle.
Note: Values in parentheses apply when INCTL.DNFS[3:0] is set to 0011b while the digital filter is enabled with INCTL.DNFE set to 1.
Note 1. C_b indicates the total capacity of the bus line.

Table 60.50 IIC timing(1)-3

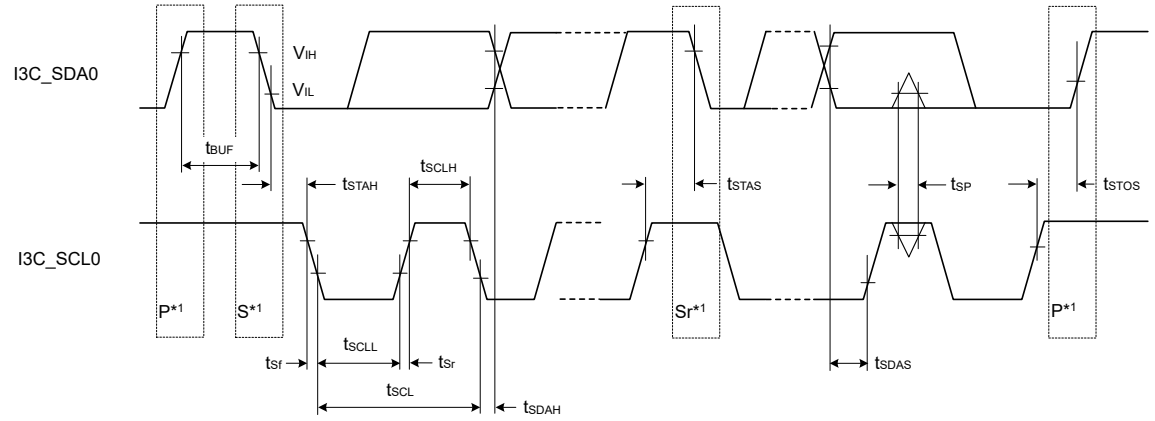
Setting of the I3C_SCL0, I3C_SDA0 pins are not required with the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	VCC	Min	Max	Unit
IIC (Fast-mode +) BFCTL.FMPE = 1	SCL input cycle time	t_{SCL}	2.70 V or above	10 (18) × t_{I3Ccyc} + 240	—	ns
	SCL input high pulse width	t_{SCLH}	2.70 V or above	5 (9) × t_{I3Ccyc} + 120	—	ns
	SCL input low pulse width	t_{SCLL}	2.70 V or above	5 (9) × t_{I3Ccyc} + 120	—	ns
	SCL, SDA rise time	t_{Sr}	2.70 V or above	—	120	ns
	SCL, SDA fall time	t_{Sf}	2.70 V or above	20 × (external pull-up voltage/3.3V)	120	ns
	SCL, SDA input spike pulse removal time	t_{SP}	2.70 V or above	0	1 (4) × t_{I3Ccyc}	ns
	SDA input bus free time when wakeup function is disabled	t_{BUF}	2.70 V or above	5 (9) × t_{I3Ccyc} + 120	—	ns
	SDA input bus free time when wakeup function is enabled			5(9) × t_{I3Ccyc} + 4 × t_{Tcyc} + 120		ns
	START condition input hold time when wakeup function is disabled	t_{STAH}	2.70 V or above	t_{I3Ccyc} + 120	—	ns
	START condition input hold time when wakeup function is enabled			1(5) × t_{I3Ccyc} + t_{Tcyc} + 120		ns
	Restart condition input setup time	t_{STAS}	2.70 V or above	120	—	ns
	Stop condition input setup time	t_{STOS}	2.70 V or above	120	—	ns
	Data input setup time	t_{SDAS}	2.70 V or above	t_{I3Ccyc} + 30	—	ns
	Data input hold time	t_{SDAH}	2.70 V or above	0	—	ns
	SCL, SDA capacitive load	C_b *1	2.70 V or above	—	550	pF

Note: t_{I3Ccyc}: I3C internal reference clock (I3Cφ) cycle. t_{Tcyc}: TCLK cycle.

Note: Values in parentheses apply when INCTL.DNFS[3:0] is set to 0011b while the digital filter is enabled with INCTL.DNFE set to 1.

Note 1. C_b indicates the total capacity of the bus line.



Test conditions:
V_{IH} = VCC × 0.7, V_{IL} = VCC × 0.3
V_{OL} = 0.6 V, I_{OL} = 6 mA (BFCTL.FMPE = 0)
V_{OL} = 0.4 V, I_{OL} = 15 mA (BFCTL.FMPE = 1)

Note 1. S, P, and Sr indicate the following conditions:
S: Start condition
P: Stop condition
Sr: Restart condition

Figure 60.72 I^2C bus interface input/output timing

Table 60.51 IIC timing(2)

Setting of the I3C_SCL0, I3C_SDA0 pins are not required with the Port Drive Capability bit in the PmnPFS register.

Parameter			Symbol	VCC	Min	Max	Unit
IIC (Hs-mode) BFCTL.HSME = 1	SCL input cycle time		t_{SCL}	3.00 V or above	47 (49) × t_{I3Ccyc}	—	ns
				1.68 to 1.95 V	48 (50) × t_{I3Ccyc}	—	
	SCL input high pulse width	Cb = 400 pF	t_{SCLH}	3.00 V or above	36 (37) × t_{I3Ccyc}	—	ns
				1.68 to 1.95 V	31 (32) × t_{I3Ccyc}	—	
		Cb = 100 pF		3.00 V or above	18 (19) × t_{I3Ccyc}	—	
				1.68 to 1.95 V	19 (20) × t_{I3Ccyc}	—	
	SCL input low pulse width	Cb = 400 pF	t_{SCLL}	3.00 V or above	61 (62) × t_{I3Ccyc}	—	ns
				1.68 to 1.95 V	61 (62) × t_{I3Ccyc}	—	
		Cb = 100 pF		3.00 V or above	29 (30) × t_{I3Ccyc}	—	
				1.68 to 1.95 V	29 (30) × t_{I3Ccyc}	—	
	SCL rise time	Cb = 400 pF	t_{SrCL}	3.00 V or above	—	80	ns
				1.68 to 1.95 V	—	80	
		Cb = 100 pF		3.00 V or above	—	40	
				1.68 to 1.95 V	—	40	
	SDA rise time	Cb = 400 pF	t_{SrDA}	3.00 V or above	—	160	ns
				1.68 to 1.95 V	—	160	
		Cb = 100 pF		3.00 V or above	—	80	
				1.68 to 1.95 V	—	80	
	SCL fall time	Cb = 400 pF	t_{SfCL}	3.00 V or above	—	80	ns
				1.68 to 1.95 V	—	80	
		Cb = 100 pF		3.00 V or above	—	40	
				1.68 to 1.95 V	—	40	
	SDA fall time	Cb = 400 pF	t_{SfDA}	3.00 V or above	—	160	ns
				1.68 to 1.95 V	—	160	
		Cb = 100 pF		3.00 V or above	—	80	
				1.68 to 1.95 V	—	80	
	SCL, SDA input spike pulse removal time		t_{SP}	3.00 V or above	0	1 (1) × t_{I3Ccyc}	ns
				1.68 to 1.95 V	0	1 (1) × t_{I3Ccyc}	
	Repeated START condition input setup time		t_{STAS}	3.00 V or above	40	—	ns
				1.68 to 1.95 V	40	—	
	STOP condition input setup time		t_{STOS}	3.00 V or above	40	—	ns
				1.68 to 1.95 V	40	—	
	Data input setup time		t_{SDAS}	3.00 V or above	10	—	ns
				1.65 to 1.95 V	10	—	
	Data input hold time	Cb = 400 pF	t_{SDAH}	3.00 V or above	0	150	ns
				1.68 to 1.95 V	0	150	
		Cb = 100 pF		3.00 V or above	0	70	
				1.68 to 1.95 V	0	70	
	SCL, SDA capacitive load		C_b*1	3.00 V or above	—	400	pF
				1.68 to 1.95 V	—	400	

Note: t_{I3Ccyc}: I3C internal reference clock (I3Cφ) cycle.
Note: Values in parentheses apply when INCTL.DNFS[3:0] is set to 0011b while the digital filter is enabled with INCTL.DNFE set to 1.
Note 1. C_b indicates the total capacity of the bus line.

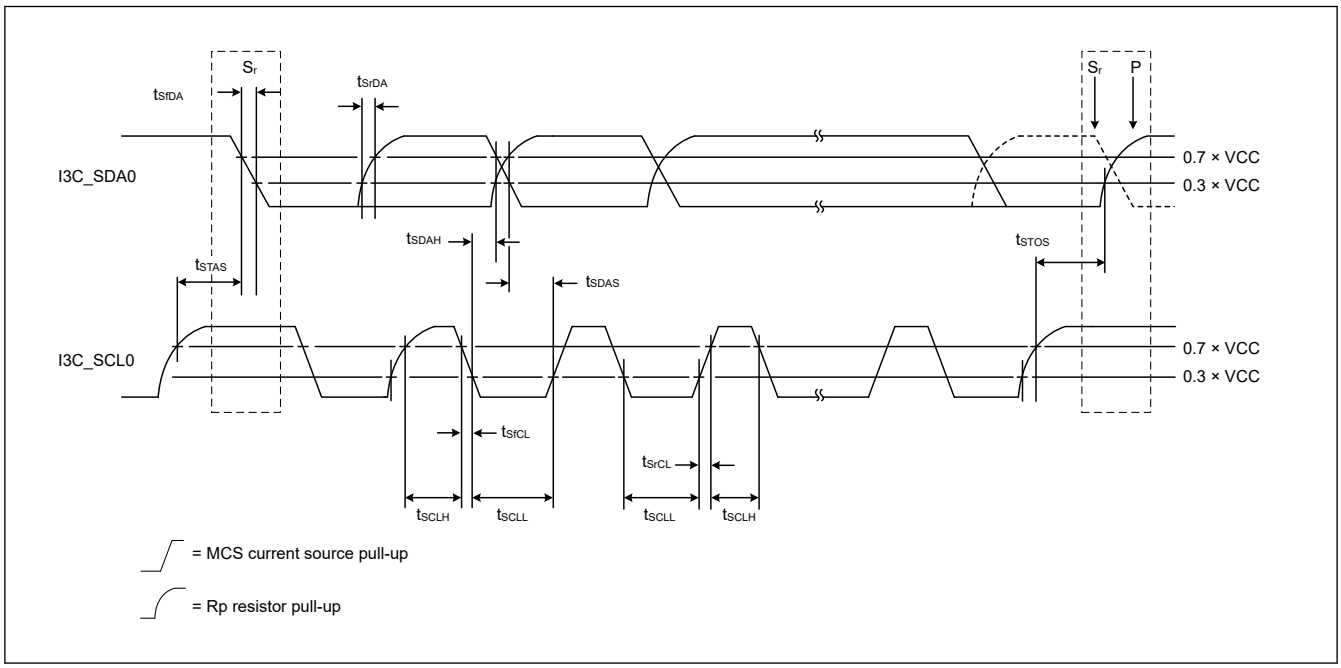


Figure 60.73 I²C bus interface input/output timing (Hs-mode)

Table 60.52 I3C timing (Open Drain Timing Parameters)

Setting of the I3C_SCL0, I3C_SDA0 pins are not required with the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	VCC	Min	Max	Unit	Test conditions
I3C Open Drain Timing Parameters	SCL Clock Low Period	t_{LOW_OD} *1 *2	3.00 V or above	200	—	ns	Figure 60.76
			1.68 to 1.95 V	200	—		
		$t_{DIG\_OD\_L}$	3.00 V or above	t_{LOW_ODmin} + t_{fDA_ODmin}	—	ns	Figure 60.76
			1.68 to 1.95 V	t_{LOW_ODmin} + t_{fDA_ODmin}	—		
	SCL Clock High Period	t_{HIGH} *3 *4	3.00 V or above	—	41	ns	Figure 60.74
			1.68 to 1.95 V	—	41		
		t_{DIG_H}	3.00 V or above	—	t_{HIGH} + t_{CF}	ns	Figure 60.74
			1.68 to 1.95 V	—	t_{HIGH} + t_{CF}		
	SDA Signal Fall Time	t_{fDA_OD}	3.00 V or above	t_{CF}	12	ns	Figure 60.76
			1.68 to 1.95 V	t_{CF}	12		
	SDA Data Setup Time Open Drain Mode	t_{SU_OD} *1	3.00 V or above	12	—	ns	Figure 60.75
			1.68 to 1.95 V	18	—		
	Clock After START (S) Condition	t_{CAS} *5 *6	3.00 V or above	38.4 nano	For ENAS0: 1 µ	seconds	Figure 60.76
					For ENAS1: 100 µ		
					For ENAS2: 2 milli		
					For ENAS3: 50 milli		
			1.68 to 1.95 V	38.4 nano	For ENAS0: 1 µ		
					For ENAS1: 100 µ		
					For ENAS2: 2 milli		
					For ENAS3: 50 milli		
	Clock Before STOP (P) Condition	t_{CBP}	3.00 V or above	t_{CASmin} / 2	—	seconds	Figure 60.77
			1.68 to 1.95 V	t_{CASmin} / 2	—		
	Current Master to Secondary Master Overlap time during handoff	$t_{MMOverlap}$	3.00 V or above	$t_{DIG\_OD\_Lmin}$	—	ns	Figure 60.83
			1.68 to 1.95 V	$t_{DIG\_OD\_Lmin}$	—		
	Bus Available Condition	t_{AVAL} *7	3.00 V or above	1	—	µs	—
			1.68 to 1.95 V	1	—		
	Bus Idle Condition	t_{IDLE}	3.00 V or above	1	—	ms	—
			1.68 to 1.95 V	1	—		
	Time Internal Where New Master Not Driving SDA Low	t_{MMLock}	3.00 V or above	$t_{AVALmin}$	—	µs	Figure 60.83
			1.68 to 1.95 V	$t_{AVALmin}$			

Note 1. This is approximately equal to t_{LOWmin} + t_{DS_ODmin} + t_{rDA_ODtyp} + t_{SU_Odmin}.

Note 2. The Master may use a shorter Low period if it knows that this is safe, i.e., that SDA is already above VIH

Note 3. Based on t_{SPIKE}, rise and fall times, and interconnect

Note 4. This maximum High period may be exceeded when the signals can be safely seen by Legacy I^2C Devices, and/or in consideration of the interconnect (e.g., a short Bus).
As a product specification, if this Max value cannot be guaranteed, change this Max value and specify that it cannot be used in the Mixed Bus.

Note 5. On a Legacy Bus where I^2C Devices need to see Start
Note 6. Slaves that do not support the optional ENTASx CCCs shall use the t_{CAS} Max value shown for ENTAS3
Note 7. On a Mixed Bus with Fm Legacy I^2C Devices, t_{AVAL} is 300 ns shorter than the Fm Bus Free Condition time (t_{BUF})

Table 60.53 I3C timing (Push-Pull Timing Parameters for SDR and HDR-DDR Modes)

Setting of the I3C_SCL0, I3C_SDA0 pins are not required with the Port Drive Capability bit in the PmnPFS register.

Parameter			Symbol	VCC	Min	Max	Unit	Test conditions
I3C Push-Pull Timing Parameters for SDR and HDR-DDR Modes	SCL Clock Frequency		f_{SCL} *1	3.00 V or above	0.01	12.5	MHz	—
				1.68 to 1.95 V	0.01	12.5		
	SCL Clock Low Period		t_{LOW}	3.00 V or above	27	—	ns	Figure 60.74
				1.68 to 1.95 V	32	—		
			t_{DIG_L} *2 *4	3.00 V or above	35	—	ns	Figure 60.74
				1.68 to 1.95 V	40	—		
	SCL Clock High Period for Mixed Bus		t_{HIGH_MIXED}	3.00 V or above	24	—	ns	Figure 60.74
				1.68 to 1.95 V	27	—		
			$t_{DIG\_H\_MIXED}$ *2 *3	3.00 V or above	32	45	ns	Figure 60.74
				1.68 to 1.95 V	35	45		
	SCL Clock High Period		t_{HIGH}	3.00 V or above	24	—	ns	Figure 60.74
				1.68 to 1.95 V	27	—		
			t_{DIG_H} *2	3.00 V or above	32	—	ns	Figure 60.74
				1.68 to 1.95 V	35	—		
	Clock in to Data Out for Slave		t_{SCO}	3.00 V or above	—	12	ns	Figure 60.79
				1.68 to 1.95 V	—	12		
	SCL Clock Rise Time		t_{CR}	3.00 V or above	—	150 × 1 / f_{SCL} (capped at 60)	ns	Figure 60.74
				1.68 to 1.95 V	—	150 × 1 / f_{SCL} (capped at 60)		
	SCL Clock Fall Time		t_{CF}	3.00 V or above	—	150 × 1 / f_{SCL} (capped at 60)	ns	Figure 60.74
				1.68 to 1.95 V	—	150 × 1 / f_{SCL} (capped at 60)		
	SDA Signal Data Hold in Push-Pull Mode	Master	t_{HD_PP} *4*5	3.00 V or above	t_{CR} + 3 and t_{CF} + 3	—	—	Figure 60.78
				1.68 to 1.95 V	t_{CR} + 3 and t_{CF} + 3	—		
		Slave	t_{HD_PP} *5	3.00 V or above	0	—	—	Figure 60.78
				1.68 to 1.95 V	0	—		
	SDA Signal Data Setup in Push-Pull Mode		t_{SU_PP}	3.00 V or above	12	N/A	ns	Figure 60.80
				1.68 to 1.95 V	18	N/A		
	Clock After Repeated START (Sr)		t_{CASr}	3.00 V or above	t_{CASmin}	N/A	ns	Figure 60.82
				1.68 to 1.95 V	t_{CASmin}	N/A		
	Clock Before Repeated START (Sr)		t_{CBSr}	3.00 V or above	t_{CASmin} / 2	N/A	ns	Figure 60.82
				1.68 to 1.95 V	t_{CASmin} / 2	N/A		
	Capacitive Load per Bus Line (SDA/SCL)		C_b	3.00 V or above	—	50	pF	—
				1.68 to 1.95 V	—	50		

Note 1. $f_{SCL} = 1 / (t_{DIG\_L} + t_{DIG\_H})$
Note 2. t_{DIG_L} and t_{DIG_H} are the clock Low and High periods as seen at the receiver end of the I3C Bus using V_{IL} and V_{IH}.
Note 3. When communicating with an I3C Device on a mixed Bus, the $t_{DIG\_H\_MIXED}$ period must be constrained in order to make sure that I^2C Devices do not interpret I3C signaling as valid I^2C signaling.
Note 4. As both edges are used, the hold time needs to be satisfied for the respective edges; i.e., t_{CF} + 3 for falling edge clocks, and t_{CR} + 3 for rising edge clocks.
Note 5. In SDR Mode the Hold time parameter is referred to as t_{HD_SDR}, and in DDR Mode it is referred to as t_{HD_DDR}.

Table 60.54 I3C timing (Push-Pull Timing Parameters for HDR-TSP and HDR-TSL Modes)

Setting of the I3C_SCL0, I3C_SDA0 pins are not required with the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	VCC	Min	Max	Unit	Test conditions
I3C Push-Pull Timing Parameters for HDR-TSP and HDR-TSL Modes	Edge-to-Edge Period	t_{EDGE} *1 *2	3.00 V or above	t_{DIG_H}	—	ns	Figure 60.74
			1.68 to 1.95 V	t_{DIG_H}	—		
	Allowed Difference Between Signals for 'Simultaneous' Change	t_{SKEW}	3.00 V or above	—	10	ns	
			1.68 to 1.95 V	—	10		
	Stable Condition Between Symbols	t_{EYE}	3.00 V or above	12	—	ns	
			1.68 to 1.95 V	12	—		
	Time Between Successive Symbols	t_{SYMBOL}	3.00 V or above	$t_{EDGE\ Min}$	—	ns	
			1.68 to 1.95 V	$t_{EDGE\ Min}$	—		
	Symbol Clock	t_{CLOCK}	3.00 V or above	1 / f_{SCL} (Max)	—	—	
			1.68 to 1.95 V	1 / f_{SCL} (Max)	—		

Note 1. Edges occur at the rate of 1 / (t_{EDGE} × 2)
Note 2. In a Mixed Bus, HDR-TSL shall respect the maximum $t_{DIG\_H\_MIXED}$ shown in Figure 60.77.

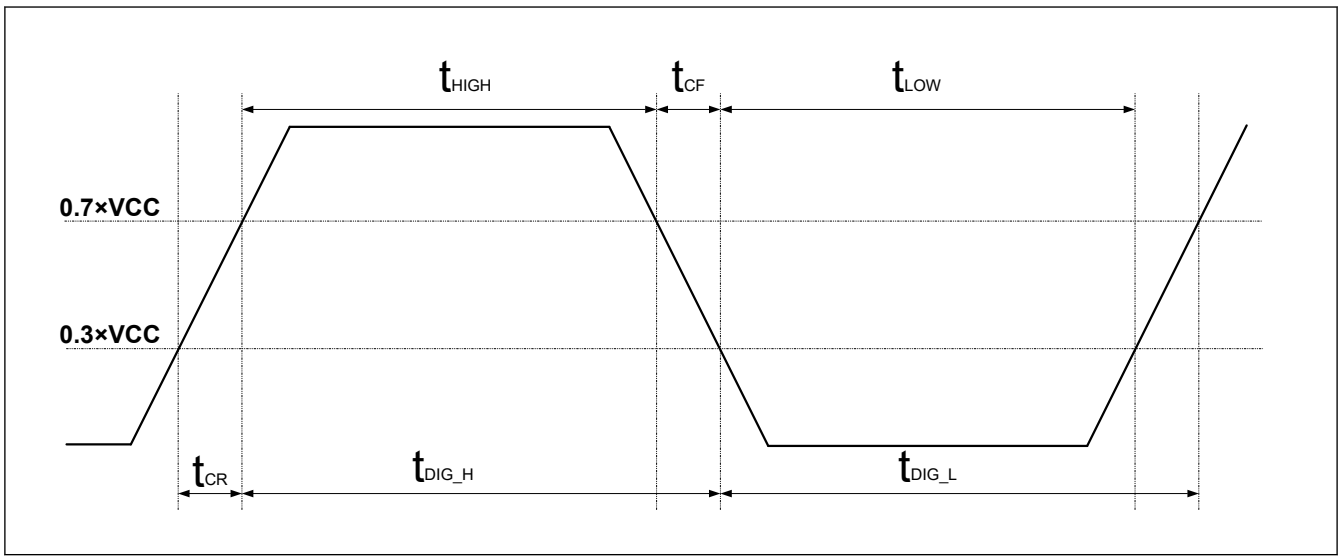


Figure 60.74 t_{DIG_H} and t_{DIG_L}

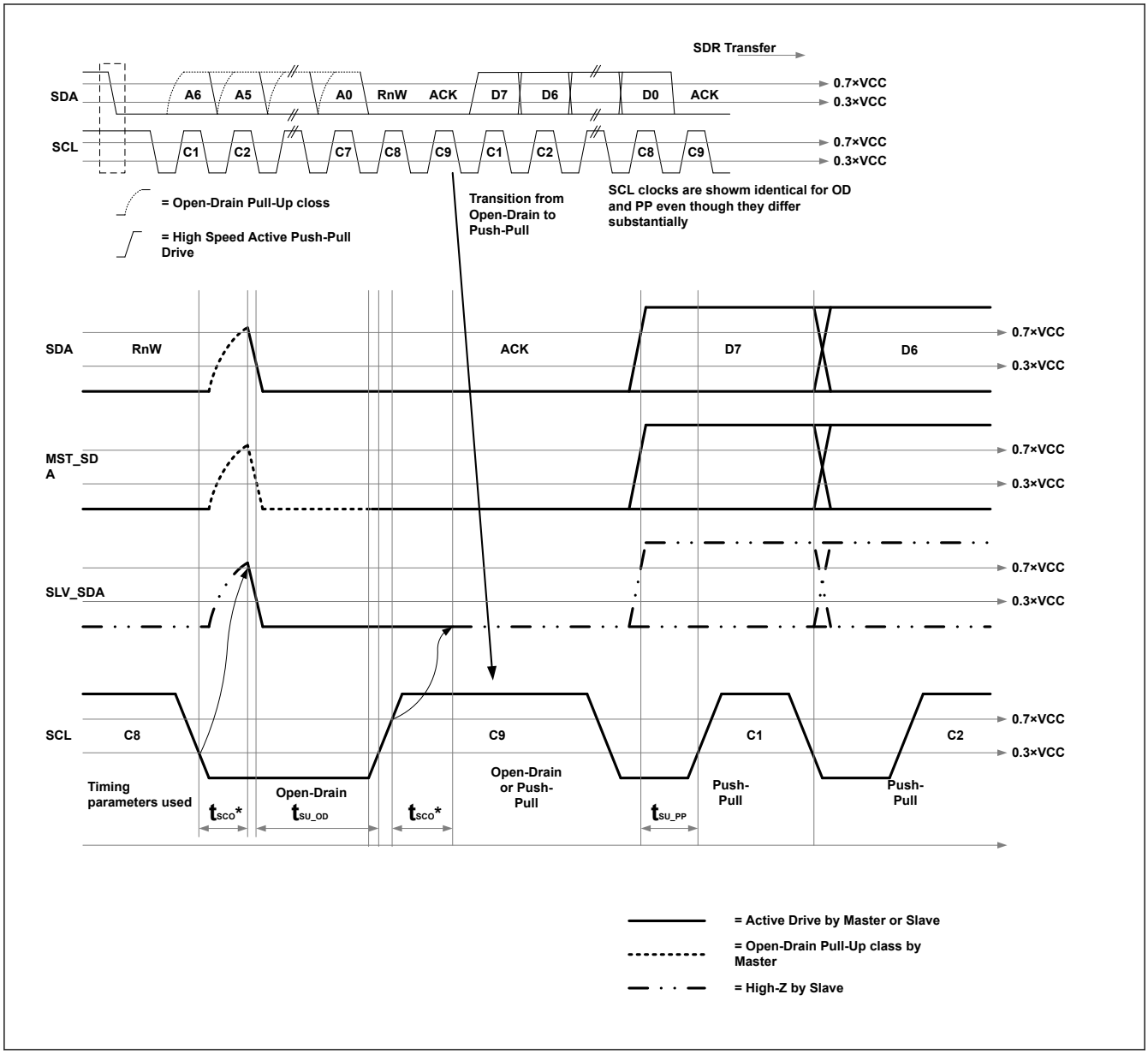


Figure 60.75 I3C Data Transfer – ACK by Slave

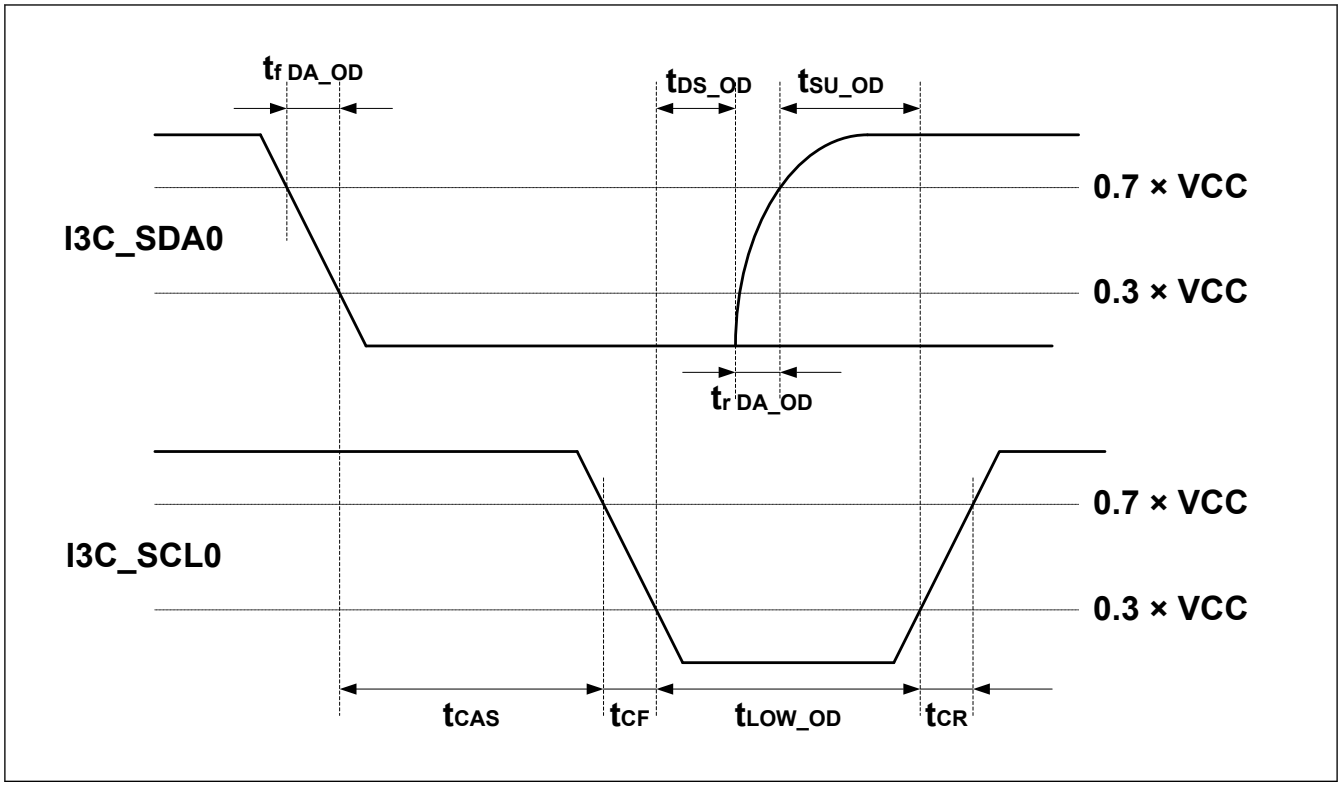


Figure 60.76 I3C START condition Timing

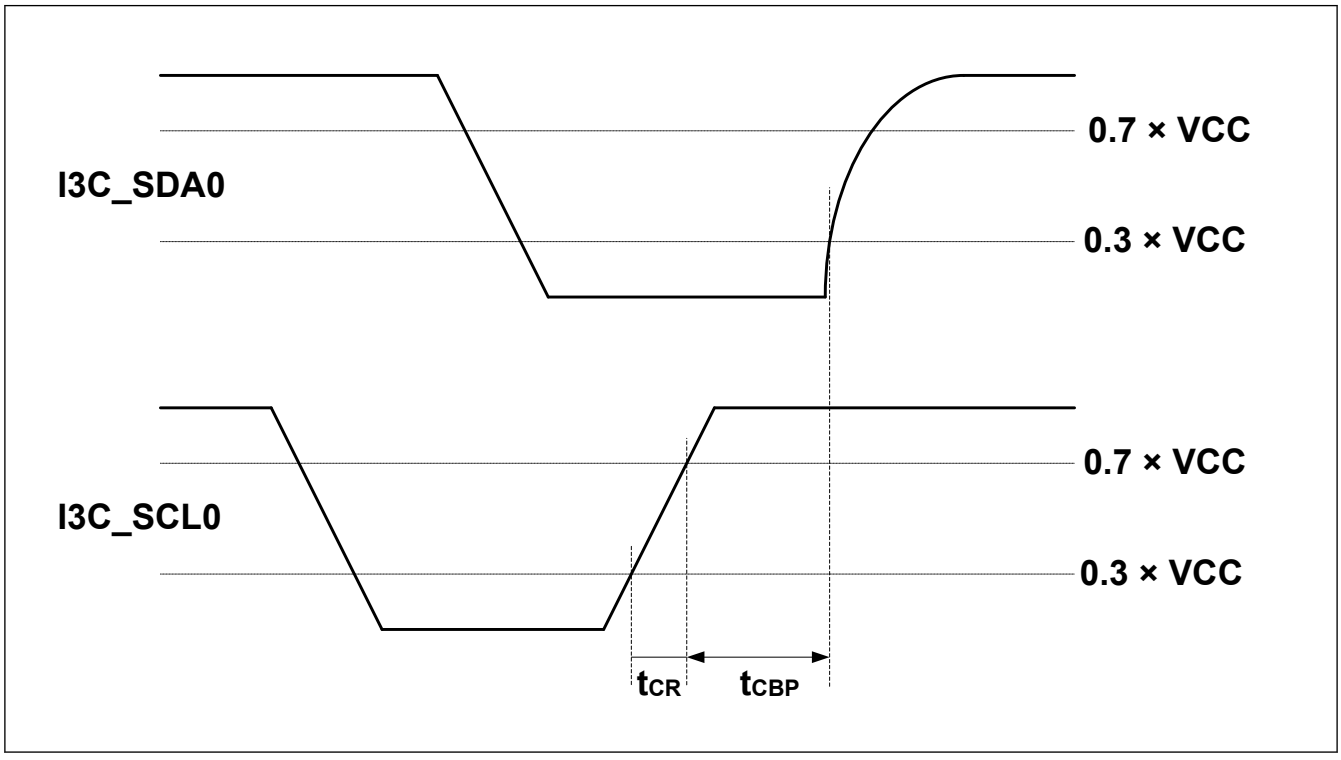


Figure 60.77 I3C STOP condition Timing

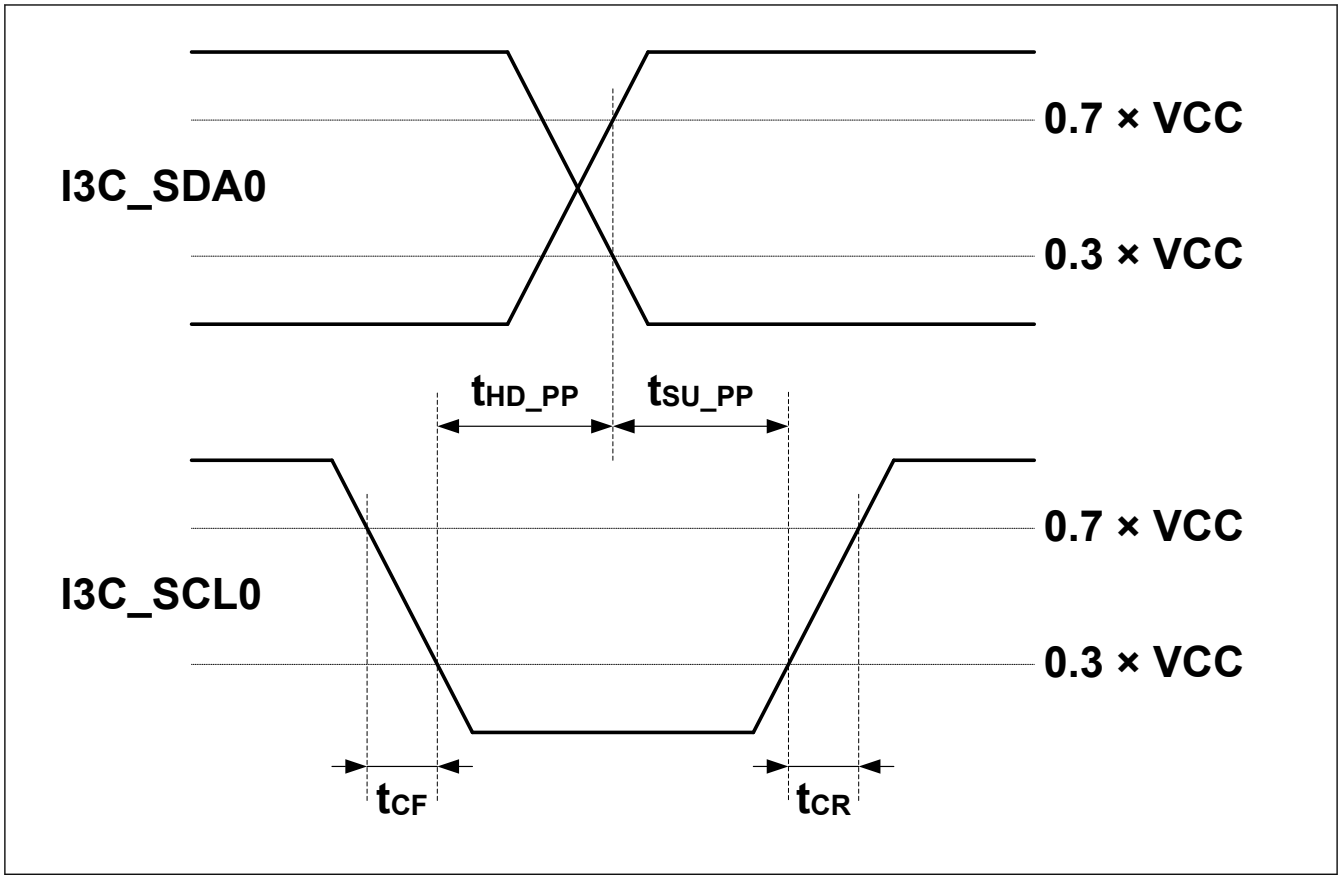


Figure 60.78 I3C Master Out Timing

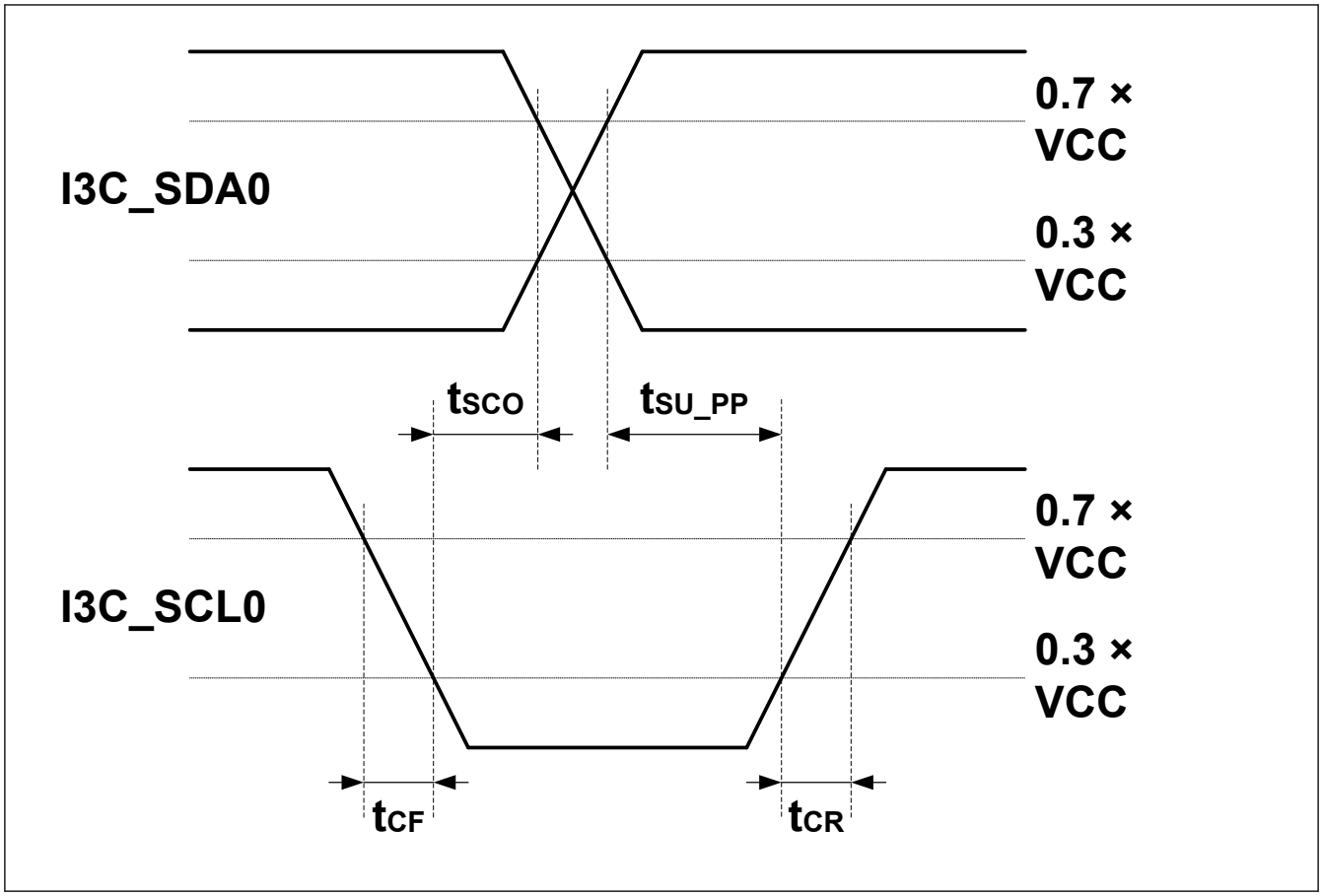


Figure 60.79 I3C Slave Out Timing

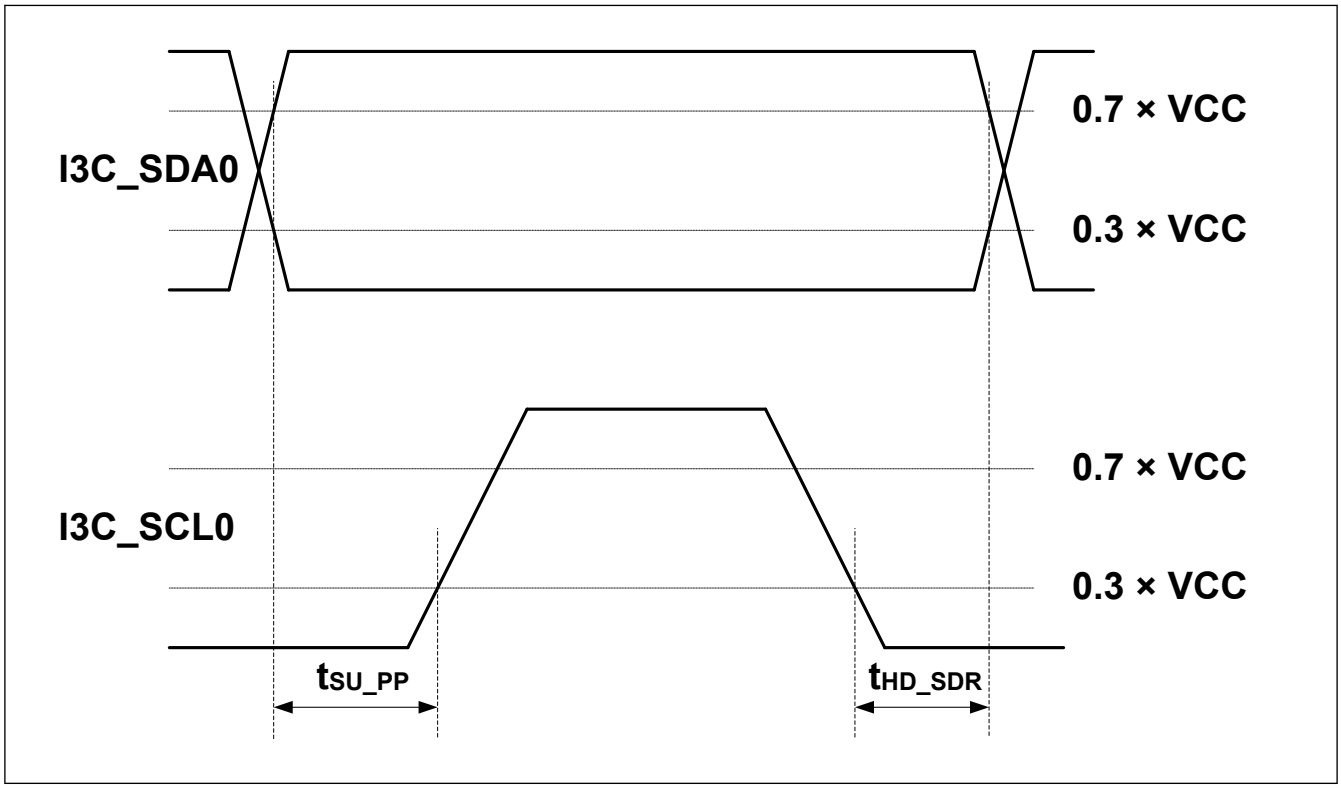


Figure 60.80 Master SDR Timing

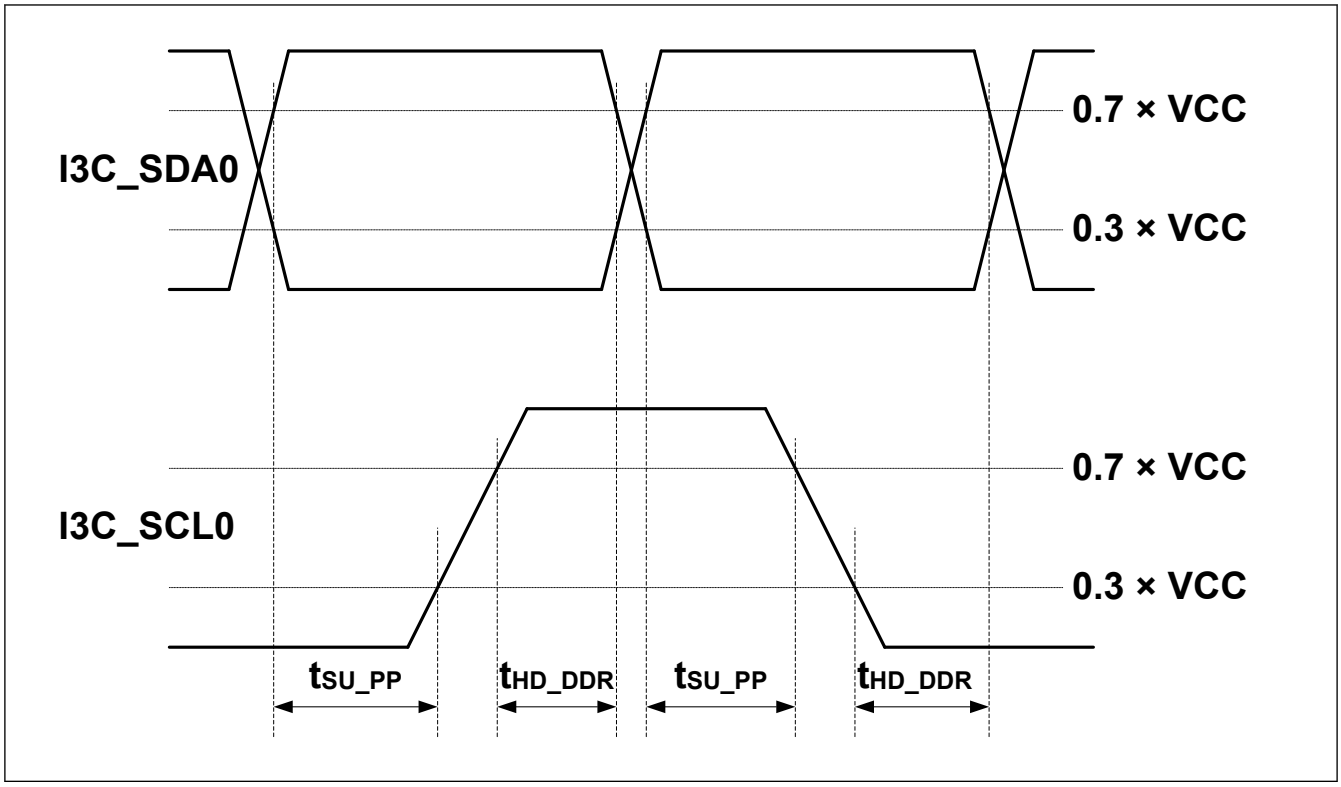


Figure 60.81 Master DDR Timing

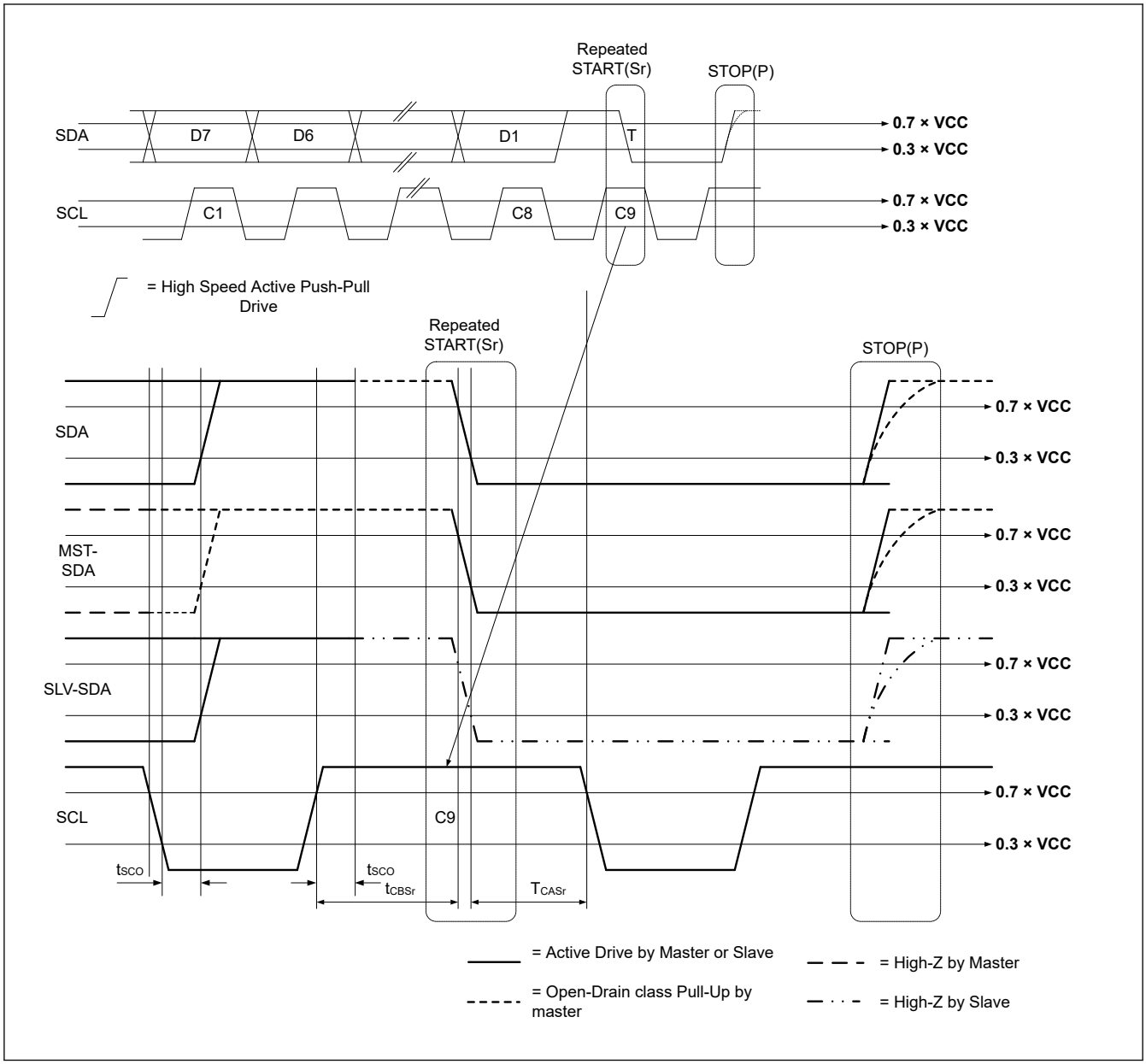


Figure 60.82 T-Bit When Master Ends Read with Repeated START and STOP

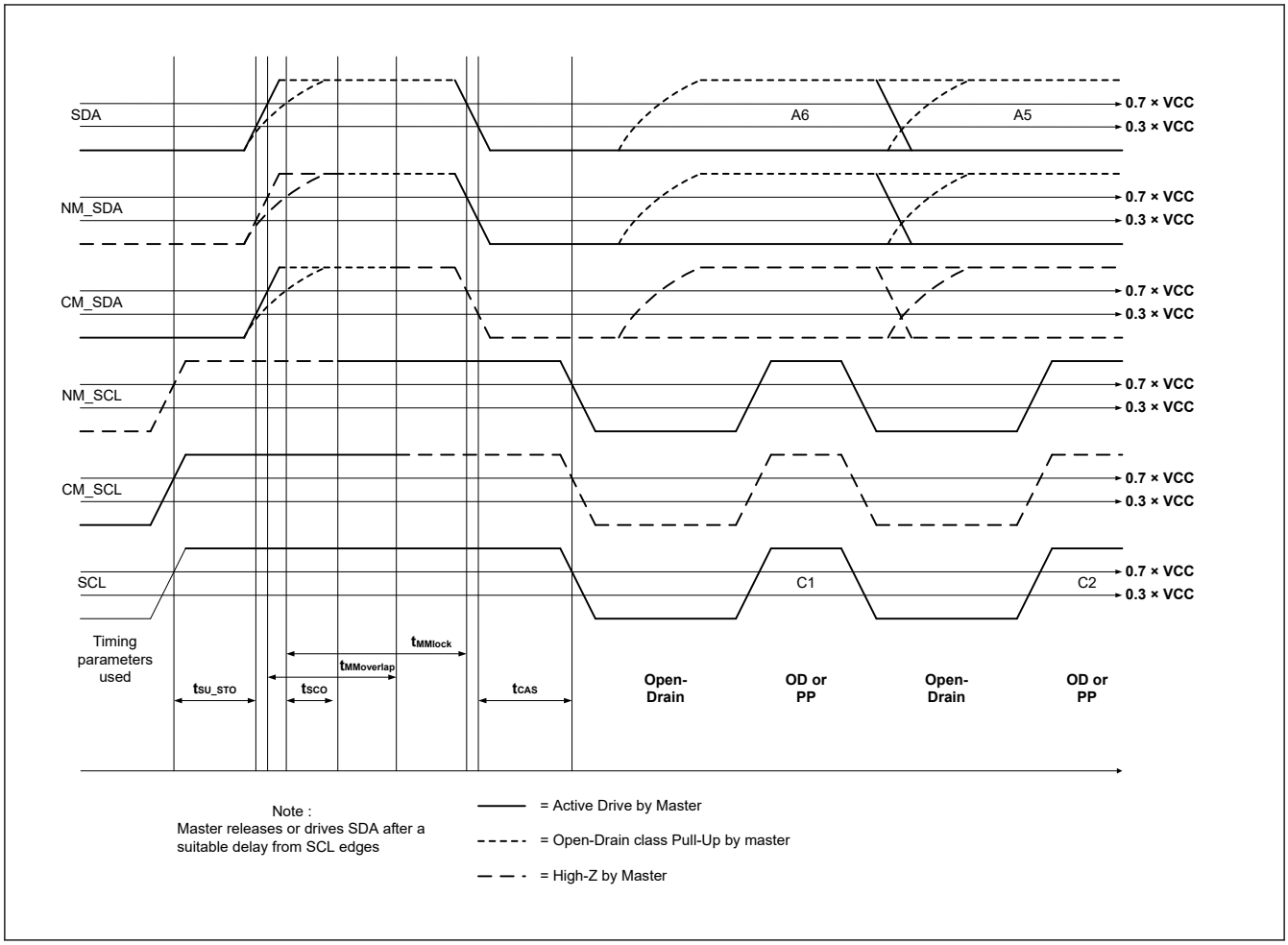
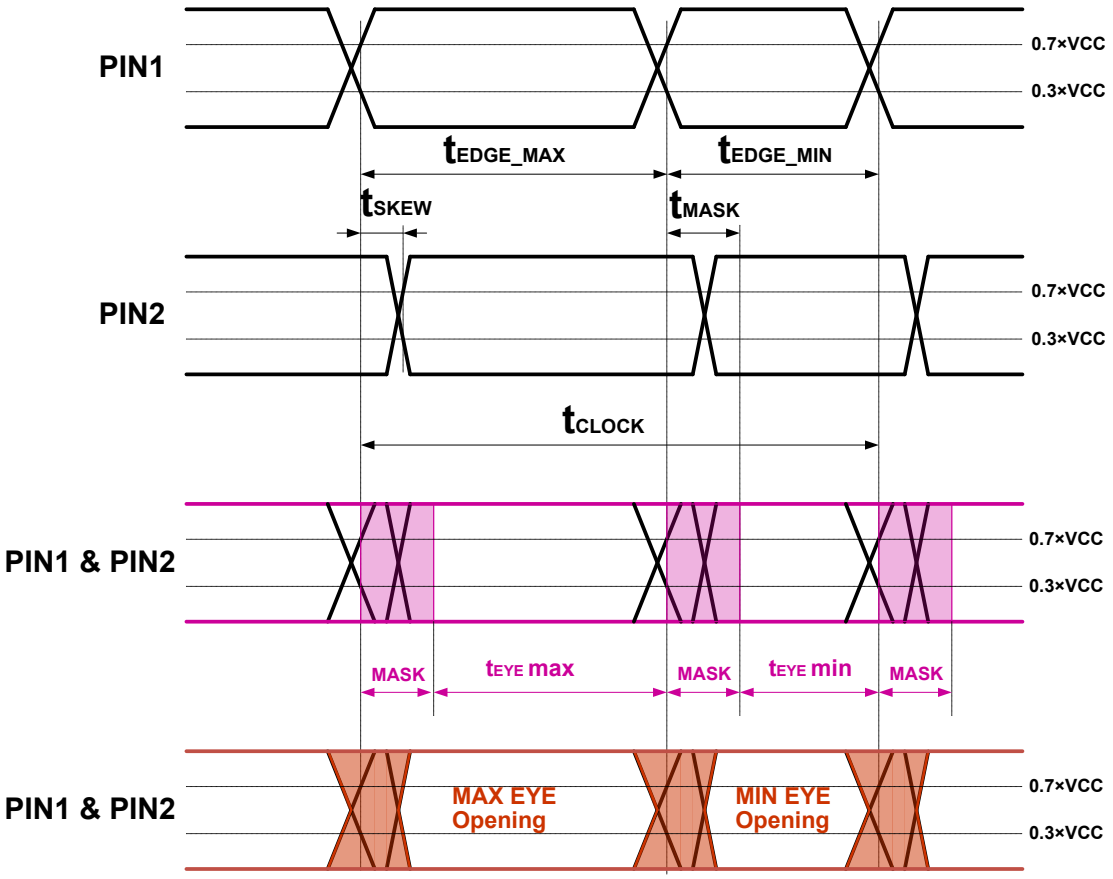


Figure 60.83 Master to Master Bus Handoff



Design Conditions

1. The The differences of load magnitude and drive strength between the SDA and SCL lines result in different slopes, as shown
2. MASK period ≧ MAX t_{SKEW} + MAX rise/fall time between 0.3 × V_{DD} and 0.7 × V_{DD}, respectively, decreased by the time crossing the hysteresis window.

Simplifying Assumptions

1. t_{SKEW} is the same for both edges
2. Base CLOCK (e.g. 12.5MHz) is asymmetrical
3. Falling edges are equal rising edges
4. The MASK is shown as starting, from the last possible moment, whereas in reality it could be shifted left (towards the start point) delayed by the hysteresis window

Figure 60.84 Ternary Protocol Timing

60.3.14 SSIE Timing

Table 60.55 SSIE timing (1 of 2)

(1) High drive output is selected with the Port Drive Capability bit in the PmnPFS register.
(2) Use pins that have a letter appended to their names, for instance "_A" , "_B" or "_C" to indicate group membership. For the SSIE interface, the AC portion of the electrical characteristics is measured for each group.

Parameter			Symbol	VCC	Min.	Max.	Unit	Comments
SSIBCK	Cycle	Master	t_O	2.70V or above	80	—	ns	Figure 60.85
				1.68V or above	80	—		
		Slave	t_I	2.70V or above	80	—	ns	
				1.68V or above	80	—		
	High level/ low level	Master	t_{HC}/t_{LC}	2.70V or above	0.35	—	t_O	
				1.68V or above	0.35	—		
		Slave		2.70V or above	0.35	—	t_I	
				1.68V or above	0.35	—		
	Rising time/ falling time	Master	t_{RC}/t_{FC}	2.70V or above	—	0.15	t_O / t_I	
				1.68V or above	—	0.15		
		Slave		2.70V or above	—	0.15	t_O / t_I	
				1.68V or above	—	0.15		

Table 60.55 SSIE timing (2 of 2)

(1) High drive output is selected with the Port Drive Capability bit in the PmnPFS register.
(2) Use pins that have a letter appended to their names, for instance "_A" , "_B" or "_C" to indicate group membership. For the SSIE interface, the AC portion of the electrical characteristics is measured for each group.

Parameter			Symbol	VCC	Min.	Max.	Unit	Comments
SSILRCK0/ SSIFS0, SSITXD0, SSIRXD0	Input set up time	Master	t_{SR}	2.70V or above	12	—	ns	Figure 60.87, Figure 60.88
				1.68V or above	20	—		
		Slave		2.70V or above	12	—	ns	
				1.68V or above	12	—		
	Input hold time	Master	t_{HR}	2.70V or above	8	—	ns	
				1.68V or above	8	—		
		Slave		2.70V or above	15	—	ns	
				1.68V or above	15	—		
	Output delay time	Master	t_{DTR}	2.70V or above	-10	5	ns	
				1.68V or above	-10	7		
		Slave		2.70V or above	0	20	ns	
				1.68V or above	0	25		
	Output delay time from SSILRCK/ SSIFS change	Slave	t_{DTRW}	2.70V or above	—	20	ns	Figure 60.89
				1.68V or above	—	25		
GTIOC2A, AUDIO_CLK	Cycle		t_{EXcyc}	2.70V or above	20	—	ns	Figure 60.86*1
				1.68V or above	40	—		
	High level/ low level		t_{EXL}/t_{EXH}	2.70V or above	0.4	—	t_{EXcyc}	
				1.68V or above	0.4	—		
	Rising time/ falling time		t_{EXr}/t_{EXr}	2.70V or above	—	0.1*2	t_{EXcyc}	
				1.68V or above	—	0.1*2		

Note 1. For slave-mode transmission, SSIE has a path, through which the signal input from the SSILRCK/SSIFS pin is used to generate transmit data, and the transmit data is logically output to the SSITXD0 or SSIDATA1 pin.
Note 2. 1µs at the longest

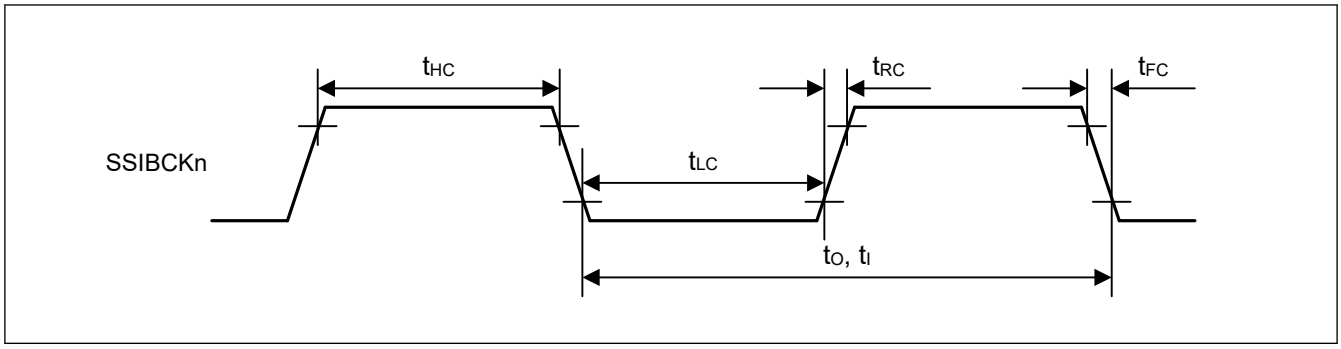


Figure 60.85 SSIE clock input/output timing

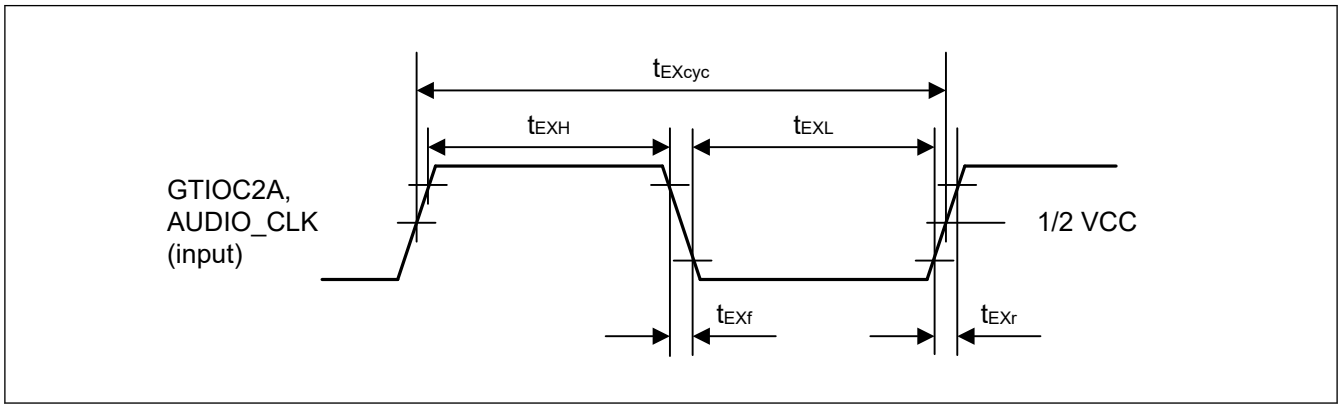


Figure 60.86 Clock input timing

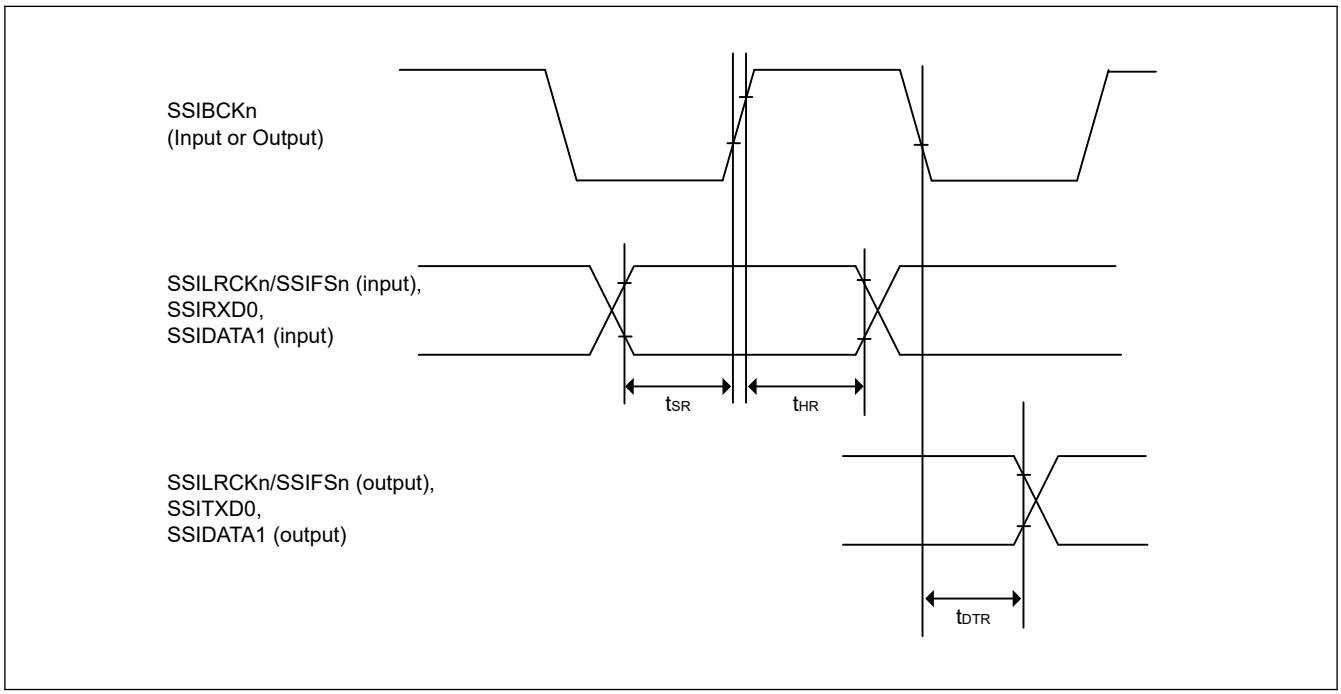


Figure 60.87 SSIE data transmit and receive timing when SSICR.BCKP = 0

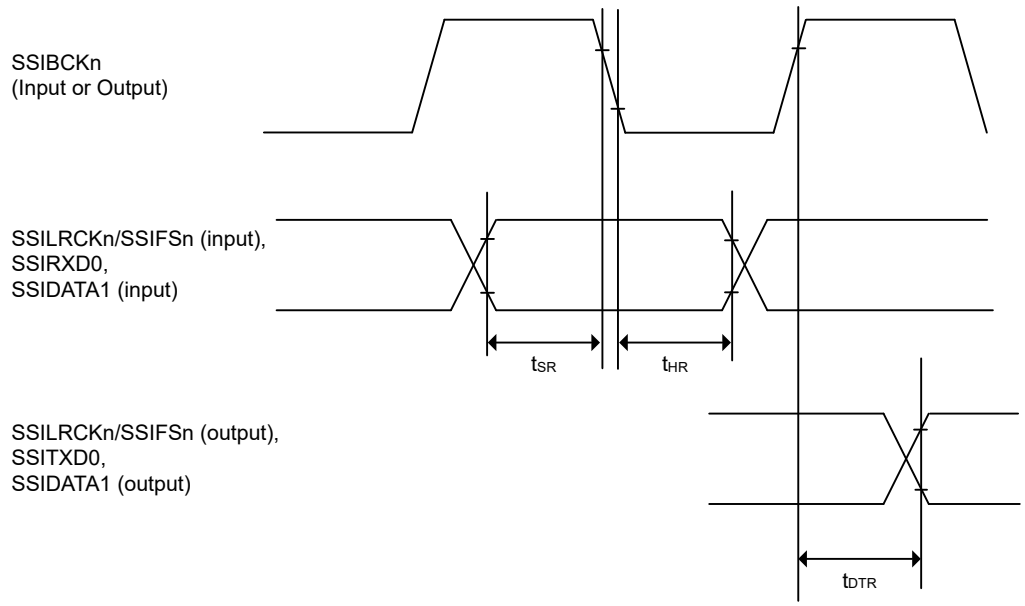
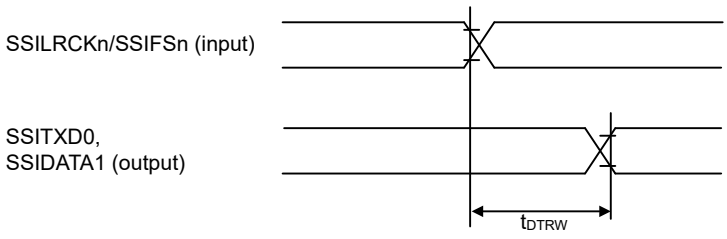


Figure 60.88 SSIE data transmit and receive timing when SSICR.BCKP = 1



MSB bit output delay after SSILRCKn/SSIFSn change for slave transmitter when DEL = 1, SDTA = 0 or DEL = 1, SDTA = 1, SWL[2:0] = DWL[2:0] in SSICR.

Figure 60.89 SSIE data output delay after SSILRCK0/SSIFS0 change

60.3.15 SD/MMC Host Interface Timing

Table 60.56 SD/MMC Host Interface signal timing

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
High-speed high drive output is selected in specific condition, please see note 1.
Clock duty ratio is 50%.

Parameter	Symbol	VCC/VCC2	Min	Max	Unit	Test conditions
SDnCLK clock cycle	t_{SDCYC}	2.70V or above	20	—	ns	Figure 60.90
		1.70 to1.95V*1	20	—		
		1.70 to1.95V	40	—		
SDnCLK clock high pulse width	t_{SDWH}	2.70V or above	6.5	—	ns	
		1.70 to1.95V*1	6.5	—		
		1.70 to1.95V	13.0	—		
SDnCLK clock low pulse width	t_{SDWL}	2.70V or above	6.5	—	ns	
		1.70 to1.95V*1	6.5	—		
		1.70 to1.95V	13.0	—		
SDnCLK clock rise time	t_{SDLH}	2.70V or above	—	3.0	ns	
		1.70 to1.95V*1	—	4.0		
		1.70 to1.95V	—	8.0		
SDnCLK clock fall time	t_{SDHL}	2.70V or above	—	3.0	ns	
		1.70 to1.95V*1	—	4.0		
		1.70 to1.95V	—	8.0		
SDnCMD/SDnDATm output data delay	t_{SDODLY}	2.70V or above	–7.0	4.0	ns	
		1.70 to1.95V*1	–7.0	7.0		
		1.70 to1.95V	–15.0	15.0		
SDnCMD/SDnDATm input data setup	t_{SDIS}	2.70V or above	4.5	—	ns	
		1.70 to1.95V*1	4.5	—		
		1.70 to1.95V	20.0	—		
SDnCMD/SDnDATm input data hold	t_{SDIH}	2.70V or above	1.5	—	ns	
		1.70 to1.95V	1.5	—		

Note: Must use pins that have a letter appended to their name, for instance “_A”, “_B”, to indicate group membership. For the SD/MMC Host interface, the AC portion of the electrical characteristics is measured for each group.
Note: If SD1DAT4_A~SD1DAT7_A are used, characteristics above is guaranteed only when VCC = VCC2.
Note 1. Only supported for Ch0 group B ("SD0*_B") and Ch1 group A ("SD1*_A")
High-speed high drive output is selected in the port drive capability bit in the PmnPFS register for the following pins:
SD0CLK_B, SD1CLK_A

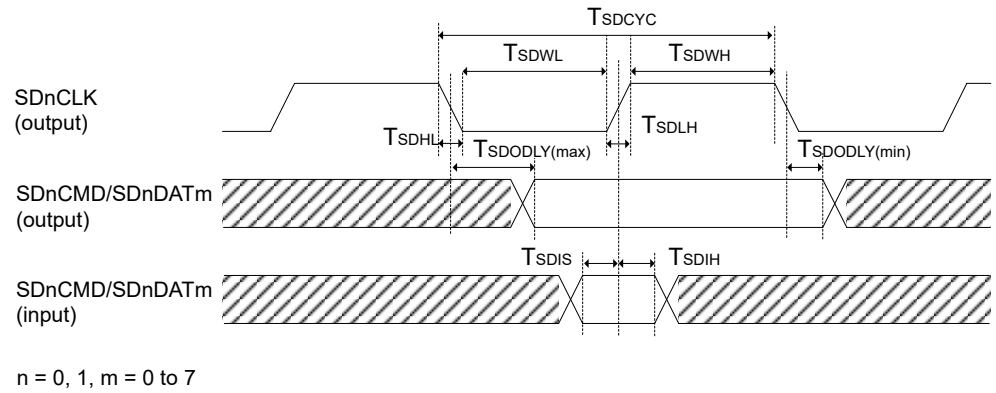


Figure 60.90 SD/MMC Host Interface signal timing

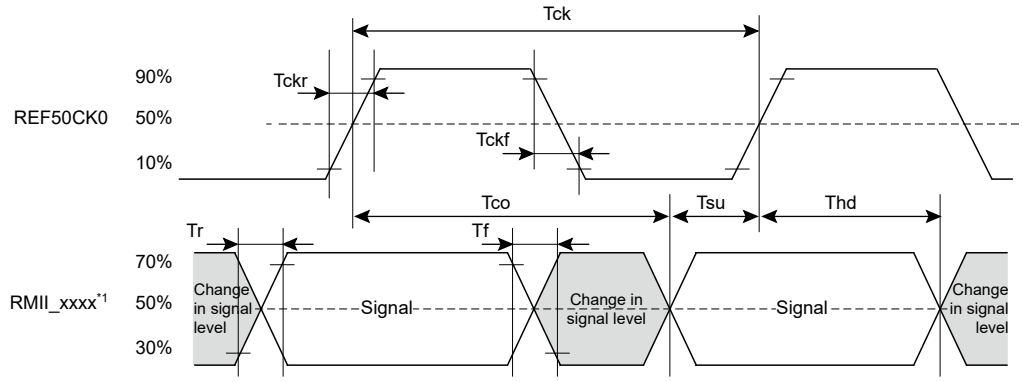
60.3.16 ETHERC Timing

Table 60.57 ETHERC timing

Conditions: ETHERC (RMII): Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register for the following pins: ET0_MDC, ET0_MDIO.
For other pins, high drive output is selected in the Port Drive Capability bit in the PmnPFS register.
ETHERC (MII): Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	VCC	Min	Max	Unit	Test conditions
ETHERC (RMII)	REF50CK0 cycle time	T_{ck}	2.70V or above	20	—	ns	Figure 60.91 to Figure 60.94
	REF50CK0 frequency, typical 50 MHz	—		—	50 + 100 ppm	MHz	
	REF50CK0 duty	—		35	65	%	
	REF50CK0 rise/fall time	$T_{ckr/ckf}$		0.5	3.5	ns	
	RMII_xxxx*1 output delay	T_{co}		2.5	12.0	ns	
	RMII_xxxx*2 setup time	T_{su}		3	—	ns	
	RMII_xxxx*2 hold time	T_{hd}		1	—	ns	
	RMII_xxxx*1, *2 rise/fall time	T_r/T_f		0.5	5.0	ns	
	ET0_WOL output delay	t_{WOLd}		1	23.5	ns	Figure 60.95
ETHERC (MII)	ET0_TX_CLK cycle time	t_{Tcyc}		40	—	ns	—
	ET0_TX_EN output delay	t_{TENd}		1	20	ns	Figure 60.96
	ET0_ETXD0 to ET_ETXD3 output delay	t_{MTDd}		1	20	ns	
	ET0_CRS setup time	t_{CRSs}		10	—	ns	
	ET0_CRS hold time	t_{CRSh}		10	—	ns	
	ET0_COL setup time	t_{COLs}		10	—	ns	Figure 60.97
	ET0_COL hold time	t_{COLh}		10	—	ns	
	ET0_RX_CLK cycle time	t_{TRcyc}		40	—	ns	—
	ET0_RX_DV setup time	t_{RDVs}		10	—	ns	Figure 60.98
	ET0_RX_DV hold time	t_{RDVh}		10	—	ns	
	ET0_ERXD0 to ET_ERXD3 setup time	t_{MRDs}		10	—	ns	
	ET0_ERXD0 to ET_ERXD3 hold time	t_{MRDh}		10	—	ns	
	ET0_RX_ER setup time	t_{RERs}		10	—	ns	Figure 60.99
	ET0_RX_ER hold time	t_{RESh}		10	—	ns	
	ET0_WOL output delay	t_{WOLd}		1	23.5	ns	Figure 60.100

Note: The following pins must use pins that have a letter appended to their name, for instance "_A", "_B", to indicate group membership. For the ETHERC (RMII) Host interface, the AC portion of the electrical characteristics is measured for each group. REF50CK0_A, REF50CK0_B, RMII0_xxxx_A, RMII0_xxxx_B.
Note 1. RMII_TXD_EN, RMII_TXD1, RMII_TXD0.
Note 2. RMII_CRS_DV, RMII_RXD1, RMII_RXD0, RMII_RX_ER.



Note 1. RMII_TXD_EN, RMII_TXD1, RMII_TXD0, RMII_CRS_DV, RMII_RXD1, RMII_RXD0, RMII_RX_ER

Figure 60.91 REF50CK0 and RMII signal timing

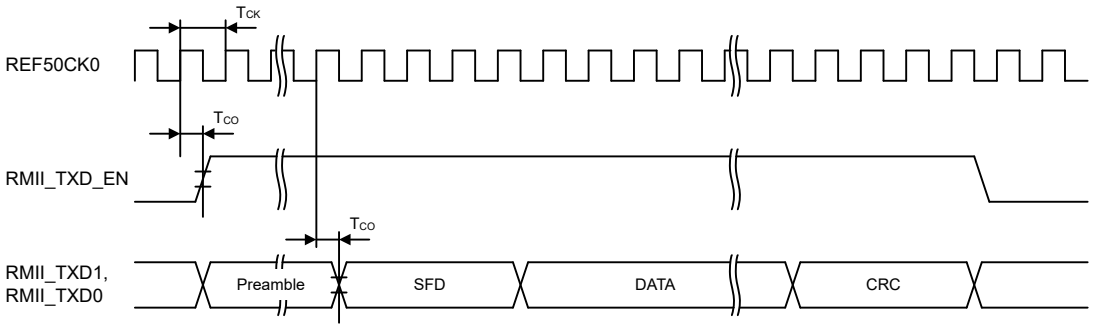


Figure 60.92 RMII transmission timing

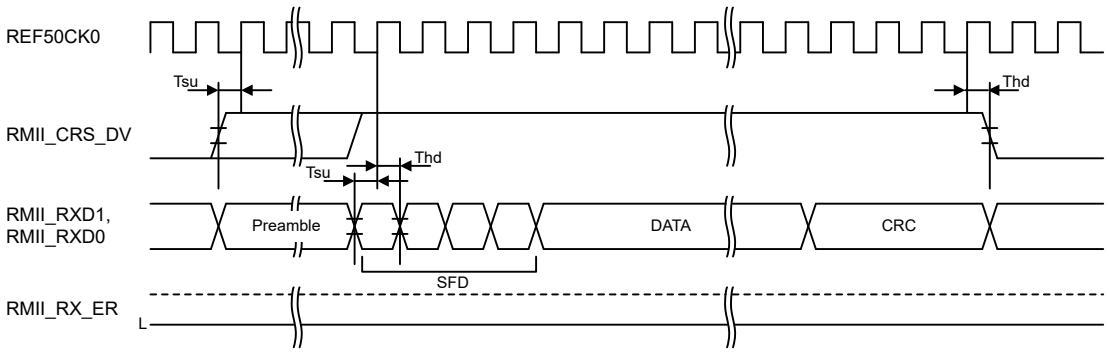


Figure 60.93 RMII reception timing in normal operation

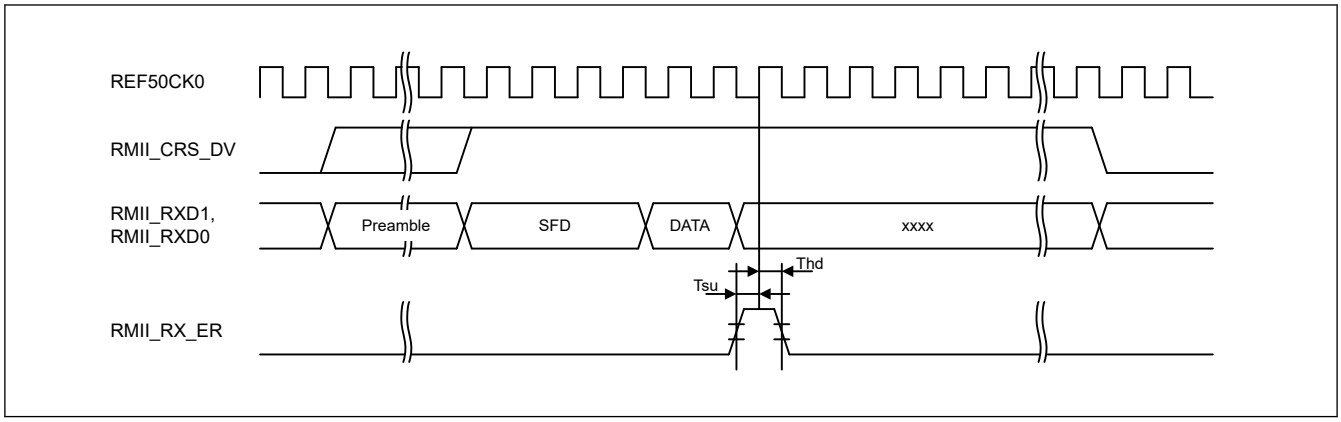


Figure 60.94 RMII reception timing when an error occurs

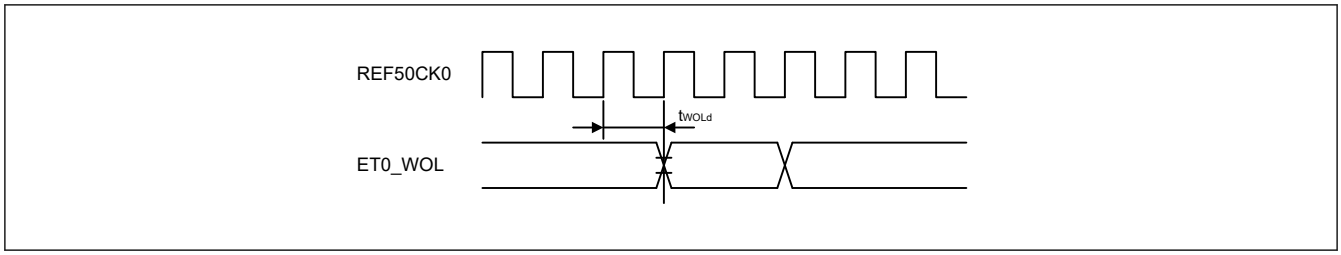


Figure 60.95 WOL output timing for RMII

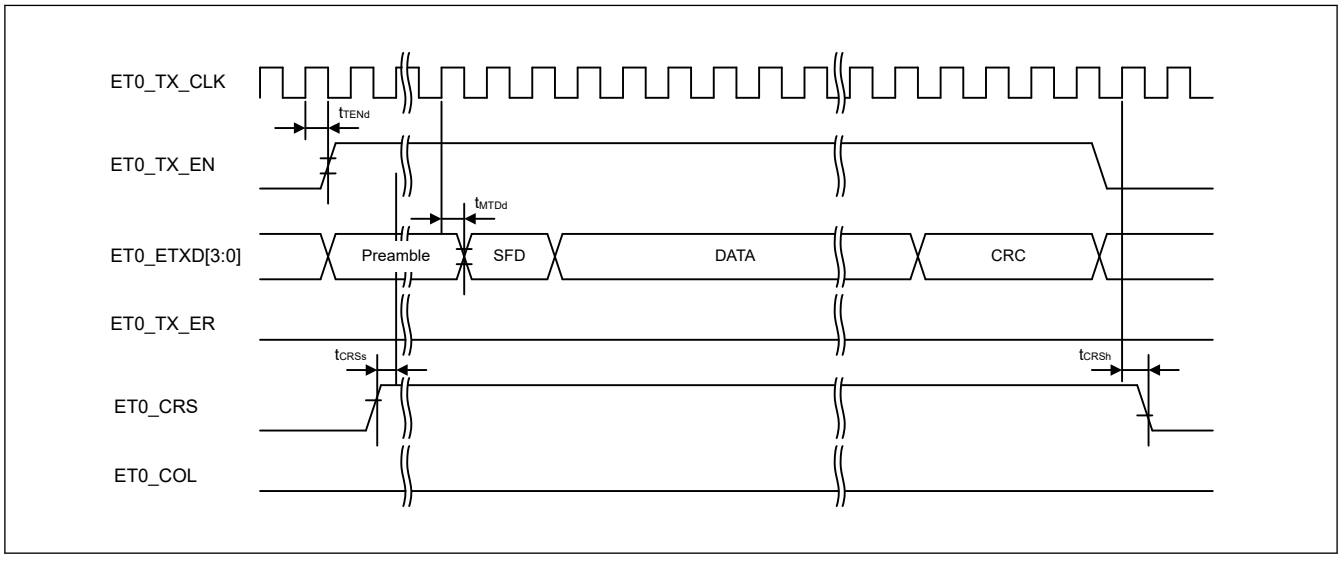


Figure 60.96 MII transmission timing in normal operation

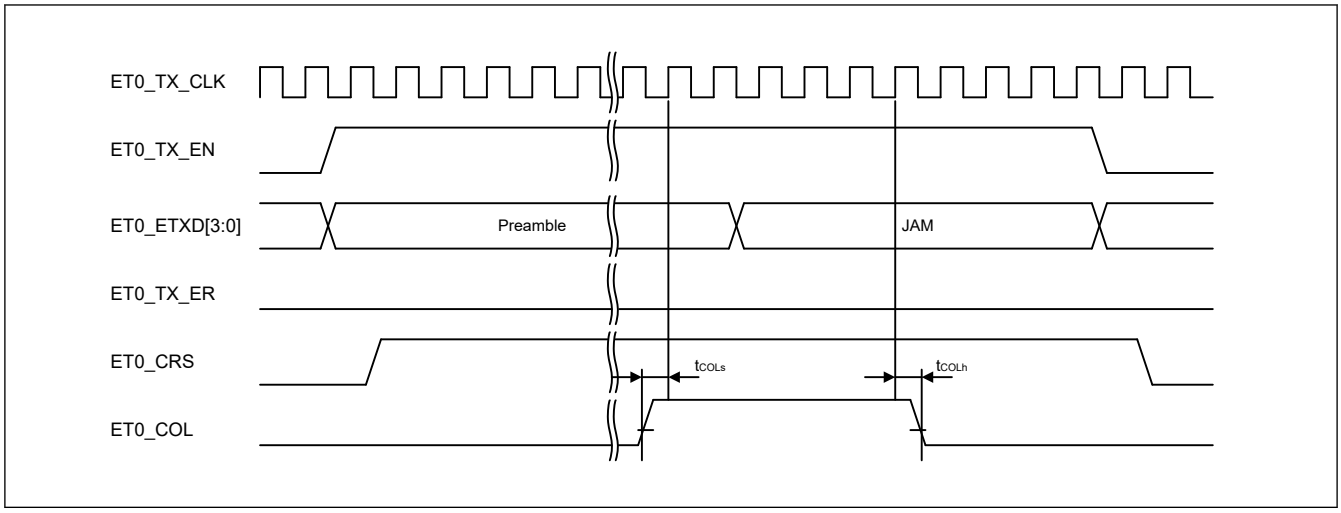


Figure 60.97 MII transmission timing when a conflict occurs

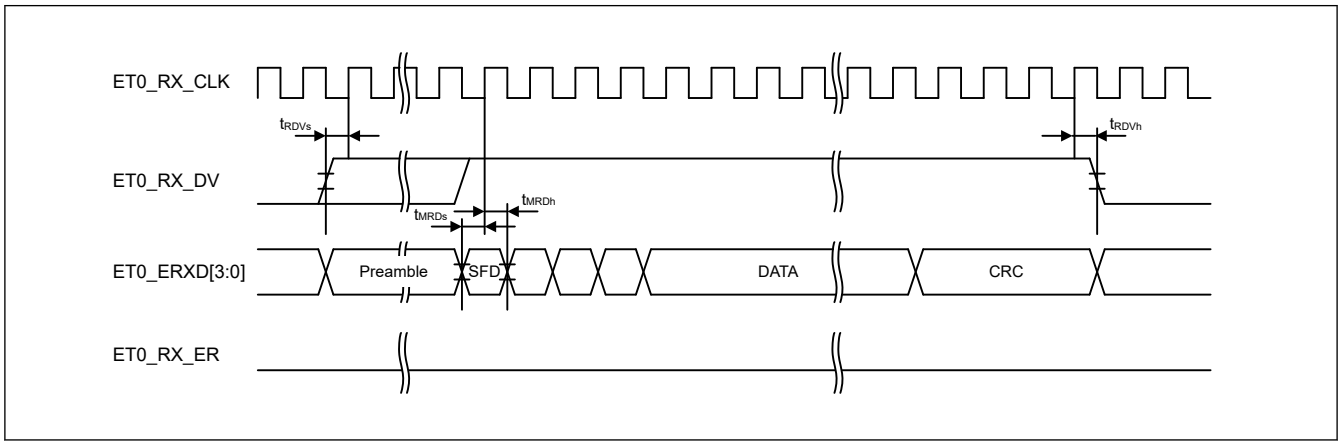


Figure 60.98 MII reception timing in normal operation

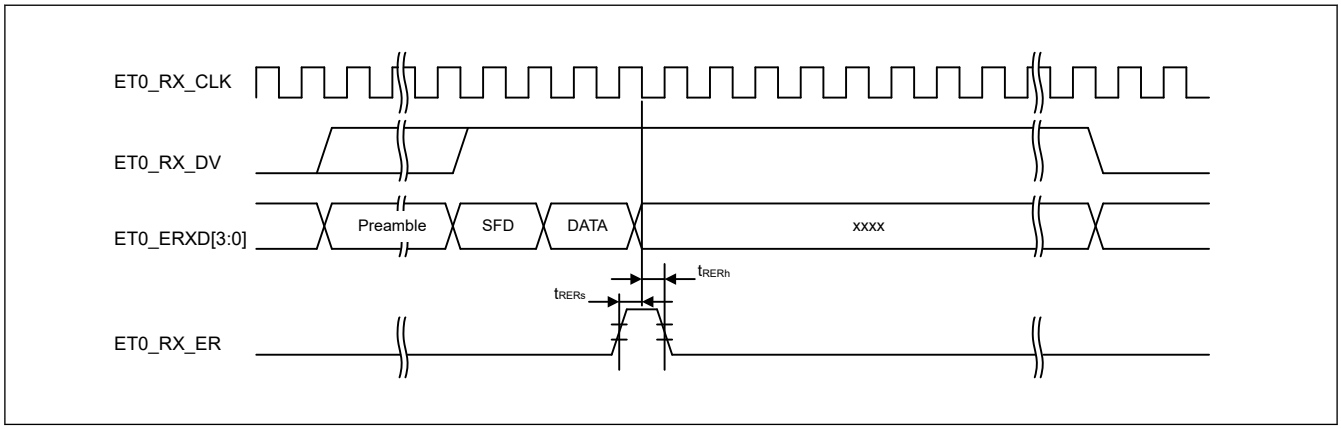


Figure 60.99 MII reception timing when an error occurs

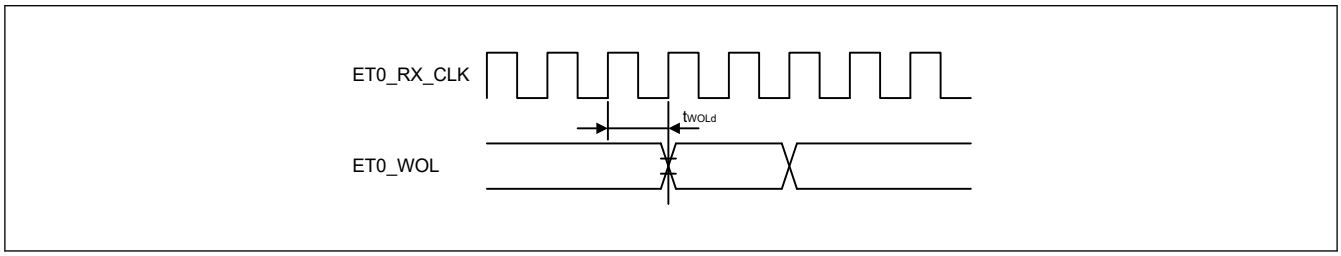


Figure 60.100 WOL output timing for MII

60.3.17 CEU Timing

Table 60.58 Capture Engine Unit Signal Timing

Parameter	Symbol	VCC	Min	Max	Unit	Test conditions
Vertical sync (VIO_VD) setup time (Camera clock rising)	t_{VVDS}	2.70 V or above	2.0	—	ns	Figure 60.101 Figure 60.102
		1.68 V or above	4.5	—		
Vertical sync (VIO_VD) setup time (Camera clock falling)	t_{VVDS}	2.70 V or above	2.5	—	ns	
		1.68 V or above	4.5	—		
Vertical sync (VIO_VD) hold time	t_{VVDH}	2.70 V or above	3.5	—	ns	
		1.68 V or above	5.5	—		
Horizontal sync (VIO_HD) setup time (Camera clock rising)	t_{VHDS}	2.70 V or above	2.0	—	ns	
		1.68 V or above	4.5	—		
Horizontal sync (VIO_HD) setup time (Camera clock falling)	t_{VHDS}	2.70 V or above	2.5	—	ns	
		1.68 V or above	4.5	—		
Horizontal sync (VIO_HD) hold time	t_{VHDH}	2.70 V or above	3.5	—	ns	
		1.68 V or above	5.5	—		
Capture image data (VIO_D) setup time (Camera clock rising)	t_{VDTS}	2.70 V or above	2.0	—	ns	
		1.68 V or above	4.5	—		
Capture image data (VIO_D) setup time (Camera clock falling)	t_{VDTS}	2.70 V or above	2.5	—	ns	
		1.68 V or above	4.5	—		
Capture image data (VIO_D) hold time	t_{VDTH}	2.70 V or above	3.5	—	ns	
		1.68 V or above	5.5	—		
Camera clock cycle	t_{VCYC}	2.70 V or above	11.5	—	ns	
		1.68 V or above	23.0	—		
Camera clock high level width	t_{VHW}	2.70 V or above	0.4 × t_{VCYC}	—	ns	
		1.68 V or above	0.4 × t_{VCYC}	—		
Camera clock low level width	t_{VLW}	2.70 V or above	0.4 × t_{VCYC}	—	ns	
		1.68 V or above	0.4 × t_{VCYC}	—		
Field identification signal (VIO_FLD) setup time (Camera clock rising)	t_{VFDS}	2.70 V or above	2.0	—	ns	
		1.68 V or above	4.5	—		
Field identification signal (VIO_FLD) setup time (Camera clock falling)	t_{VFDS}	2.70 V or above	2.5	—	ns	
		1.68 V or above	4.5	—		
Field identification signal (VIO_FLD) hold time	t_{VFDH}	2.70 V or above	3.5	—	ns	
		1.68 V or above	5.5	—		

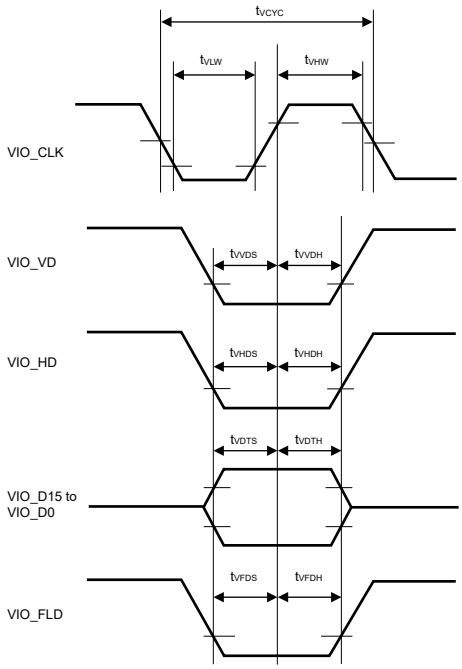


Figure 60.101 Capture Engine Unit Module Signal Timing of data capturing on the rising edge of VIO_CLK

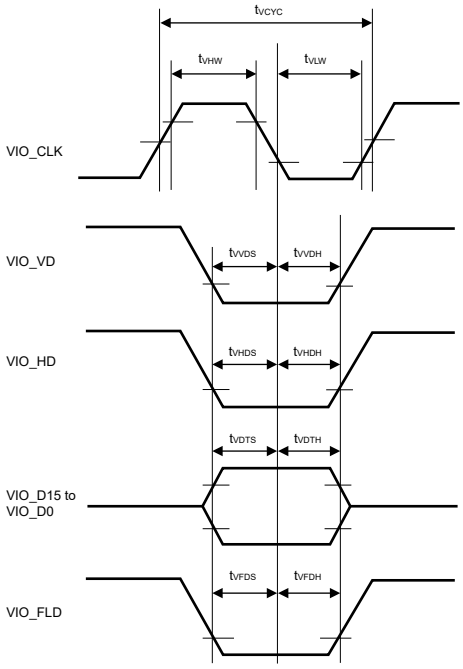


Figure 60.102 Capture Engine Unit Module Signal Timing of data capturing on the falling edge of VIO_CLK

60.3.18 GLCDC Timing

Table 60.59 GLCDC Timing

Conditions:
LCD_CLK: High drive output is selected in the port drive capability bit in the PmnPFS register.
LCD_DATA: Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Parameter		Symbol	VCC	Min	Max	Units	Test conditions
LCD_EXTCLK input clock frequency		t_{Ecyc}	2.70 V or above	—	60*1	MHz	Figure 60.103
			1.68 V or above	—	30		
LCD_EXTCLK input clock low pulse width		t_{WL}	2.70 V or above	0.45	—	t_{Ecyc}	
			1.68 V or above	0.45	—		
LCD_EXTCLK input clock high pulse width		t_{WH}	2.70 V or above	0.45	—		
			1.68 V or above	0.45	—		
LCD_CLK output clock frequency		$1/t_{Lcyc}$	2.70 V or above	—	60*1	MHz	Figure 60.104
			1.68 V or above	—	30		
LCD_CLK output clock low pulse width		t_{LOL}	2.70 V or above	0.4	0.6	t_{Lcyc}	
			1.68 V or above	0.4	0.6		
LCD_CLK output clock high pulse width		t_{LOH}	2.70 V or above	0.4	0.6	t_{Lcyc}	
			1.68 V or above	0.4	0.6		
LCD data output delay timing	_A or _B combinations*2	t_{DD}	2.70 V or above	-3.5	4.0	ns	Figure 60.105
			1.68 V or above	-5.5	6.0		
	_A and _B combinations*3		2.70 V or above	-5.0	5.5		
			1.68 V or above	-7.0	7.5		

Note 1. Parallel RGB888, 666, 565: Maximum 54 MHz
Serial RGB888: Maximum 60 MHz (4x speed)
Note 2. Use pins that have a letter appended to their names, for instance, "_A" or "_B", to indicate.
Note 3. Pins of group "_A" and "_B" combinations are used.

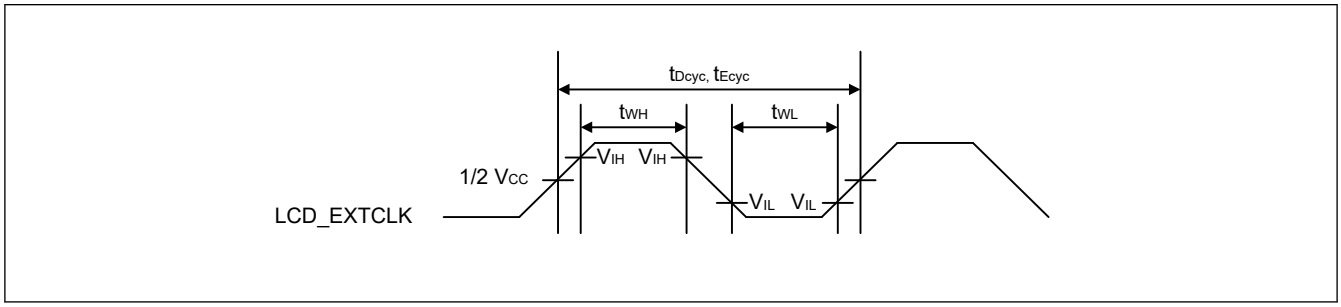


Figure 60.103 LCD_EXTCLK clock timing

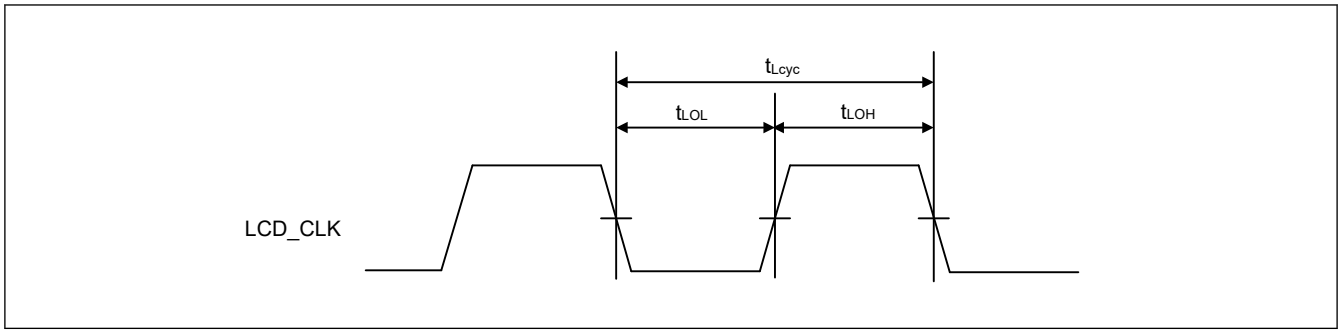


Figure 60.104 LCD_CLK clock output timing

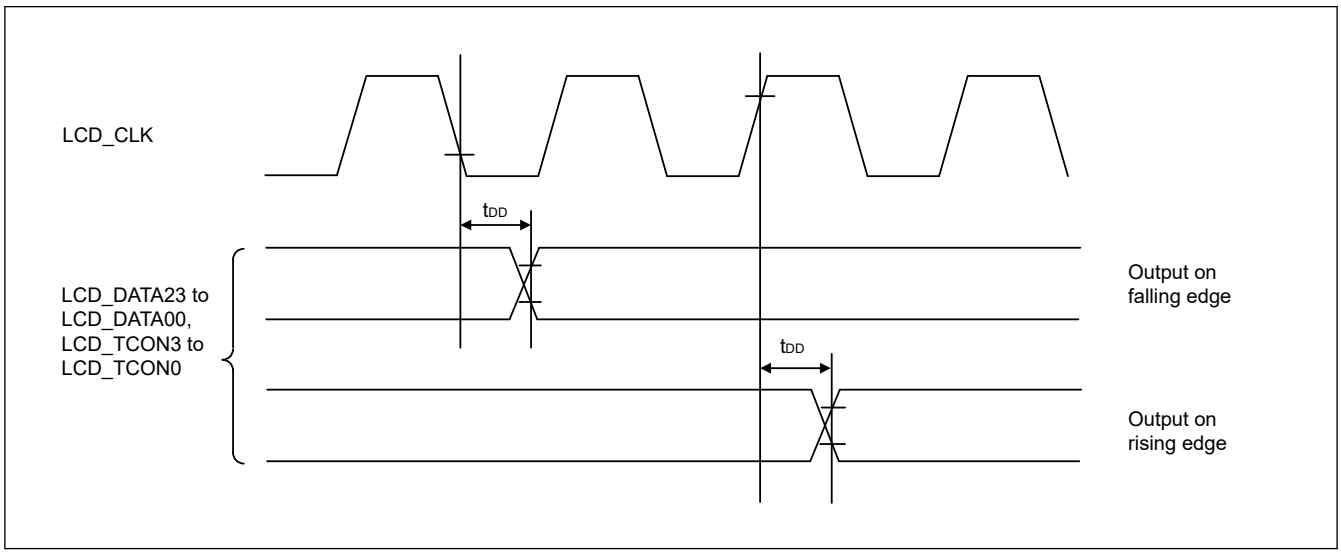


Figure 60.105 Display output timing

60.3.19 CANFD Timing

Table 60.60 CANFD interface timing

Parameter	Symbol	VCC/VCC2	Min	Max	Unit	Test conditions
Internal delay time	t_{node}	2.70 V or above	—	50	ns	Figure 60.106
		1.68 V or above (VCC) 1.65 V or above (VCC2)	—	50		

Note: Internal delay time (t_{node}) = Internal transfer delay time (t_{output}) + Internal receive delay time (t_{input})

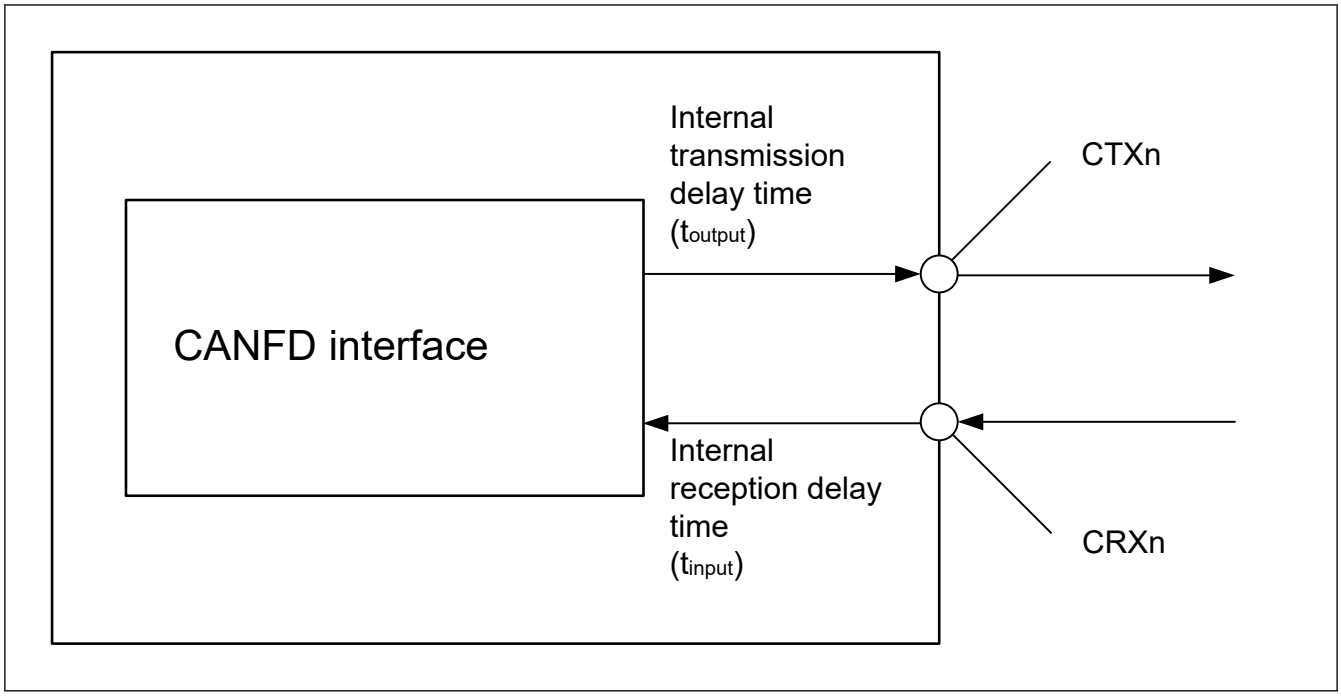


Figure 60.106 CANFD interface condition

60.4 USB Characteristics

60.4.1 USBFS Timing

Table 60.61 USBFS low-speed characteristics for host only (USB_DP and USB_DM pin characteristics)

Conditions: VCC = VCC_USB = 3.0 to 3.6 V, USBCLK = 48 MHz

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Input characteristics	Input high voltage	V_{IH}	2.0	—	—	V	—
	Input low voltage	V_{IL}	—	—	0.8	V	—
	Differential input sensitivity	V_{DI}	0.2	—	—	V	| USB_DP - USB_DM |
	Differential common-mode range	V_{CM}	0.8	—	2.5	V	—
Output characteristics	Output high voltage	V_{OH}	2.8	—	3.6	V	I_{OH} = –200 µA
	Output low voltage	V_{OL}	0.0	—	0.3	V	I_{OL} = 2 mA
	Cross-over voltage	V_{CRS}	1.3	—	2.0	V	Figure 60.107
	Rise time	t_{LR}	75	—	300	ns	
	Fall time	t_{LF}	75	—	300	ns	
	Rise/fall time ratio	t_{LR} / t_{LF}	80	—	125	%	t_{LR}/ t_{LF}
Pull-up and pull-down characteristics	USB_DP and USB_DM pull-down resistance in host controller mode	R_{pd}	14.25	—	24.80	kΩ	—

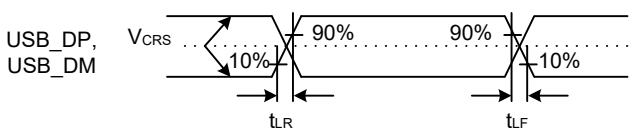


Figure 60.107 USB_DP and USB_DM output timing in low-speed mode

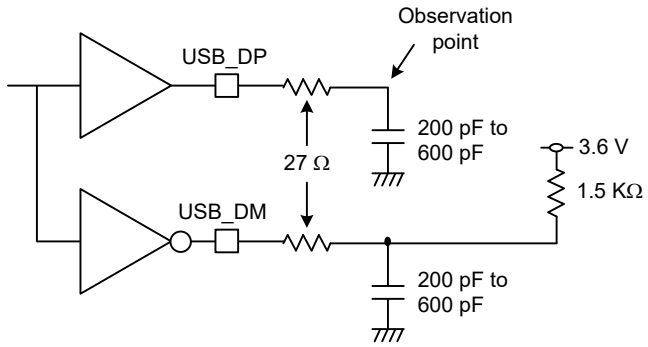


Figure 60.108 Test circuit in low-speed mode

Table 60.62 USBFS full-speed characteristics (USB_DP and USB_DM pin characteristics)

Conditions: VCC = VCC_USB = 3.0 to 3.6 V, USBCLK = 48 MHz

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Input characteristics	Input high voltage	V_{IH}	2.0	—	—	V	—
	Input low voltage	V_{IL}	—	—	0.8	V	—
	Differential input sensitivity	V_{DI}	0.2	—	—	V	| USB_DP - USB_DM |
	Differential common-mode range	V_{CM}	0.8	—	2.5	V	—
Output characteristics	Output high voltage	V_{OH}	2.8	—	3.6	V	I_{OH} = –200 µA
	Output low voltage	V_{OL}	0.0	—	0.3	V	I_{OL} = 2 mA
	Cross-over voltage	V_{CRS}	1.3	—	2.0	V	Figure 60.109
	Rise time	t_{LR}	4	—	20	ns	
	Fall time	t_{LF}	4	—	20	ns	
	Rise/fall time ratio	t_{LR} / t_{LF}	90	—	111.11	%	t_{FR}/ t_{FF}
	Output resistance	Z_{DRV}	28	—	44	Ω	USBFS: Rs = 27 Ω included
Pull-up and pull-down characteristics	DM pull-up resistance in device controller mode	R_{pu}	0.900	—	1.575	kΩ	During idle state
			1.425	—	3.090	kΩ	During transmission and reception
	USB_DP and USB_DM pull-down resistance in host controller mode	R_{pd}	14.25	—	24.80	kΩ	—

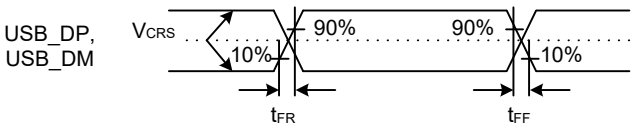


Figure 60.109 USB_DP and USB_DM output timing in full-speed mode

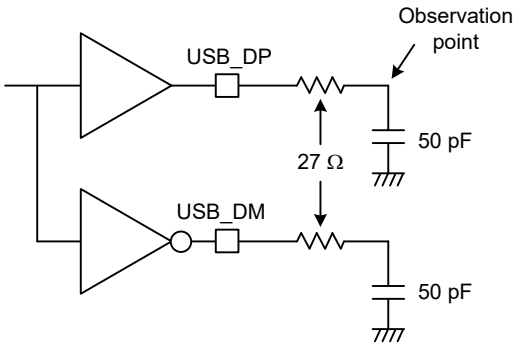


Figure 60.110 Test circuit in full-speed mode

60.4.2 USBHS Timing

Table 60.63 USBHS low-speed characteristics for host only (USB_DP and USB_DM pin characteristics)

Conditions: USBHS_RREF = 2.2 kΩ ± 1%, USBMCLK = 12/20/24/48 MHz, USBCLK = 48MHz, USB60CLK = 60MHz

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Input characteristics	Input high voltage	V_{IH}	2.0	—	—	V	—
	Input low voltage	V_{IL}	—	—	0.8	V	—
	Differential input sensitivity	V_{DI}	0.2	—	—	V	| USB_DP - USB_DM |
	Differential common-mode range	V_{CM}	0.8	—	2.5	V	—
Output characteristics	Output high voltage	V_{OH}	2.8	—	3.6	V	I_{OH} = –200 µA
	Output low voltage	V_{OL}	0.0	—	0.3	V	I_{OL} = 2 mA
	Cross-over voltage	V_{CRS}	1.3	—	2.0	V	Figure 60.111
	Rise time	t_{LR}	75	—	300	ns	
	Fall time	t_{LF}	75	—	300	ns	
	Rise/fall time ratio	t_{LR} / t_{LF}	80	—	125	%	t_{LR}/ t_{LF}
Pull-up and pull-down characteristics	USB_DP and USB_DM pull-down resistance in host controller mode	R_{pd}	14.25	—	24.80	kΩ	—

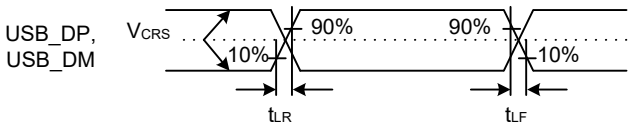


Figure 60.111 USB_DP and USB_DM output timing in low-speed mode

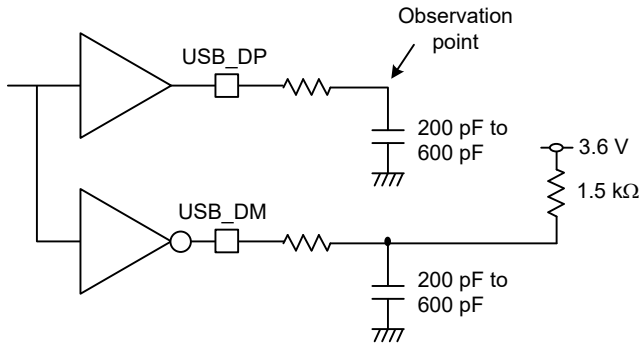


Figure 60.112 Test circuit in low-speed mode

Table 60.64 USBHS full-speed characteristics (USB_DP and USB_DM pin characteristics)

Conditions: USBHS_RREF = 2.2 kΩ ± 1%, USBMCLK = 12/20/24/48 MHz, USBCLK = 48MHz, USB60CLK = 60MHz

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Input characteristics	Input high voltage	V_{IH}	2.0	—	—	V	—
	Input low voltage	V_{IL}	—	—	0.8	V	—
	Differential input sensitivity	V_{DI}	0.2	—	—	V	| USB_DP - USB_DM |
	Differential common-mode range	V_{CM}	0.8	—	2.5	V	—
Output characteristics	Output high voltage	V_{OH}	2.8	—	3.6	V	I_{OH} = –200 µA
	Output low voltage	V_{OL}	0.0	—	0.3	V	I_{OL} = 2 mA
	Cross-over voltage	V_{CRS}	1.3	—	2.0	V	Figure 60.113
	Rise time	t_{LR}	4	—	20	ns	
	Fall time	t_{LF}	4	—	20	ns	
	Rise/fall time ratio	t_{LR} / t_{LF}	90	—	111.11	%	t_{FR}/ t_{FF}
	Output resistance	Z_{DRV}	40.5	—	49.5	Ω	Rs Not used (PHYSET.REPSEL[1:0] = 01b and PHYSET. HSEB = 0)
Pull-up and pull-down characteristics	DM pull-up resistance in device controller mode	R_{pu}	0.900	—	1.575	kΩ	During idle state
			1.425	—	3.090	kΩ	During transmission and reception
	USB_DP and USB_DM pull-down resistance in host controller mode	R_{pd}	14.25	—	24.80	kΩ	—

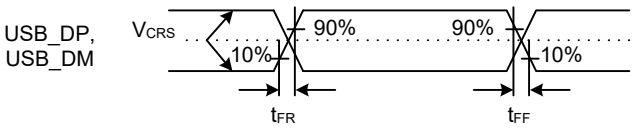


Figure 60.113 USB_DP and USB_DM output timing in full-speed mode

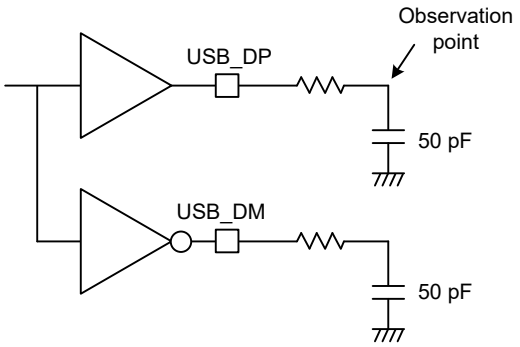


Figure 60.114 Test circuit in full-speed mode

Table 60.65 USB High Speed Characteristics (USB_DP and USB_DM Pin Characteristics)

Conditions: USBHS_RREF = 2.2 kΩ ± 1%, USBMCLK = 12/20/24/48 MHz

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Input characteristics	Squelch detect sensitivity	V_{HSSQ}	100	—	150	mV	Figure 60.115
	Disconnect detect sensitivity	V_{HSDSC}	525	—	625	mV	Figure 60.116
	Common mode voltage	V_{HSCM}	-50	—	500	mV	—
Output characteristics	Idle state	V_{HSOI}	-10	—	10	mV	—
	Output high level voltage	V_{HSOH}	360	—	440	mV	—
	Output low level voltage	V_{HSOL}	-10	—	10	mV	—
	Chirp J output voltage (difference)	V_{CHIRPJ}	700	—	1100	mV	—
	Chirp K output voltage (difference)	V_{CHIRPK}	-900	—	-500	mV	—
AC characteristics	Rise time	t_{HSR}	500	—	—	ps	—
	Fall time	t_{HSF}	500	—	—	ps	Figure 60.117
	Output resistance	Z_{HSDRV}	40.5	—	49.5	Ω	—

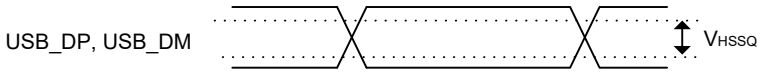


Figure 60.115 USB_DP and USB_DM squelch detect sensitivity (high-speed)

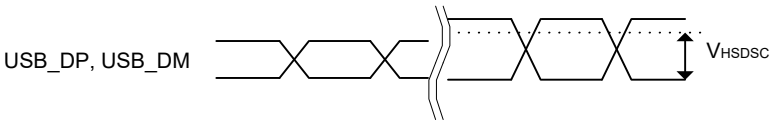


Figure 60.116 USB_DP and USB_DM disconnect detect sensitivity (high-speed)

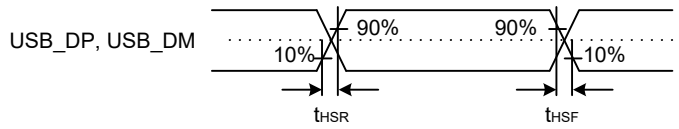


Figure 60.117 USB_DP and USB_DM output timing (high-speed)

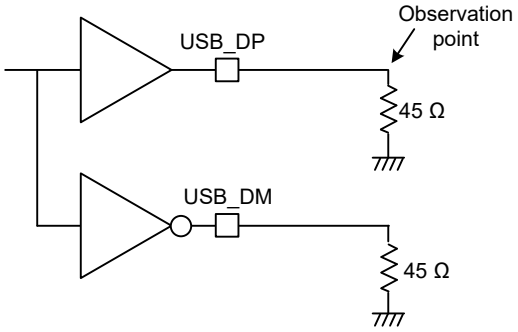


Figure 60.118 Test circuit (high-speed)

Table 60.66 USBHS high-speed characteristics (USB_DP and USB_DM pin characteristics)

Conditions: USBHS_RREF = 2.2 kΩ ± 1%, USBMCLK = 12/20/24/48 MHz

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Battery Charging Specification	D+ sink current	I_{DP_SINK}	25	—	175	µA	—
	D- sink current	I_{DM_SINK}	25	—	175	µA	—
	DCD source current	I_{DP_SRC}	7	—	13	µA	—
	Data detection voltage	V_{DAT_REF}	0.25	—	0.40	V	—
	D+ source voltage	V_{DP_SRC}	0.5	—	0.7	V	Output current = 250 µA
	D- source voltage	V_{DM_SRC}	0.5	—	0.7	V	Output current = 250 µA

60.5 MIPI D-PHY Characteristics

Table 60.67 Pin Characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Note
Pin leakage current	I_{LEAK}	−100	—	100	µA	
Pin signal voltage range	V_{PIN}	−50	—	1350	mV	
Ground shift	V_{GNDSH}	−50	—	50	mV	

Table 60.68 HS-TX Characteristics (1 of 2)

Parameter	Symbol	Min	Typ	Max	Unit	Note
HS transmit static common-mode voltage	V_{CMTX}	150	200	250	mV	
Vcmtx mismatch when output is Differential-1 or Differential-0	$\lvert\Delta V_{CMTX(1,0)}\rvert$	—	—	5.0	mV	
HS transmit differential voltage	|VOD|	140	200	270	mV	
VOD mismatch when output is Differential-1 or Differential-0	|ΔVOD|	—	—	14.0	mV	
HS output high voltage	V_{OHHS}	—	—	360	mV	

Table 60.68 HS-TX Characteristics (2 of 2)

Parameter		Symbol	Min	Typ	Max	Unit	Note
Single ended output impedance		Z_{OS}	40.0	50.0	62.5	Ω	
Single ended output impedance mismatch		ΔZ_{OS}	—	—	20	%	
Data rate		—	40	—	720	Mbps	1 lane
TX Data to Clock Skew		$T_{SKEW[TX]}$	−0.20	—	0.20	UIhs	
Common-level variations above 450MHz		$\Delta V_{CMTX(HF)}$	—	—	15.0	mVrms	
Common-level variations between 50-450MHz		$\Delta V_{CMTX(LF)}$	—	—	25.0	mVpeak	
20%-80% rise time and fall time		t_R	—	—	0.35	UIhs	
			100	—	—	ps	
		t_F	—	—	0.35	UIhs	
			100	—	—	ps	
Clock UI instantaneous		UI_{INST}	—	—	12.5	ns	
Clock UI variation		ΔUI	−10 %	—	10 %	UIhs	
Differential Return Loss	fh_{MIN}	Sdd_{TX}	—	—	−15.00	dB	
	fh_{MIN}		—	—	−4.50		
	f_{MAX}		—	—	−2.50		
Common-mode Return Loss	$1/4f_{INT\ MIN}$	Scc_{TX}	—	—	0.00	dB	
	$f_{INT\ MIN}$		—	—	−6.00		
	f_{MAX}		—	—	−6.00		

Table 60.69 LP-TX Characteristics (1 of 2)

Parameter		Symbol	Min	Typ	Max	Unit	Note
Thevenin output high level		V_{OH}	1.10	1.20	1.30	V	
Thevenin output low level		V_{OL}	−50	—	50	mV	
Output impedance of LP transmitter	high input	Z_{OLP}	110	—	—	Ω	
	low input	Z_{OLP}	110	—	—	Ω	
15%-85% rise time and fall time		T_{RLP}	—	—	25.0	ns	
		T_{FLP}	—	—	25.0	ns	
30%-85% rise time and fall time		T_{REOT}	—	—	35.0	ns	
Pulse width of the LP exclusive-OR clock	First LP exclusive-OR clock pulse after Stop state or last pulse before Stop state	$T_{LP-PULSE-TX}$	40	—	—	ns	
	All other pulses		20	—	—	ns	
Period of the LP exclusive-OR clock		$T_{LP-PER-TX}$	90	—	—	ns	

Table 60.69 LP-TX Characteristics (2 of 2)

Parameter		Symbol	Min	Typ	Max	Unit	Note
Slew rate at C_{LOAD} = 0pF	rise	$\delta V/\delta t_{SR}$	—	—	500	mV/ns	
	fall		—	—	500	mV/ns	
Slew rate at C_{LOAD} = 5pF	rise		—	—	300	mV/ns	
	fall		—	—	300	mV/ns	
Slew rate at C_{LOAD} = 20pF	rise		—	—	250	mV/ns	
	fall		—	—	250		
Slew rate at C_{LOAD} = 70pF	rise		—	—	150	mV/ns	
	fall		—	—	150	mV/ns	
Slew rate at CLOAD = 0 to 70pF (falling edge only)			30	—	—	mV/ns	
Slew rate at CLOAD = 0 to 70pF (rising edge only)			30	—	—	mV/ns	
Slew rate at CLOAD = 0 to 70pF (rising edge only)			30 − 0.075 × (Vo,inst − 700)	—	—	mV/ns	Vo,inst is the instantaneous output voltage in millivolts
Load capacitance		C_{LOAD}	0	—	70	pF	

Table 60.70 LP-RX Characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Note
Logic 1 input voltage	V_{IH}	740	—	—	mV	
Logic 0 input voltage, not in ULP state	V_{IL}	—	—	550	mV	
Logic 0 input voltage, ULP state	$V_{IL-ULPS}$	—	—	300	mV	
Input hysteresis	V_{HYST}	25.0	—	—	mV	
Input pulse rejection	e_{SPIKE}	—	—	300	Vps	
Minimum pulse width response	T_{MIN-RX}	20	—	—	ns	
Peak interference amplitude	V_{INT}	—	—	200	mV	
Interference frequency	f_{INT}	450	—	—	MHz	

Table 60.71 LP-CD Characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Note
Logic 1 contention threshold	V_{IHCD}	450	—	—	mV	
Logic 0 contention threshold	V_{ILCD}	—	—	200	mV	
Input pulse rejection	e_{SPIKE}	—	—	300	Vps	
Peak interference amplitude	V_{INT}	—	—	200	mV	
Interference frequency	f_{INT}	450	—	—	MHz	

60.6 ADC12 Characteristics

Table 60.72 A/D conversion characteristics for unit 0 (DCDC mode) (1 of 3)

Conditions: PCLKC = 1 to 60 MHz

Parameter	Min	Typ	Max	Unit	Test conditions
Frequency	1	—	60	MHz	—
Analog input capacitance	—	—	30	pF	—
Quantization error	—	±0.5	—	LSB	—

Table 60.72 A/D conversion characteristics for unit 0 (DCDC mode) (2 of 3)

Conditions: PCLKC = 1 to 60 MHz

Parameter			Min	Typ	Max	Unit	Test conditions
Resolution			—	—	12	Bits	—
Channel-dedicated sample-and-hold circuits in use (AN000 to AN002)	Conversion time*1 (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	1.06 (0.4 + 0.25)*2	—	—	µs	• Sampling of channel-dedicated sample-and-hold circuits in 24 states • Sampling in 15 states
	Offset error		—	±1.5	±3.5	LSB	AN000 to AN002 = 0.25 V
	Full-scale error		—	±1.5	±3.5	LSB	AN000 to AN002 = VREFH0 - 0.25 V
	Absolute accuracy		—	±2.5	±10.5	LSB	LQFP package AVCC0 = 2.7 to 3.6V VREFH0 = 2.7V to AVCC0
			—	±2.5	±7.5		LQFP package AVCC0 = VREFH0 = 2.7 to 3.6V
			—	±2.5	±5.5		BGA package AVCC0 = 2.7 to 3.6V VREFH0 = 2.7V to AVCC0
	DNL differential nonlinearity error		—	±1.0	±2.0	LSB	—
	INL integral nonlinearity error		—	±1.5	±4.0	LSB	—
	Holding characteristics of sample-and hold circuits		—	—	20	µs	—
	Dynamic range		0.25	—	VREFH 0 - 0.25	V	—
High-precision channels, Channel-dedicated sample-and-hold circuits not in use (AN000 to AN002, AN004 to AN008)	Conversion time*1 (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	0.48 (0.267)*2	—	—	µs	Sampling in 16 states
		Max. = 400 Ω	0.40 (0.183)*2	—	—	µs	Sampling in 11 states AVCC0 = VREFH0 = 3.0 to 3.6 V
	Offset error		—	±1.0	±2.5	LSB	—
	Full-scale error		—	±1.0	±3.5	LSB	—
	Absolute accuracy		—	±2.0	±7.5	LSB	LQFP package AVCC0 = 2.7 to 3.6V VREFH0 = 2.7V to AVCC0
			—	±2.0	±6.0		LQFP package AVCC0 = VREFH0 = 2.7 to 3.6V
			—	±2.0	±5.5		BGA package AVCC0 = 2.7 to 3.6V VREFH0 = 2.7V to AVCC0
	DNL differential nonlinearity error		—	±0.5	±2.0	LSB	—
	INL integral nonlinearity error		—	±1.0	±2.5	LSB	—

Table 60.72 A/D conversion characteristics for unit 0 (DCDC mode) (3 of 3)

Conditions: PCLKC = 1 to 60 MHz

Parameter			Min	Typ	Max	Unit	Test conditions
Normal-precision channels (AN016 to AN019)	Conversion time*1 (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	0.88 (0.667)*2	—	—	µs	Sampling in 40 states
	Offset error		—	±1.0	±5.5	LSB	—
	Full-scale error		—	±1.0	±5.5	LSB	—
	Absolute accuracy		—	±2.0	±10.0	LSB	LQFP package AVCC0 = 2.7 to 3.6V VREFH0 = 2.7V to AVCC0
			—	±2.0	±7.5		LQFP package AVCC0 = VREFH0 = 2.7 to 3.6V
			—	±2.0	±7.5		BGA package AVCC0 = 2.7 to 3.6V VREFH0 = 2.7V to AVCC0
	DNL differential nonlinearity error		—	±0.5	±4.5	LSB	—
	INL integral nonlinearity error		—	±1.0	±5.5	LSB	—

Note: These specification values apply when only one A/D is operating and D/A and ACMPHS are not operating and there is no access to the external bus during A/D conversion.
If other A/D, D/A, or ACMPHS is operating or bus access occurs during A/D conversion, values might not fall within the indicated ranges.
The use of ports 0 as digital outputs is not allowed when the 12-Bit A/D converter is used.
The characteristics apply when AVCC0, AVSS0, VREFH0, VREFH, VREFL0, VREFL, and 12-bit A/D converter input voltage is stable.

Note 1. The conversion time includes the sampling and comparison times. The number of sampling states is indicated for the test conditions.

Note 2. Values in parentheses indicate the sampling time.

Table 60.73 A/D conversion characteristics for unit 1 (DCDC mode) (1 of 2)

Conditions: PCLKC = 1 to 60 MHz

Parameter			Min	Typ	Max	Unit	Test conditions
Frequency			1	—	60	MHz	—
Analog input capacitance			—	—	30	pF	—
Quantization error			—	±0.5	—	LSB	—
Resolution			—	—	12	Bits	—
High-precision channels (AN100 to AN102, AN104 to AN106)	Conversion time*1 (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	0.48 (0.267)*2	—	—	µs	Sampling in 16 states
		Max. = 400 Ω	0.40 (0.183)*2	—	—	µs	Sampling in 11 states AVCC0 = VREFH = 3.0 to 3.6 V
	Offset error		—	±1.0	±2.5	LSB	—
	Full-scale error		—	±1.0	±3.5	LSB	—
	Absolute accuracy		—	±2.0	±7.5	LSB	LQFP package AVCC0 = 2.7 to 3.6V VREFH = 2.7V to AVCC0
			—	±2.0	±6.0		LQFP package AVCC0 = VREFH = 2.7 to 3.6V
			—	±2.0	±5.5		BGA package AVCC0 = 2.7 to 3.6V VREFH = 2.7V to AVCC0
	DNL differential nonlinearity error		—	±0.5	±2.0	LSB	—
	INL integral nonlinearity error		—	±1.0	±2.5	LSB	—

Table 60.73 A/D conversion characteristics for unit 1 (DCDC mode) (2 of 2)

Conditions: PCLKC = 1 to 60 MHz

Parameter			Min	Typ	Max	Unit	Test conditions
Normal-precision channels (AN116 to AN122)	Conversion time*1 (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	0.88 (0.667)*2	—	—	µs	Sampling in 40 states
	Offset error		—	±1.0	±5.5	LSB	—
	Full-scale error		—	±1.0	±5.5	LSB	—
	Absolute accuracy		—	±2.0	±10.0	LSB	LQFP package AVCC0 = 2.7 to 3.6V VREFH = 2.7V to AVCC0
			—	±2.0	±7.5		LQFP package AVCC0 = VREFH = 2.7 to 3.6V
			—	±2.0	±7.5		BGA package AVCC0 = 2.7 to 3.6V VREFH = 2.7V to AVCC0
	DNL differential nonlinearity error		—	±0.5	±4.5	LSB	—
	INL integral nonlinearity error		—	±1.0	±5.5	LSB	—

Note: These specification values apply when only one A/D is operating and D/A and ACMPHS are not operating and there is no access to the external bus during A/D conversion.
If other A/D, D/A, or ACMPHS is operating or bus access occurs during A/D conversion, values might not fall within the indicated ranges.
The use of ports 0 as digital outputs is not allowed when the 12-Bit A/D converter is used.
The characteristics apply when AVCC0, AVSS0, VREFH0, VREFH, VREFL0, VREFL, and 12-bit A/D converter input voltage are stable.

Note 1. The conversion time includes the sampling and comparison times. The number of sampling states is indicated for the test conditions.

Note 2. Values in parentheses indicate the sampling time.

Table 60.74 A/D conversion characteristics for unit 0 (External VDD mode) (1 of 2)

Conditions: PCLKC = 1 to 60 MHz
AVCC0 = 2.7 to 3.6 V, VREFH0 = 2.7 to 3.6 V

Parameter			Min	Typ	Max	Unit	Test conditions
Frequency			1	—	60	MHz	—
Analog input capacitance			—	—	30	pF	—
Quantization error			—	±0.5	—	LSB	—
Resolution			—	—	12	Bits	—
Channel-dedicated sample-and-hold circuits in use (AN000 to AN002)	Conversion time*1 (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	1.06 (0.4 + 0.25)*2	—	—	µs	• Sampling of channel-dedicated sample-and-hold circuits in 24 states • Sampling in 15 states
	Offset error		—	±1.5	±3.5	LSB	AN000 to AN002 = 0.25 V
	Full-scale error		—	±1.5	±3.5	LSB	AN000 to AN002 = VREFH0 - 0.25 V
	Absolute accuracy		—	±2.5	±5.5	LSB	—
	DNL differential nonlinearity error		—	±1.0	±2.0	LSB	—
	INL integral nonlinearity error		—	±1.5	±3.0	LSB	—
	Holding characteristics of sample-and hold circuits		—	—	20	µs	—
	Dynamic range		0.25	—	VREFH 0 - 0.25	V	—

Table 60.74 A/D conversion characteristics for unit 0 (External VDD mode) (2 of 2)

Conditions: PCLKC = 1 to 60 MHz
AVCC0 = 2.7 to 3.6 V, VREFH0 = 2.7 to 3.6 V

Parameter			Min	Typ	Max	Unit	Test conditions
High-precision channels, Channel-dedicated sample-and-hold circuits not in use (AN000 to AN002, AN004 to AN008)	Conversion time*1 (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	0.48 (0.267)*2	—	—	µs	Sampling in 16 states
		Max. = 400 Ω	0.40 (0.183)*2	—	—	µs	Sampling in 11 states AVCC0 = 3.0 to 3.6 V 3.0 V ≤ VREFH0 ≤ AVCC0
	Offset error		—	±1.0	±2.5	LSB	—
	Full-scale error		—	±1.0	±3.5	LSB	—
	Absolute accuracy		—	±2.0	±4.5	LSB	—
	DNL differential nonlinearity error		—	±0.5	±1.5	LSB	—
	INL integral nonlinearity error		—	±1.0	±2.5	LSB	—
Normal-precision channels (AN016 to AN019)	Conversion time*1 (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	0.88 (0.667)*2	—	—	µs	Sampling in 40 states
	Offset error		—	±1.0	±5.5	LSB	—
	Full-scale error		—	±1.0	±5.5	LSB	—
	Absolute accuracy		—	±2.0	±7.5	LSB	—
	DNL differential nonlinearity error		—	±0.5	±4.5	LSB	—
	INL integral nonlinearity error		—	±1.0	±5.5	LSB	—

Note: These specification values apply when only one A/D is operating and D/A and ACMPHS are not operating and there is no access to the external bus during A/D conversion.
If other A/D, D/A, or ACMPHS is operating or bus access occurs during A/D conversion, values might not fall within the indicated ranges.
The use of ports 0 as digital outputs is not allowed when the 12-Bit A/D converter is used.
The characteristics apply when AVCC0, AVSS0, VREFH0, VREFH, VREFL0, VREFL, and 12-bit A/D converter input voltage is stable.

Note 1. The conversion time includes the sampling and comparison times. The number of sampling states is indicated for the test conditions.

Note 2. Values in parentheses indicate the sampling time.

Table 60.75 A/D conversion characteristics for unit 1 (External VDD mode) (1 of 2)

Conditions: PCLKC = 1 to 60 MHz
AVCC0 = 2.7 to 3.6 V, VREFH = 2.7 to 3.6 V

Parameter	Min	Typ	Max	Unit	Test conditions
Frequency	1	—	60	MHz	—
Analog input capacitance	—	—	30	pF	—
Quantization error	—	±0.5	—	LSB	—
Resolution	—	—	12	Bits	—

Table 60.75 A/D conversion characteristics for unit 1 (External VDD mode) (2 of 2)

Conditions: PCLKC = 1 to 60 MHz
AVCC0 = 2.7 to 3.6 V, VREFH = 2.7 to 3.6 V

Parameter			Min	Typ	Max	Unit	Test conditions
High-precision channels (AN100 to AN102, AN104 to AN106)	Conversion time*1 (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	0.48 (0.267)*2	—	—	µs	Sampling in 16 states
		Max. = 400 Ω	0.40 (0.183)*2	—	—	µs	Sampling in 11 states AVCC0 = 3.0 to 3.6 V 3.0 V ≤ VREFH ≤ AVCC0
	Offset error		—	±1.0	±2.5	LSB	—
	Full-scale error		—	±1.0	±3.5	LSB	—
	Absolute accuracy		—	±2.0	±4.5	LSB	—
	DNL differential nonlinearity error		—	±0.5	±1.5	LSB	—
	INL integral nonlinearity error		—	±1.0	±2.5	LSB	—
Normal-precision channels (AN116 to AN122)	Conversion time*1 (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	0.88 (0.667)*2	—	—	µs	Sampling in 40 states
	Offset error		—	±1.0	±5.5	LSB	—
	Full-scale error		—	±1.0	±5.5	LSB	—
	Absolute accuracy		—	±2.0	±7.5	LSB	—
	DNL differential nonlinearity error		—	±0.5	±4.5	LSB	—
	INL integral nonlinearity error		—	±1.0	±5.5	LSB	—

Note: These specification values apply when only one A/D is operating and D/A and ACMPHS are not operating and there is no access to the external bus during A/D conversion.
If other A/D, D/A, or ACMPHS is operating or bus access occurs during A/D conversion, values might not fall within the indicated ranges.
The use of ports 0 as digital outputs is not allowed when the 12-Bit A/D converter is used.
The characteristics apply when AVCC0, AVSS0, VREFH0, VREFH, VREFL0, VREFL, and 12-bit A/D converter input voltage are stable.

Note 1. The conversion time includes the sampling and comparison times. The number of sampling states is indicated for the test conditions.

Note 2. Values in parentheses indicate the sampling time.

Table 60.76 A/D internal reference voltage characteristics

Parameter	Min	Typ	Max	Unit	Test conditions
A/D internal reference voltage	1.13	1.18	1.28	V	—
Sampling time	4.15	—	—	µs	—

For the characteristics of VBATT 1/3 voltage monitor, see section 60.11. VBATT Characteristics.

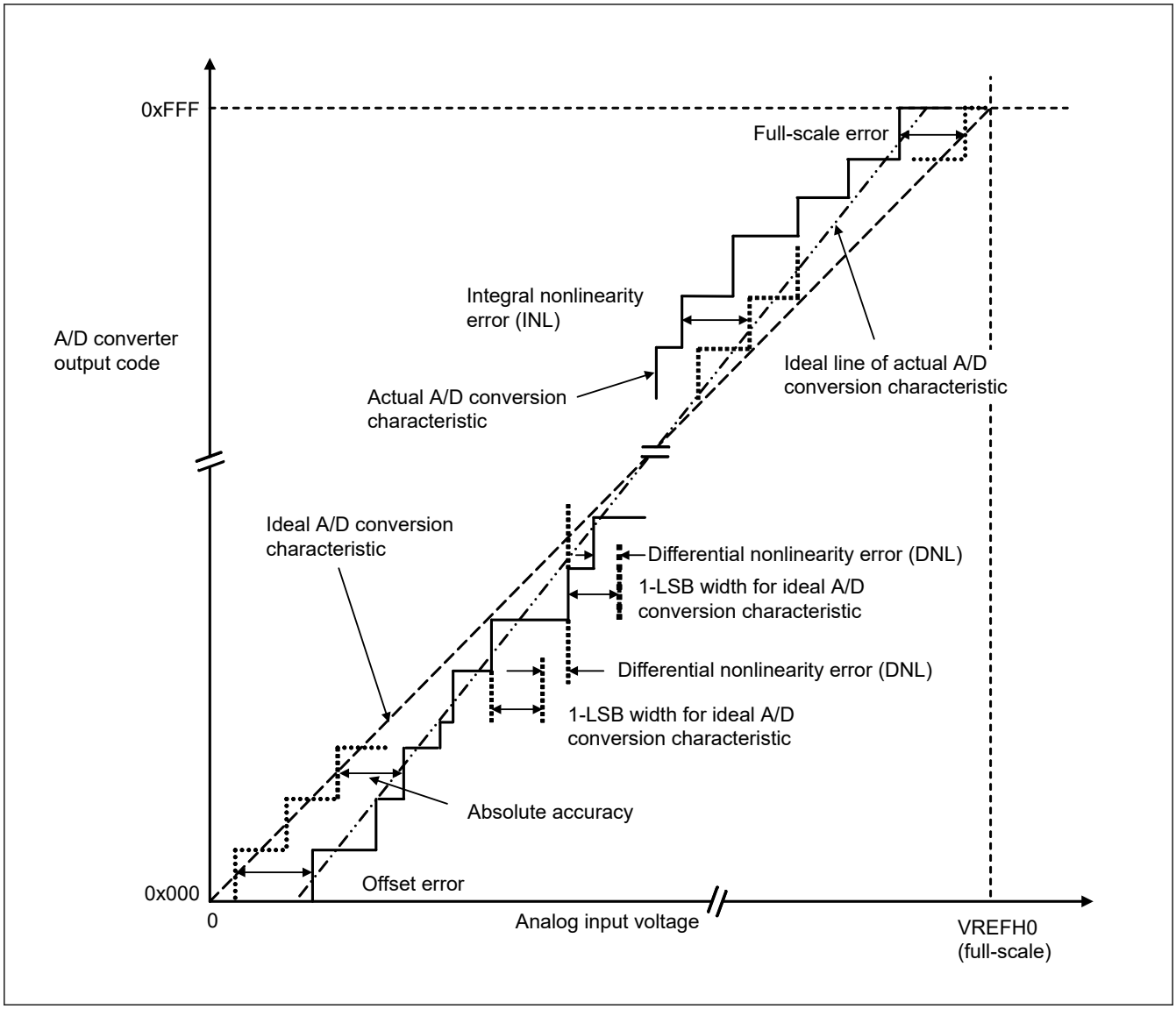


Figure 60.119 Illustration of ADC12 characteristic terms

Absolute accuracy

Absolute accuracy is the difference between output code based on the theoretical A/D conversion characteristics, and the actual A/D conversion result. When measuring absolute accuracy, the voltage at the midpoint of the width of the analog input voltage (1-LSB width), which can meet the expectation of outputting an equal code based on the theoretical A/D conversion characteristics, is used as an analog input voltage. For example, if 12-bit resolution is used and the reference voltage VREFH0 = 3.072 V, then the 1-LSB width becomes 0.75 mV, and 0 mV, 0.75 mV, and 1.5 mV are used as the analog input voltages. If the analog input voltage is 6 mV, an absolute accuracy of ±5 LSB means that the actual A/D conversion result is in the range of 0x003 to 0x00D, though an output code of 0x008 can be expected from the theoretical A/D conversion characteristics.

Integral nonlinearity error (INL)

Integral nonlinearity error is the maximum deviation between the ideal line when the measured offset and full-scale errors are zeroed, and the actual output code.

Differential nonlinearity error (DNL)

Differential nonlinearity error is the difference between the 1-LSB width based on the ideal A/D conversion characteristics and the width of the actual output code.

Offset error

Offset error is the difference between the transition point of the ideal first output code and the actual first output code.

Full-scale error

Full-scale error is the difference between the transition point of the ideal last output code and the actual last output code.

Table 60.77 A/D input voltage range for each input channel

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Input voltage range	Unit 0	AN000 to AN002, AN004 to AN008	VI	VREFL0	—	VREFH0	V	—
		AN016 to AN018		VREFL0	—	VREFH0		VREFH0 ≤ VCC
				VREFL0	—	VCC		VCC < VREFH0
		AN019		VREFL0	—	VREFH0		VREFH0 ≤ VCC2
				VREFL0	—	VCC2		VCC2 < VREFH0
	Unit 1	AN100 to AN102, AN104 to AN106		VREFL	—	VREFH		—
		AN116 to AN118		VREFL	—	VREFH		VREFH ≤ VCC
				VREFL	—	VCC		VCC < VREFH
		AN119 to AN122		VREFL	—	VREFH		VREFH ≤ VCC2
				VREFL	—	VCC2		VCC2 < VREFH

60.7 DAC12 Characteristics

Table 60.78 D/A conversion characteristics (1 of 2)

Parameter		Min	Typ	Max	Unit	Test conditions
Resolution		—	—	12	Bits	—
Without output amplifier (for pin output, AVCC0 ≥ 1.65V)						
Absolute accuracy	VREFH ≥ 2.7V	—	—	±24	LSB	Resistive load 2 MΩ
	VREFH < 2.7V	—	—	±36		
INL	VREFH ≥ 2.7V	—	±2.0	±8.0	LSB	Resistive load 2 MΩ
	VREFH < 2.7V	—	±2.0	±8.0		
DNL	VREFH ≥ 2.7V	—	±1.0	±2.0	LSB	—
	VREFH < 2.7V		±1.0	±3.0		
Output impedance		—	8.5	—	kΩ	—
Conversion time	VREFH ≥ 2.7V	—	—	3.0	µs	Resistive load 2 MΩ, Capacitive load 20 pF
	VREFH < 2.7V	—	—	6.0		
Output voltage range		0	—	VREFH	V	—
Without output amplifier (for internal output, AVCC0 ≥ 1.65V)						
Absolute accuracy	VREFH≥2.7V	—	—	±4.0	LSB	—
	VREFH < 2.7V	—	—	±6.0		
Conversion time	VREFH ≥ 2.7V	—	—	3.0	µs	—
	VREFH < 2.7V	—	—	6.0		

Table 60.78 D/A conversion characteristics (2 of 2)

Parameter		Min	Typ	Max	Unit	Test conditions
Output voltage range		0	—	VREFH	V	—
With output amplifier (AVCC0 ≥ 2.70V)						
INL		—	±2.0	±4.0	LSB	—
DNL		—	±1.0	±2.0	LSB	—
Conversion time		—	—	3.5	µs	—
Resistive load		5	—	—	kΩ	—
Capacitive load		—	—	50	pF	—
Output voltage range	VREFH ≥ 2.7V	0.20	—	VREFH – 0.20	V	—
	VREFH < 2.7V	0.22	—	VREFH – 0.22		—

60.8 TSN Characteristics

Table 60.79 TSN characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Relative accuracy	—	—	± 1.0	—	°C	—
Temperature slope	—	—	4.0	—	mV/°C	—
Output voltage (at 25 °C)	—	—	1.24	—	V	—
Temperature sensor start time	t_{START}	—	—	30	µs	—
Sampling time	—	4.15	—	—	µs	—

60.9 OSC Stop Detect Characteristics

Table 60.80 Oscillation stop detection circuit characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Detection time	t_{dr}	—	—	1	ms	Figure 60.120

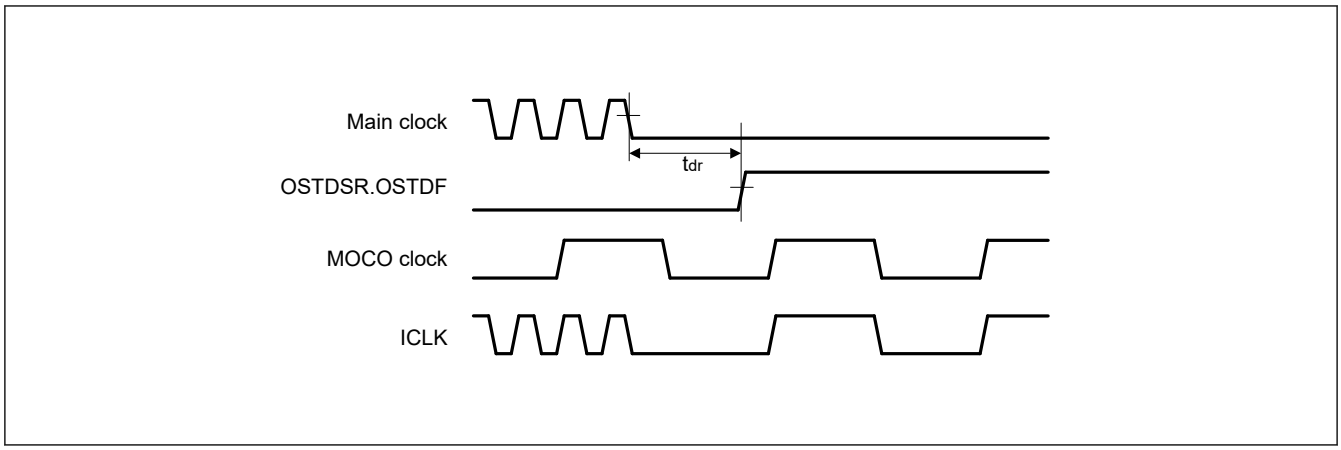


Figure 60.120 Oscillation stop detection timing

60.10 POR and PVD Characteristics

Table 60.81 Power-on reset circuit and voltage detection circuit characteristics (1 of 2)

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Voltage detection level	Power-on reset (POR)	Tj = 25℃	V_{POR1}	1.55	1.60	1.68	V	Figure 60.121
		Tj = 125℃		1.55	1.60	1.70		
		Tj = 25℃	V_{POR2}	1.65	1.70	1.79		
		Tj = 125℃		1.65	1.70	1.81		
	Voltage detection circuit (PVD0)		V_{det0_0}	2.76	2.85	2.99		Figure 60.122
			V_{det0_1}	2.50	2.58	2.71		
			V_{det0_2}	2.08	2.15	2.27		
			V_{det0_3}	1.94	2.00	2.12		
			V_{det0_4}	1.84	1.90	2.01		
			V_{det0_5}	1.74	1.80	1.91		
			V_{det0_6}	1.65	1.70	1.81		
			V_{det0_7}	1.55	1.60	1.70		
	Voltage detection circuit (PVDn) (n = 1, 2)		$V_{detn\_3\_rise}$	3.78	3.92	4.10		Figure 60.123
			$V_{detn\_3\_fall}$	3.72	3.86	4.04		
			$V_{detn\_4\_rise}$	3.09	3.20	3.35		
			$V_{detn\_4\_fall}$	3.03	3.14	3.29		
			$V_{detn\_5\_rise}$	3.05	3.16	3.31		
			$V_{detn\_5\_fall}$	2.99	3.10	3.25		
			$V_{detn\_6\_rise}$	3.03	3.14	3.29		
			$V_{detn\_6\_fall}$	2.97	3.08	3.23		
			$V_{detn\_7\_rise}$	2.81	2.91	3.05		
			$V_{detn\_7\_fall}$	2.75	2.85	2.99		
			$V_{detn\_8\_rise}$	2.79	2.89	3.03		
			$V_{detn\_8\_fall}$	2.73	2.83	2.97		
			$V_{detn\_9\_rise}$	2.76	2.86	3.00		
			$V_{detn\_9\_fall}$	2.70	2.80	2.94		
			$V_{detn\_10\_rise}$	2.58	2.67	2.80		
			$V_{detn\_10\_fall}$	2.53	2.62	2.75		
			$V_{detn\_11\_rise}$	2.30	2.38	2.51		
			$V_{detn\_11\_fall}$	2.25	2.33	2.46		
Voltage detection level	Voltage detection circuit (PVDn) (n = 1, 2)		$V_{detn\_12\_rise}$	1.88	1.94	2.05	V	Figure 60.123
			$V_{detn\_12\_fall}$	1.84	1.90	2.01		
			$V_{detn\_13\_rise}$	1.84	1.90	2.01		
			$V_{detn\_13\_fall}$	1.80	1.86	1.97		
			$V_{detn\_14\_rise}$	1.72	1.78	1.89		
			$V_{detn\_14\_fall}$	1.68	1.74	1.85		
			$V_{detn\_15\_rise}$	1.69	1.75	1.85		
			$V_{detn\_15\_fall}$	1.65	1.71	1.81		

Table 60.81 Power-on reset circuit and voltage detection circuit characteristics (2 of 2)

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Internal reset time*1	Power-on reset time		t_{POR1}	—	—	8.2	ms	Figure 60.121
			t_{POR2}	—	—	4.5		Figure 60.121
	PVD0 reset time		t_{PVD0}	—	—	*1		Figure 60.121
	PVD1 reset time		t_{PVD1}	—	—	*1		Figure 60.122
	PVD2 reset time		t_{PVD2}	—	—	*1		
Minimum VCC down time (POR)*2	100mV < VD		t_{VOFFP}	500	—	—	µs	Figure 60.121
	50mV < VD ≤ 100mV			900	—	—		
	VD ≤ 50mV			2000	—	—		
Minimum VCC down time (PVD)*2	PVD0 (OFS1(_SEC).PVDLPSEL = 0 in Deep Software Standby mode 1, 2)		t_{VOFF}	400	—	—	µs	Figure 60.122
	PVD0 (Other than above), PVD1, PVD2			200	—	—	µs	Figure 60.122
Response delay time (POR)	100mV < VD		t_{detp}	—	—	500	µs	Figure 60.121
	50mV < VD ≤ 100mV			—	—	900		
	VD ≤ 50mV			—	—	2000		
Response delay time (PVD)	PVD0 (OFS1(_SEC).PVDLPSEL = 0 in Deep Software Standby mode 1, 2)	50mV < VD	t_{det}	—	—	200	µs	Figure 60.122, Figure 60.123
		50mV ≥ VD		—	—	400		
	PVD0 (Other than above), PVD1, PVD2	100mV < VD		—	—	10		
		100mV ≥ VD		—	—	200		
PVD operation stabilization time (after PVD is enabled)			$T_{d\ (E-A)}$	—	—	20	µs	Figure 60.123

Note 1. The maximum value of t_{PVD0} is equal to t_{DSBY} because the internal reset time is maximized when returning from Deep Software Standby mode.
The maximum value of t_{PVD1}, t_{PVD2} are equal to t_{DSTBY} because the internal reset time is maximized when returning from Deep Software Standby mode.

Note 2. The minimum VCC down time indicates the time when VCC is below the minimum value of voltage detection levels V_{POR1}, V_{det1}, and V_{det2} for the POR / PVD.

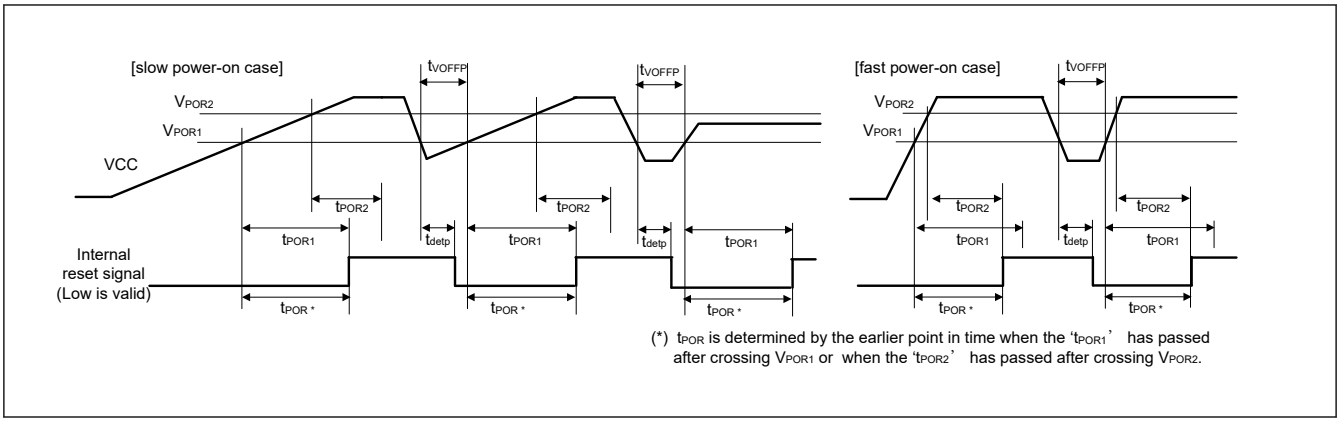


Figure 60.121 Power-on reset timing

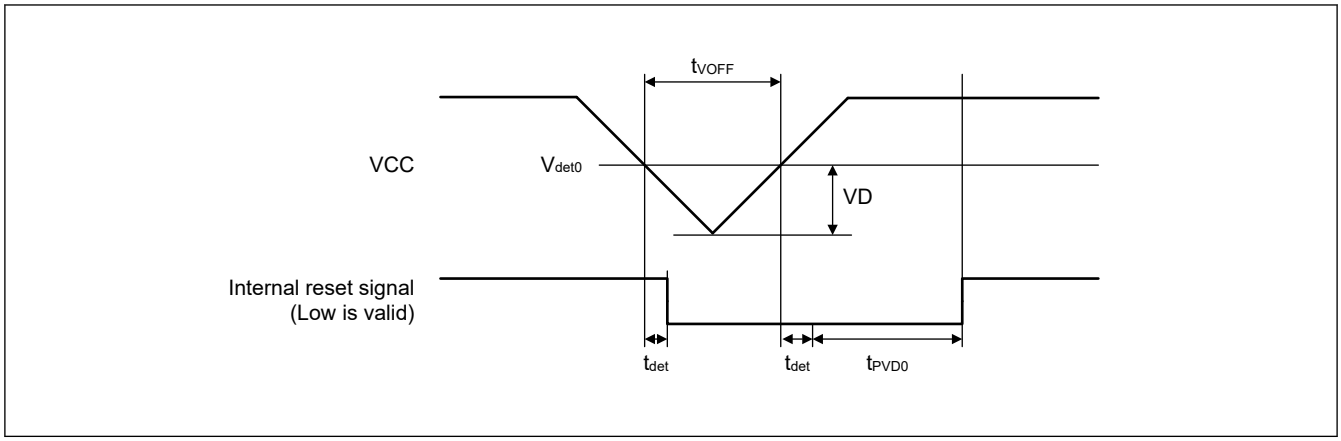


Figure 60.122 Voltage detection circuit timing (V_{det0})

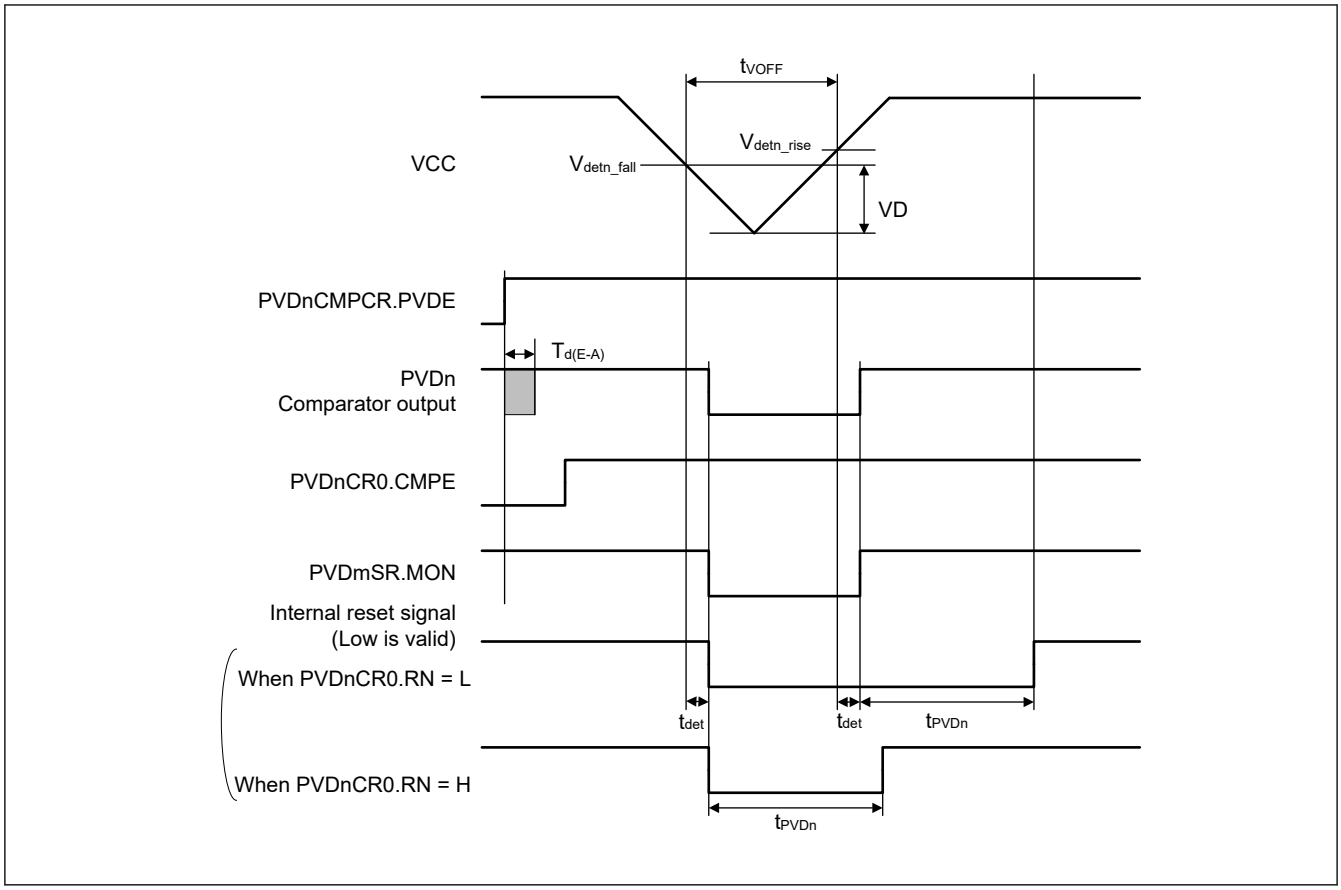


Figure 60.123 Voltage detection circuit timing (V_{detn}) (n = 1, 2)

60.11 VBATT Characteristics

Table 60.82 Battery backup function characteristics

Conditions: VCC = VCC_DCDC = VCC_USB = 1.68 to 3.6 V, VBATT = 1.62 to 3.6 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Voltage level for switching to battery backup OFS1(_SEC).PVDAS and PVDLPSEL are 0 in Deep Software Standby mode 1, 2 (VDETVATT_n follows VDSEL[2:0] setting for PVD0)	$V_{DETBATT\_0}$	2.760	2.850	2.990	V	Figure 60.124
	$V_{DETBATT\_1}$	2.500	2.580	2.710		
	$V_{DETBATT\_2}$	2.080	2.150	2.270		
	$V_{DETBATT\_3}$	1.940	2.000	2.120		
	$V_{DETBATT\_4}$	1.840	1.900	2.010		
	$V_{DETBATT\_5}$	1.740	1.800	1.910		
	$V_{DETBATT\_6}$	1.650	1.700	1.810		
Voltage level for switching to battery backup (Other than above)	$V_{DETBATT\_0}$	2.710	2.800	2.940	V	
	$V_{DETBATT\_1}$	2.450	2.530	2.660		
	$V_{DETBATT\_2}$	2.030	2.100	2.220		
	$V_{DETBATT\_3}$	1.855	1.950	2.065		
	$V_{DETBATT\_4}$	1.790	1.850	1.960		
	$V_{DETBATT\_5}$	1.690	1.750	1.860		
VCC drop detection stabilization wait time*2	t_{DETWT}	—	—	20	µs	
Lower-limit VBATT voltage for power supply switching caused by VCC voltage drop	V_{BATTSW}	2.0	—	—	V	Figure 60.124
VCC-off period for starting power supply switching*1 (OFS1(_SEC).PVDAS and PVDLPSEL are 0 in Deep Software Standby mode 1, 2)	$t_{VOFFBATT}$	400	—	—	µs	
VCC-off period for starting power supply switching*1 (Other than above)		200	—	—		
Backup domain power-down detection level	$V_{PDR\ (BATR)}$	1.45	1.50	1.60	V	Figure 60.125
Time delay in assertion of the reset signal for the backup domain*3	$t_{p\ (PDRL)}$	—	—	2000	µs	
Time delay in negation of the reset signal for the backup domain	$t_{p\ (PDRH)}$	—	—	3000		
VBATT monitor operation stabilization time (after VBATTMNSELR.VBTMNSEL is changed to 1)	t_{MONWT}	—	—	4.2	µs	—
VBATT voltage monitor level	$V_{MONBATT}$	—	VBATT / 3	—	V	—
VBATT current increase (when VBATTMNSELR.VBTMNSEL is 1 compared to the case that VBATTMNSELR.VBTMNSEL is 0)	$I_{VBATTSELB}$	—	1.50	2.35	µA	—
VCC current increase (when VBATTMNSELR.VBTMNSEL is 1 compared to the case that VBATTMNSELR.VBTMNSEL is 0)	$I_{VBATTSELC}$	—	330	577	µA	—

Note 1. The VCC-off period for starting power supply switching indicates the period in which VCC is below the minimum value of the voltage level for switching to battery backup ($V_{DETBATT}$).
In addition, this period indicates the time tVOFFP when VCC is below the minimum value of voltage detection levels V_{POR1}.
Note 2. Stable time when VBTBPCR2.VDETLVL is changed or VBTBPCR1.BPWSWSTP is changed from 1 to 0.
Note 3. When the VBATT_R recovers within this period, the backup domain reset signal may not be generated.

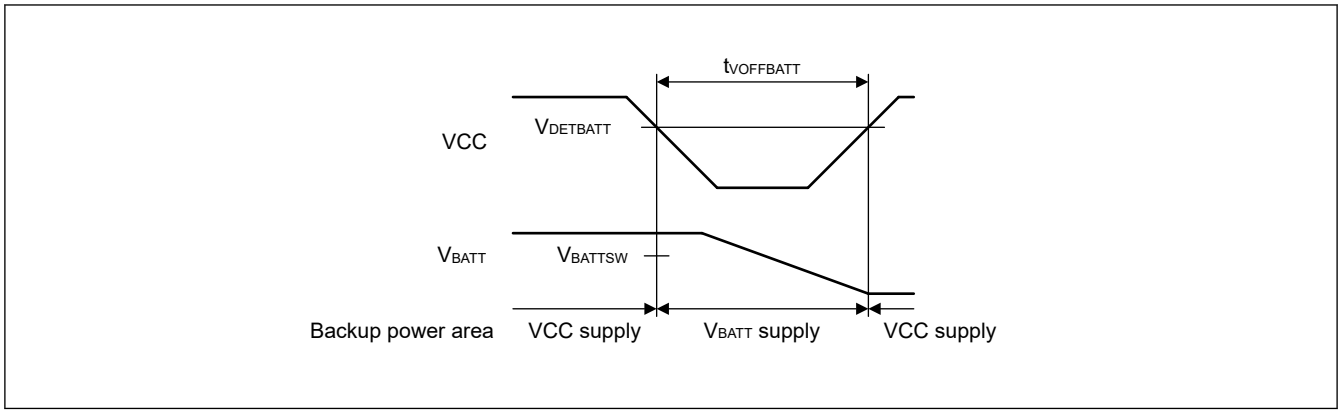


Figure 60.124 Battery backup function characteristics

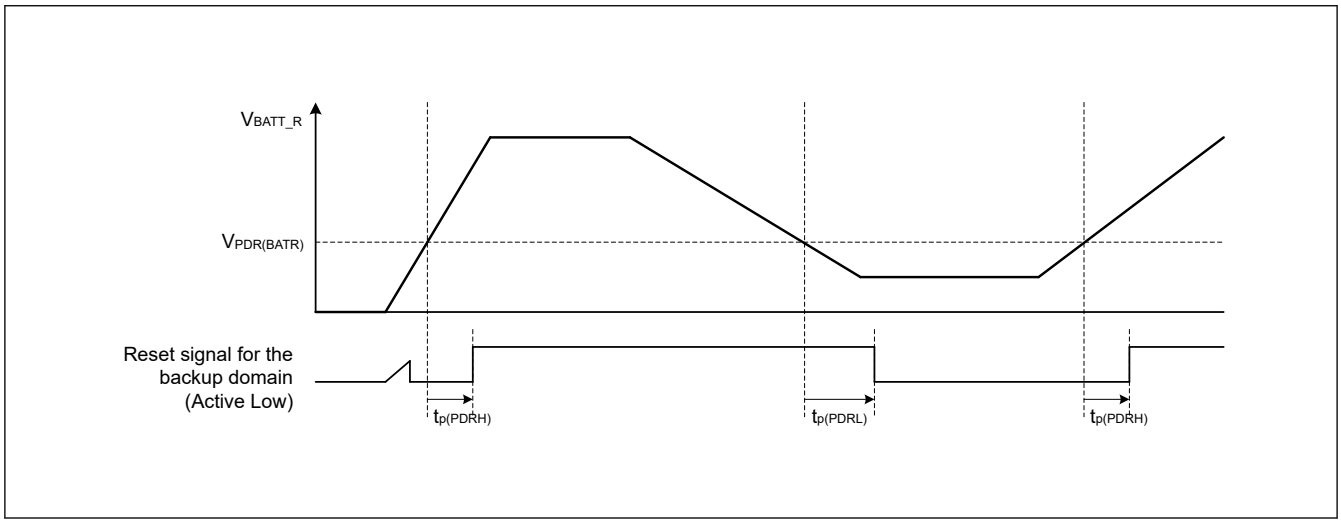


Figure 60.125 Backup Domain Reset Characteristics

60.12 ACMPHS Characteristics

Table 60.83 ACMPHS

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Reference voltage range			VREF	0	—	AVCC0	V	—
Input voltage range	ACMPHS0		VI	0	—	AVCC0	V	—
	ACMPHS1	IVCMP1 to IVCMP3		0	—	AVCC0		—
		IVCMP0		0	—	AVCC0		VCC >= AVCC0
				0	—	VCC		VCC < AVCC0
Output delay*1			Td	—	50	100	ns	VI = VREF ± 100mV
Internal reference voltage			Vref	1.13	1.18	1.28	V	—

Note 1. This value is the internal propagation delay.

60.13 Flash Memory Characteristics

60.13.1 Code Flash Memory Characteristics

Table 60.84 Code flash memory characteristics

Conditions: Program or erase: FCLK = 4 to 60 MHz
Read: FCLK ≤ 60 MHz

Parameter		Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
			Min	Typ*6	Max	Min	Typ*6	Max		
Programming time N_{PEC} ≤ 100 times	128-byte	t_{P128}	—	0.75	13.2	—	0.34	6.0	ms	
	8-KB	t_{P8K}	—	49	176	—	22	80	ms	
	32-KB	t_{P32K}	—	194	704	—	88	320	ms	
Programming time N_{PEC} > 100 times	128-byte	t_{P128}	—	0.91	15.8	—	0.41	7.2	ms	
	8-KB	t_{P8K}	—	60	212	—	27	96	ms	
	32-KB	t_{P32K}	—	234	848	—	106	384	ms	
Erasure time N_{PEC} ≤ 100 times	8-KB	t_{E8K}	—	78	216	—	43	120	ms	
	32-KB	t_{E32K}	—	283	864	—	157	480	ms	
Erasure time N_{PEC} > 100 times	8-KB	t_{E8K}	—	94	260	—	52	144	ms	
	32-KB	t_{E32K}	—	341	1040	—	189	576	ms	
Reprogramming/erasure cycle*4		N_{PEC}	10000*1	—	—	10000*1	—	—	Times	
Suspend delay during programming		t_{SPD}	—	—	264	—	—	120	µs	
Programming resume time		t_{PRT}	—	—	110	—	—	50	µs	
First suspend delay during erasure in suspend priority mode		t_{SESD1}	—	—	216	—	—	120	µs	
Second suspend delay during erasure in suspend priority mode		t_{SESD2}	—	—	1.7	—	—	1.7	ms	
Suspend delay during erasure in erasure priority mode		t_{SEED}	—	—	1.7	—	—	1.7	ms	
First erasing resume time during erasure in suspend priority mode*5		t_{REST1}	—	—	1.7	—	—	1.7	ms	
Second erasing resume time during erasure in suspend priority mode		t_{REST2}	—	—	144	—	—	80	µs	
Erasing resume time during erasure in erasure priority mode		t_{REET}	—	—	144	—	—	80	µs	
Forced stop command		t_{FD}	—	—	32	—	—	20	µs	
Data hold time*2		t_{DRP}	10*2 *3	—	—	10*2 *3	—	—	Years	Tj = +125°C
			20*2 *3	—	—	20*2 *3	—	—		Tj = +105°C
			30*2 *3	—	—	30*2 *3	—	—		Tj = +85°C

Note 1. This is the minimum number of times to guarantee all the characteristics after reprogramming. The guaranteed range is from 1 to the minimum value.
Note 2. This indicates the minimum value of the characteristic when reprogramming is performed within the specified range.
Note 3. This result is obtained from reliability testing.
Note 4. The reprogram/erase cycle is the number of erasures for each block. When the reprogram/erase cycle is n times (n = 10,000), erasing can be performed n times for each block. For example, when 128-byte programming is performed 64 times for different addresses in 8-KB blocks, and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address several times as one erasure is not enabled. Overwriting is prohibited.
Note 5. Time for resumption includes time for reapplying the erasing pulse (up to one full pulse) that was cut off at the time of suspension.
Note 6. The reference value at VCC = 3.3V and room temperature.

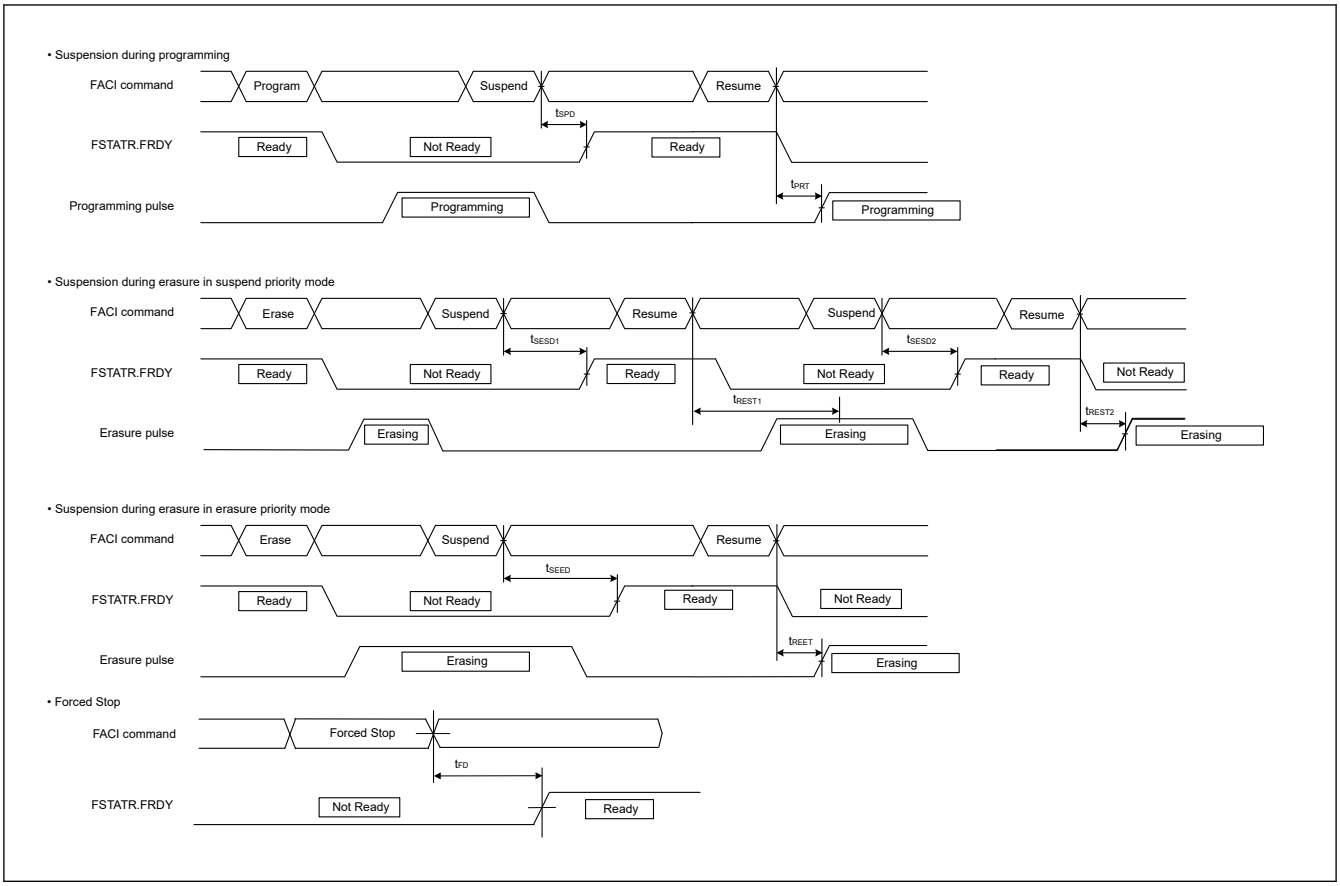


Figure 60.126 Suspension and forced stop timing for flash memory programming and erasure

60.13.2 Data Flash Memory Characteristics

Table 60.85 Data flash memory characteristics (1 of 2)

Conditions: Program or erase: FCLK = 4 to 60 MHz
Read: FCLK ≤ 60 MHz

Parameter		Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
			Min	Typ*6	Max	Min	Typ*6	Max		
Programming time	4-byte	t_{DP4}	—	0.36	3.8	—	0.16	1.7	ms	
	8-byte	t_{DP8}	—	0.38	4.0	—	0.17	1.8		
	16-byte	t_{DP16}	—	0.42	4.5	—	0.19	2.0		
Erasure time	64-byte	t_{DE64}	—	3.1	18	—	1.7	10	ms	
	128-byte	t_{DE128}	—	4.7	27	—	2.6	15		
	256-byte	t_{DE256}	—	8.9	50	—	4.9	28		
Blank check time	4-byte	t_{DBC4}	—	—	84	—	—	30	µs	
Reprogramming/erasure cycle*1		N_{DPEC}	125000*2	—	—	125000*2	—	—	—	
Suspend delay during programming	4-byte	t_{DSPD}	—	—	264	—	—	120	µs	
	8-byte		—	—	264	—	—	120		
	16-byte		—	—	264	—	—	120		
Programming resume time		t_{DPRT}	—	—	110	—	—	50	µs	

Table 60.85 Data flash memory characteristics (2 of 2)

Conditions: Program or erase: FCLK = 4 to 60 MHz
Read: FCLK ≤ 60 MHz

Parameter		Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
			Min	Typ*6	Max	Min	Typ*6	Max		
First suspend delay during erasure in suspend priority mode	64-byte	t_{DSESD1}	—	—	216	—	—	120	µs	
	128-byte		—	—	216	—	—	120		
	256-byte		—	—	216	—	—	120		
Second suspend delay during erasure in suspend priority mode	64-byte	t_{DSESD2}	—	—	300	—	—	300	µs	
	128-byte		—	—	390	—	—	390		
	256-byte		—	—	570	—	—	570		
Suspend delay during erasing in erasure priority mode	64-byte	t_{DSEED}	—	—	300	—	—	300	µs	
	128-byte		—	—	390	—	—	390		
	256-byte		—	—	570	—	—	570		
First erasing resume time during erasure in suspend priority mode*5		t_{DREST1}	—	—	300	—	—	300	µs	
Second erasing resume time during erasure in suspend priority modeFirst erasing resume time during erasure in suspend priority mode		t_{DREST2}	—	—	126	—	—	70	µs	
Erasing resume time during erasure in erasure priority mode		t_{DREET}	—	—	126	—	—	70	µs	
Forced stop command		t_{FD}	—	—	32	—	—	20	µs	
Data hold time*3		t_{DRP}	10*3 *4	—	—	10*3 *4	—	—	Year	Tj = +125°C
			20*3 *4	—	—	20*3 *4	—	—		Tj = +105°C
			30*3 *4	—	—	30*3 *4	—	—		Tj = +85°C

Note 1. The reprogram/erase cycle is the number of erasures for each block. When the reprogram/erase cycle is n times (n = 125,000), erasing can be performed n times for each block. For example, when 4-byte programming is performed 16 times for different addresses in 64-byte blocks, and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address several times as one erasure is not enabled. Overwriting is prohibited.
Note 2. This is the minimum number of times to guarantee all the characteristics after reprogramming. The guaranteed range is from 1 to the minimum value.
Note 3. This indicates the minimum value of the characteristic when reprogramming is performed within the specified range.
Note 4. This result is obtained from reliability testing.
Note 5. Time for resumption includes time for reapplying the erasing pulse (up to one full pulse) that was cut off at the time of suspension.
Note 6. The reference value at VCC = 3.3 V and room temperature.

60.13.3 Option Setting Memory (Code flash memory) Characteristics

Table 60.86 Option setting memory (Code flash memory) characteristics (1 of 2)

Conditions: Program: FCLK = 4 to 60 MHz
Read: FCLK ≤ 60 MHz

Parameter	Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
		Min	Typ*4	Max	Min	Typ*4	Max		
Programming time N_{OPC} ≤ 200 times	t_{OP}	—	83	309	—	45	162	ms	

Table 60.86 Option setting memory (Code flash memory) characteristics (2 of 2)

Conditions: Program: FCLK = 4 to 60 MHz
Read: FCLK ≤ 60 MHz

Parameter	Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
		Min	Typ*4	Max	Min	Typ*4	Max		
Programming time N_{OPC} > 200 times	t_{OP}	—	100	371	—	55	195	ms	
Reprogramming cycle	N_{OPC}	20000*1	—	—	20000*1	—	—	Times	
Data hold time*2	t_{DRP}	10*2 *3	—	—	10*2 *3	—	—	Years	Tj = +125°C
		20*2 *3	—	—	20*2 *3	—	—		Tj = +105°C
		30*2 *3	—	—	30*2 *3	—	—		Tj = +85°C

Note 1. This is the minimum number of times to guarantee all the characteristics after reprogramming. The guaranteed range is from 1 to the minimum value.
Note 2. This indicates the minimum value of the characteristic when reprogramming is performed within the specified range.
Note 3. This result is obtained from reliability testing.
Note 4. The reference value at VCC = 3.3 V and room temperature.

60.13.4 Option Setting Memory (Data flash memory) Characteristics

Table 60.87 Option Setting Memory (Data flash memory) characteristics

Conditions: Program: FCLK = 4 to 60 MHz
Read: FCLK ≤ 60 MHz

Parameter	Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
		Min	Typ*4	Max	Min	Typ*4	Max		
Command time for configuration set (4 / 16 Byte)	t_{DCCT}	—	68	515	—	35	255	ms	
Update Cycles in Configuration area	N_{cupc}	125000 *1	—	—	125000 *1	—	—	Times	
Data hold time*2	t_{DRP}	10*2 *3	—	—	10*2 *3	—	—	Years	Tj = +125°C
		20*2 *3	—	—	20*2 *3	—	—		Tj = +105°C
		30*2 *3	—	—	30*2 *3	—	—		Tj = +85°C

Note 1. This is the minimum number of times to guarantee all the characteristics after reprogramming. The guaranteed range is from 1 to the minimum value.
Note 2. This indicates the minimum value of the characteristic when reprogramming is performed within the specified range.
Note 3. This result is obtained from reliability testing.
Note 4. The reference value at VCC = 3.3 V and room temperature.

60.13.5 Anti-rollback counter Characteristics

Table 60.88 Anti-rollback counter characteristics (1 of 2)

Conditions: Program: FCLK = 4 to 60 MHz
Read: FCLK ≤ 60 MHz

Parameter	Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
		Min	Typ*4	Max	Min	Typ*4	Max		
Command time for increment counter and refresh counter	t_{IRCT}	—	11.9	81	—	6.3	42	ms	
Command time for read counter	t_{RCT}	—	—	25	—	—	5	µs	
Update Cycles (total of increment and refreshing)	N_{cupc}	125000 *1	—	—	125000 *1	—	—	Times	

Table 60.88 Anti-rollback counter characteristics (2 of 2)

Conditions: Program: FCLK = 4 to 60 MHz
Read: FCLK ≤ 60 MHz

Parameter	Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
		Min	Typ*4	Max	Min	Typ*4	Max		
Data hold time*2	t_{DRP}	10*2 *3	—	—	10*2 *3	—	—	Years	Tj = +125°C
		20*2 *3	—	—	20*2 *3	—	—		Tj = +105°C
		30*2 *3	—	—	30*2 *3	—	—		Tj = +85°C

Note 1. This is the minimum number of times to guarantee all the characteristics after reprogramming. The guaranteed range is from 1 to the minimum value.
Note 2. This indicates the minimum value of the characteristic when reprogramming is performed within the specified range.
Note 3. This result is obtained from reliability testing.
Note 4. The reference value at VCC = 3.3 V and room temperature.

60.14 Boundary Scan

Table 60.89 Boundary scan characteristics

Parameter	VCC	Symbol	Min	Typ	Max	Unit	Test conditions
TCK clock cycle time	1.68 V or above	t_{TCKcyc}	100	—	—	ns	Figure 60.127
TCK clock high pulse width	1.68 V or above	t_{TCKH}	0.45	—	—	t_{TCKcyc}	
TCK clock low pulse width	1.68 V or above	t_{TCKL}	0.45	—	—	t_{TCKcyc}	
TCK clock rise time	1.68 V or above	t_{TCKr}	—	—	0.05*2	t_{TCKcyc}	
TCK clock fall time	1.68 V or above	t_{TCKf}	—	—	0.05*2	t_{TCKcyc}	
TMS setup time	1.68 V or above	t_{TMSS}	20	—	—	ns	Figure 60.128
TMS hold time	1.68 V or above	t_{TMSH}	20	—	—	ns	
TDI setup time	1.68 V or above	t_{TDIS}	20	—	—	ns	
TDI hold time	1.68 V or above	t_{TDIH}	20	—	—	ns	
TDO data delay	1.68 V or above	t_{TDOD}	—	—	40	ns	
Capture register setup time	1.68 V or above	t_{CAPTS}	20	—	—	ns	Figure 60.129
Capture register hold time	1.68 V or above	t_{CAPTH}	20	—	—	ns	
Update register delay time	1.68 V or above	$t_{UPDATED}$	—	—	40	ns	
Boundary scan circuit startup time*1	1.68 V or above	T_{BSSTUP}	t_{RESWP}	—	—	—	Figure 60.130

Note 1. Boundary scan does not function until the power-on reset becomes negative.
Note 2. 1 µs at the longest

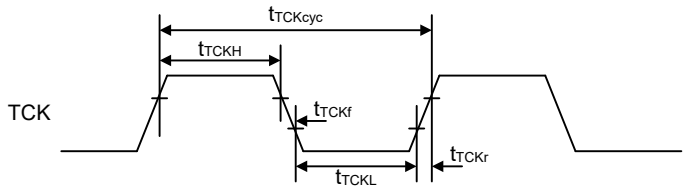


Figure 60.127 Boundary scan TCK timing

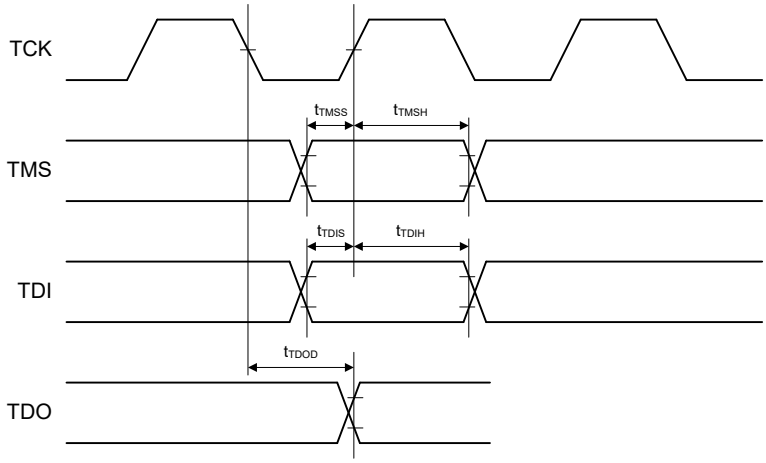


Figure 60.128 Boundary scan input/output timing (1)

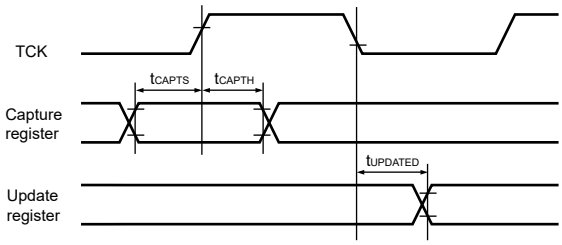


Figure 60.129 Boundary scan input/output timing (2)

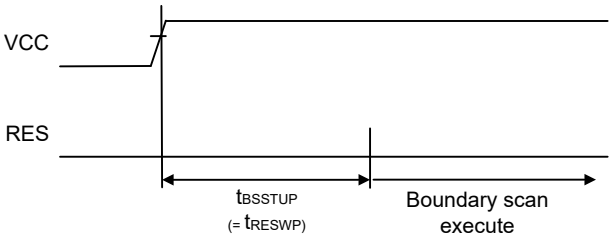


Figure 60.130 Boundary scan circuit startup timing

60.15 Joint European Test Action Group (JTAG)

Table 60.90 JTAG

Parameter	VCC	Symbol	Min	Typ	Max	Unit	Test conditions
TCK clock cycle time	2.7 V or above	t_{TCKcyc}	40.0	—	—	ns	Figure 60.131
	1.68 V or above		40.0	—	—	ns	
TCK clock high pulse width	2.7 V or above	t_{TCKH}	0.375	—	—	t_{TCKcyc}	
	1.68 V or above		0.375	—	—	t_{TCKcyc}	
TCK clock low pulse width	2.7 V or above	t_{TCKL}	0.375	—	—	t_{TCKcyc}	
	1.68 V or above		0.375	—	—	t_{TCKcyc}	
TCK clock rise time	2.7 V or above	t_{TCKr}	—	—	0.125*1	t_{TCKcyc}	
	1.68 V or above		—	—	0.125*1	t_{TCKcyc}	
TCK clock fall time	2.7 V or above	t_{TCKf}	—	—	0.125*1	t_{TCKcyc}	
	1.68 V or above		—	—	0.125*1	t_{TCKcyc}	
TMS setup time	2.7 V or above	t_{TMSS}	8.0	—	—	ns	Figure 60.132
	1.68 V or above		8.0	—	—	ns	
TMS hold time	2.7 V or above	t_{TMSH}	8.0	—	—	ns	
	1.68 V or above		8.0	—	—	ns	
TDI setup time	2.7 V or above	t_{TDIS}	8.0	—	—	ns	
	1.68 V or above		8.0	—	—	ns	
TDI hold time	2.7 V or above	t_{TDIH}	8.0	—	—	ns	
	1.68 V or above		8.0	—	—	ns	
TDO data delay time	2.7 V or above	t_{TDOD}	—	—	20.0	ns	
	1.68 V or above		—	—	28.0	ns	

Note 1. 1 µs at the longest

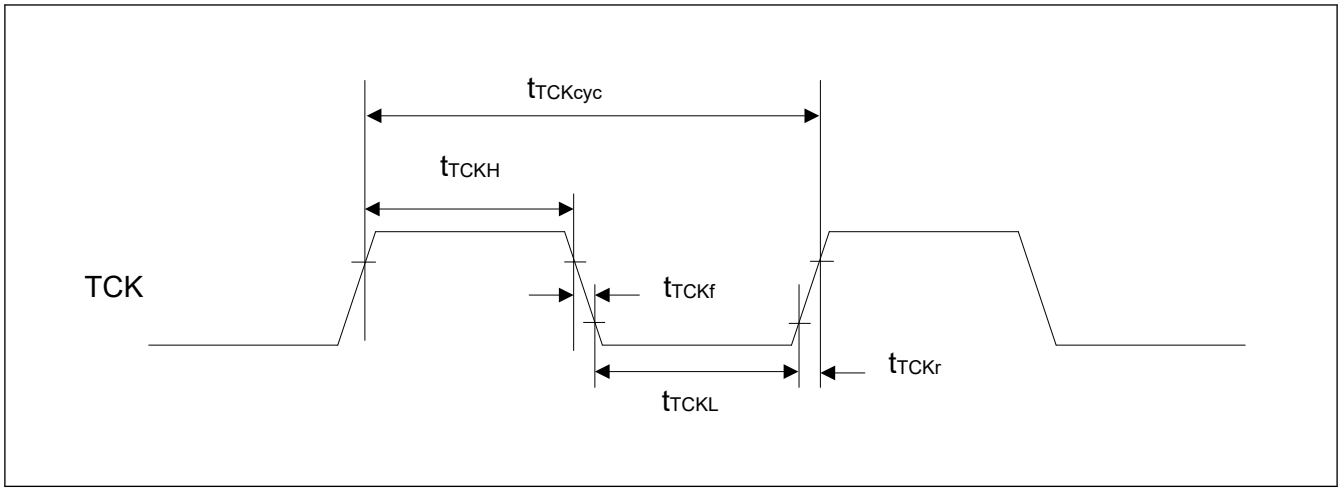


Figure 60.131 JTAG TCK timing

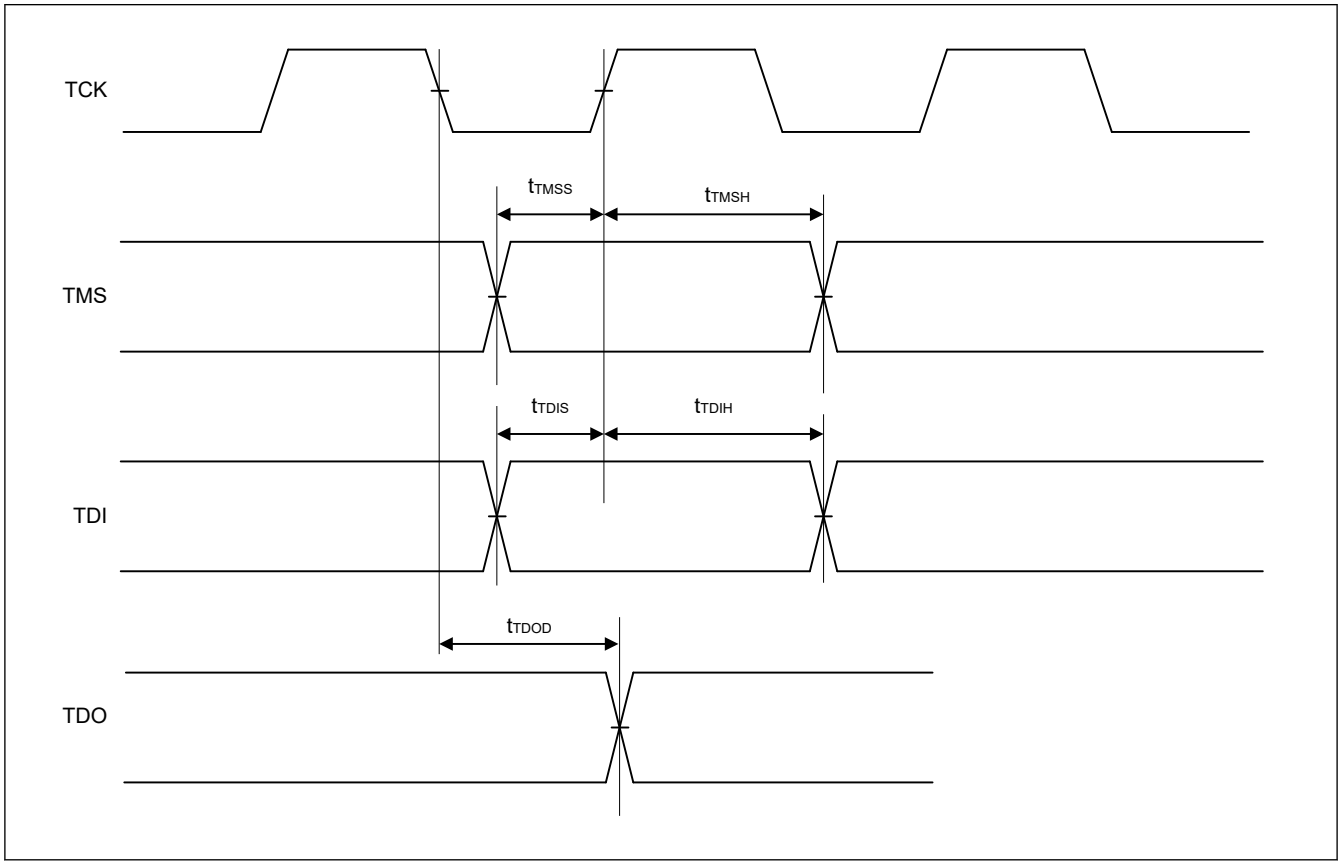


Figure 60.132 JTAG input/output timing

60.16 Serial Wire Debug (SWD)

Table 60.91 SWD

Parameter	VCC	Symbol	Min	Typ	Max	Unit	Test conditions
SWCLK clock cycle time	2.7 V or above	$t_{SWCKcyc}$	40.0	—	—	ns	Figure 60.133
	1.68 V or above		40.0	—	—	ns	
SWCLK clock high pulse width	2.7 V or above	t_{SWCKH}	0.375	—	—	$t_{SWCKcyc}$	
	1.68 V or above		0.375	—	—	$t_{SWCKcyc}$	
SWCLK clock low pulse width	2.7 V or above	t_{SWCKL}	0.375	—	—	$t_{SWCKcyc}$	
	1.68 V or above		0.375	—	—	$t_{SWCKcyc}$	
SWCLK clock rise time	2.7 V or above	t_{SWCKr}	—	—	0.125*1	$t_{SWCKcyc}$	
	1.68 V or above		—	—	0.125*1	$t_{SWCKcyc}$	
SWCLK clock fall time	2.7 V or above	t_{SWCKf}	—	—	0.125*1	$t_{SWCKcyc}$	
	1.68 V or above		—	—	0.125*1	$t_{SWCKcyc}$	
SWDIO setup time	2.7 V or above	t_{SWDS}	8.0	—	—	ns	Figure 60.134
	1.68 V or above		8.0	—	—	ns	
SWDIO hold time	2.7 V or above	t_{SWDH}	8.0	—	—	ns	
	1.68 V or above		8.0	—	—	ns	
SWDIO data delay time	2.7 V or above	t_{SWDD}	2.0	—	28.0	ns	
	1.68 V or above		2.0	—	32.0	ns	

Note 1. 1 µs at the longest

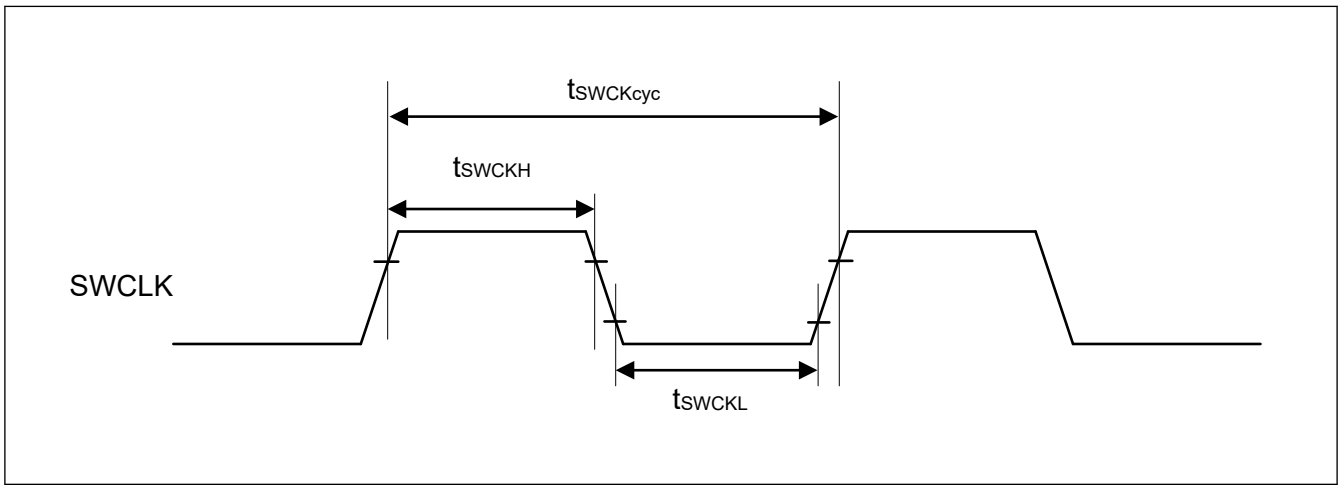


Figure 60.133 SWD SWCLK timing

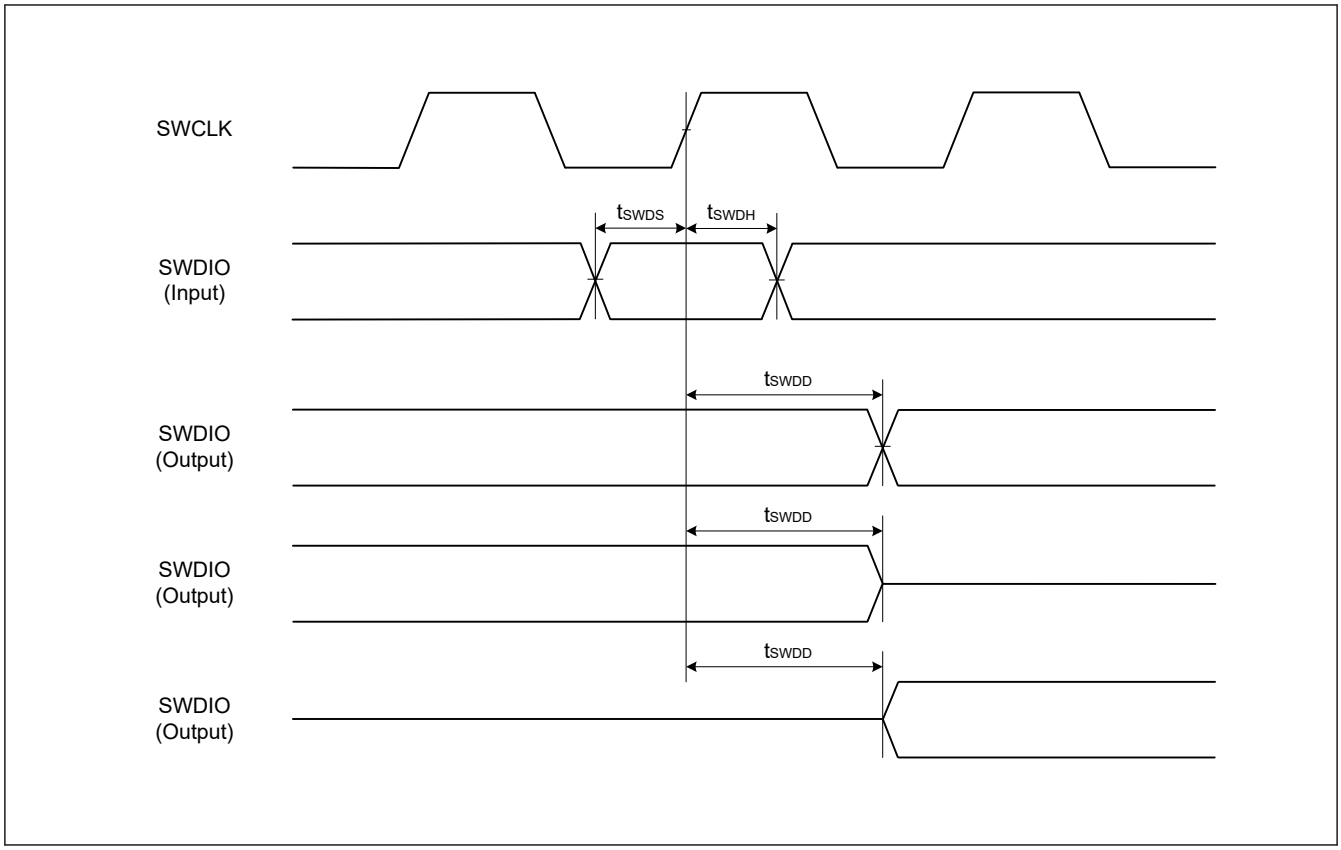


Figure 60.134 SWD input/output timing

60.17 Embedded Trace Macro Interface (ETM)

Table 60.92 ETM (1 of 2)

Conditions: High-speed high drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter	VCC	Symbol	Min	Typ	Max	Unit	Test conditions
TCLK clock cycle time	2.7 V or above	$t_{TCLKcyc}$	16.6	—	—	ns	Figure 60.135
	1.68 V or above		16.6	—	—	ns	
TCLK clock high pulse width	2.7 V or above	t_{TCLKH}	7.3	—	—	ns	
	1.68 V or above		6.3	—	—	ns	
TCLK clock low pulse width	2.7 V or above	t_{TCLKL}	7.3	—	—	ns	
	1.68 V or above		6.3	—	—	ns	
TCLK clock rise time	2.7 V or above	t_{TCLKr}	—	—	1.0	ns	
	1.68 V or above		—	—	2.0	ns	
TCLK clock fall time	2.7 V or above	t_{TCLKf}	—	—	1.0	ns	
	1.68 V or above		—	—	2.0	ns	

Table 60.92 ETM (2 of 2)

Conditions: High-speed high drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter	VCC	Symbol	Min	Typ	Max	Unit	Test conditions
TDATA[3:0] output valid time	2.7 V or above	t_{TRDV}	—	—	$t_{TCLKcyc}/4 + 1.6$	ns	Figure 60.136
	1.68 V or above		—	—	$t_{TCLKcyc}/4 + 1.6$	ns	
TDATA[3:0] output hold time	2.7 V or above	t_{TRDH}	1.5	—	—	ns	
	1.68 V or above		1.5	—	—	ns	

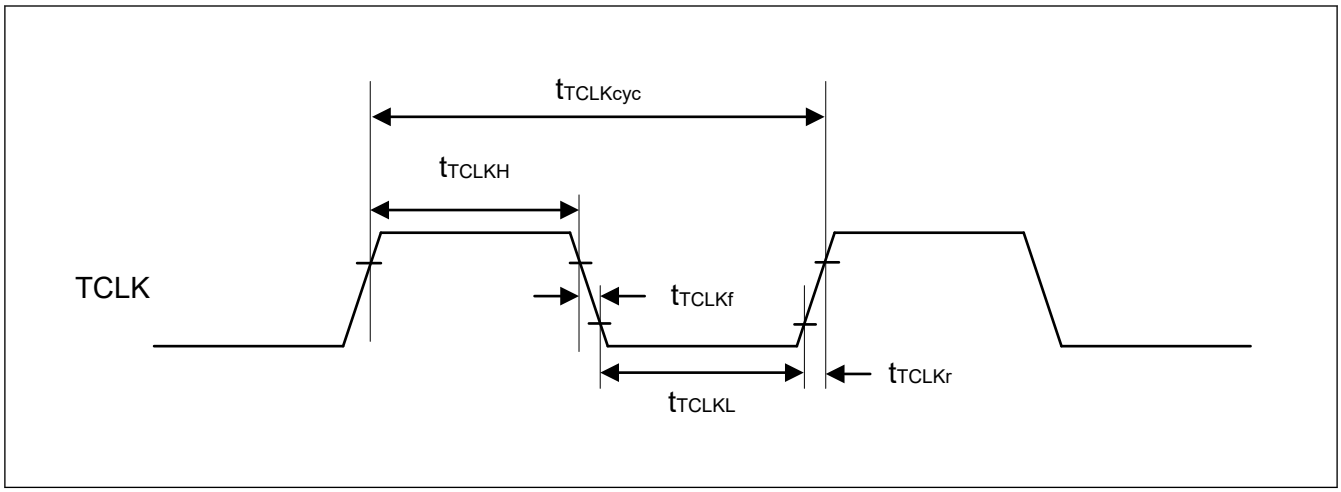


Figure 60.135 ETM TCLK timing

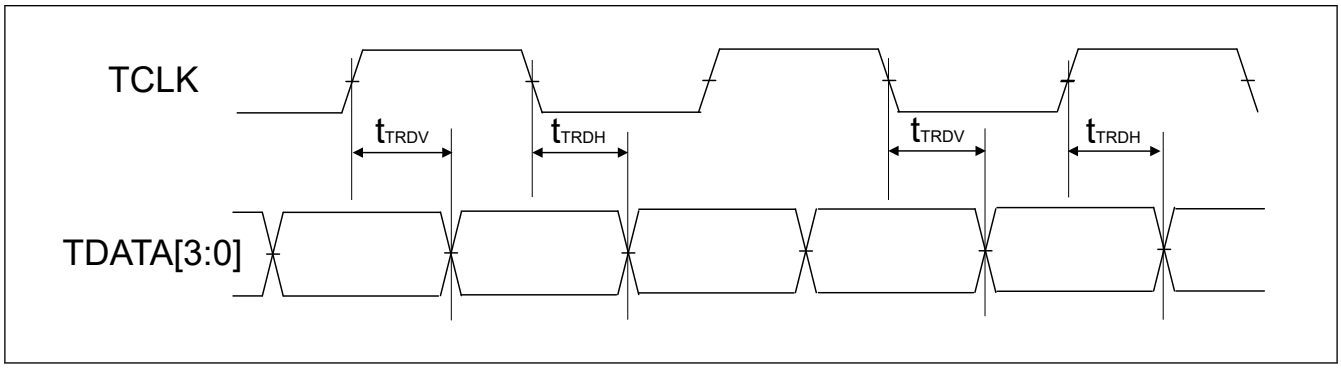


Figure 60.136 ETM output timing

Appendix 1. Port States in Each Processing Mode

Function	Pin function	Reset	Software Standby mode(SSTBY)		Deep Software Standby mode 1,2,3 (DSTBY1,2,3)		After Deep Software Standby mode is canceled (return to startup mode)	
			OPE=0	OPE=1	DSTBY1	DSTBY2/ DSTBY3	IOKEEP = 0	IOKEEP = 1[*1]
Mode	MD	Pull-up	Keep-I		Keep		Pull-up	Keep
JTAG/SWD	TCK/TMS/TDI/SWCLK	Pull-up	TCK/TDI/TMS/SWCLK input		TCK/TDI/TMS/SWCLK input		TCK/TDI/TMS/SWCLK input	
	TDO	Output	TDO output		TDO output		TDO output	
	SWDIO	Pull-up	SWDIO inout		SWDIO inout		SWDIO inout	
Trace	TCLK/TDATAx/SWO	TCLK/ TDATAx/SWO output	TCLK/TDATAx/SWO output		TCLK/TDATAx/SWO output		TCLK/TDATAx/SWO output	
IRQ	IRQx	Hi-Z	Hi-Z[*2]		Keep		Hi-Z	Keep
	IRQx-DS (x:Other than 5)	Hi-Z	Hi-Z[*2]		Keep[*3]		Hi-Z	Keep
	IRQ5-DS	Hi-Z	Hi-Z[*2]		Keep[*3]		Hi-Z	
AGT	AGTIOn	Hi-Z	AGTIOn inout		Keep		Hi-Z	Keep
	AGTOn/AGTOAn/ AGTOBn	Hi-Z	AGTOn/AGTOAn/AGTOBn output		Keep		Hi-Z	Keep
ULPT	ULPTEEn/ULPTEVIn	Hi-Z	ULPTEEn/ULPTEVIn input		Keep		Hi-Z	Keep
	ULPTEEn-DS/ ULPTEVIn-DS	Hi-Z	ULPTEEn-DS/ULPTEVIn-DS input		ULPTEEn-DS/ ULPTEVIn-DS input	Hi-Z	Hi-Z	Keep
	ULPTOn/ ULPTOAn/ ULPTOBn	Hi-Z	ULPTOn/ULPTOAn/ULPTOBn output		Keep		Hi-Z	Keep
	ULPTOn-DS/ ULPTOAn-DS/ ULPTOBn-DS	Hi-Z	ULPTOn/ULPTOAn-DS/ ULPTOBn-DS output		ULPTOn/ ULPTOAn-DS/ ULPTOBn-DS output	Keep	Hi-Z	From DSTBY1: ULPTOn/ ULPTOAn-DS/ ULPTOBn-DS output From DSTBY2,3: Keep
IIC	SCLn/SDAn	Hi-Z	Keep-O[*2]		Keep		Hi-Z	Keep
I3C	I3C_SCL0/I3C_SDA0	Hi-Z	Keep-O[*2]		Hi-Z		Hi-Z	
USBFS	USB_OVRCURx	Hi-Z	Hi-Z[*2]		Keep		Hi-Z	Keep
	USB_OVRCURx-DS/ USB_VBUS	Hi-Z	Hi-Z[*2]		Keep[*3]	Keep	Hi-Z	Keep
	USB_DP/USB_DM	Hi-Z	Keep-O[*4]		Keep[*3]	Keep	Hi-Z	Keep
USBHS	USBHS_OVRCURx	Hi-Z	Hi-Z[*2]		Keep		Hi-Z	Keep
	USBHS_OVRCURx-DS /USBHS_VBUS	Hi-Z	Hi-Z[*2]		Keep[*3]	Keep	Hi-Z	Keep
	USBHS_DP/ USBHS_DM	Hi-Z	Keep-O[*4]		Keep[*5]	Keep	Hi-Z	Keep
RTC	RTCICx	Hi-Z	Hi-Z[*2]		Keep[*3]		Hi-Z	Keep
	RTCOUT	Hi-Z	RTCOUT output		Keep		Hi-Z	Keep
ACMPHS	VCOUT	Hi-Z	VCOUT output		Keep		Hi-Z	Keep
CLKOUT	CLKOUT	Hi-Z	CLKOUT output		Keep		Hi-Z	Keep
DAC	DAn	Hi-Z	D/A output retained		Hi-Z		Hi-Z	

Function	Pin function	Reset	Software Standby mode(SSTBY)		Deep Software Standby mode 1,2,3 (DSTBY1,2,3)		After Deep Software Standby mode is canceled (return to startup mode)	
			OPE=0	OPE=1	DSTBY1	DSTBY2/ DSTBY3	IOKEEP = 0	IOKEEP = 1*1
External bus (CS, SDRAM area)	EBCLK/SDCLK	Hi-Z	High-level output		Keep		Hi-Z	Keep
	Dxx/DQxx	Hi-Z	Hi-Z		Hi-Z		Hi-Z	
	Axx/DQMx	Hi-Z	Hi-Z	Keep-O	Keep		Hi-Z	Keep
	BC0	Hi-Z	Hi-Z	*6	Keep		Hi-Z	Keep
	BC1 to BC3/CSx/RD/WRx/WE	Hi-Z	Hi-Z	High-level output	Keep		Hi-Z	Keep
	ALE	Hi-Z	Hi-Z	Low-level output	Keep		Hi-Z	Keep
	CKE/SDCS/RAS/CAS	Hi-Z	Hi-Z	SDSELF.SFEN = 0: High-level output SDSELF.SFEN = 1: Low-level output	Keep		Hi-Z	Keep
P400/P401	Other than function IRQ5-DS	Hi-Z	Keep-O*2		Hi-Z		Hi-Z	
Others	—	Hi-Z	Keep-O		Keep		Hi-Z	Keep

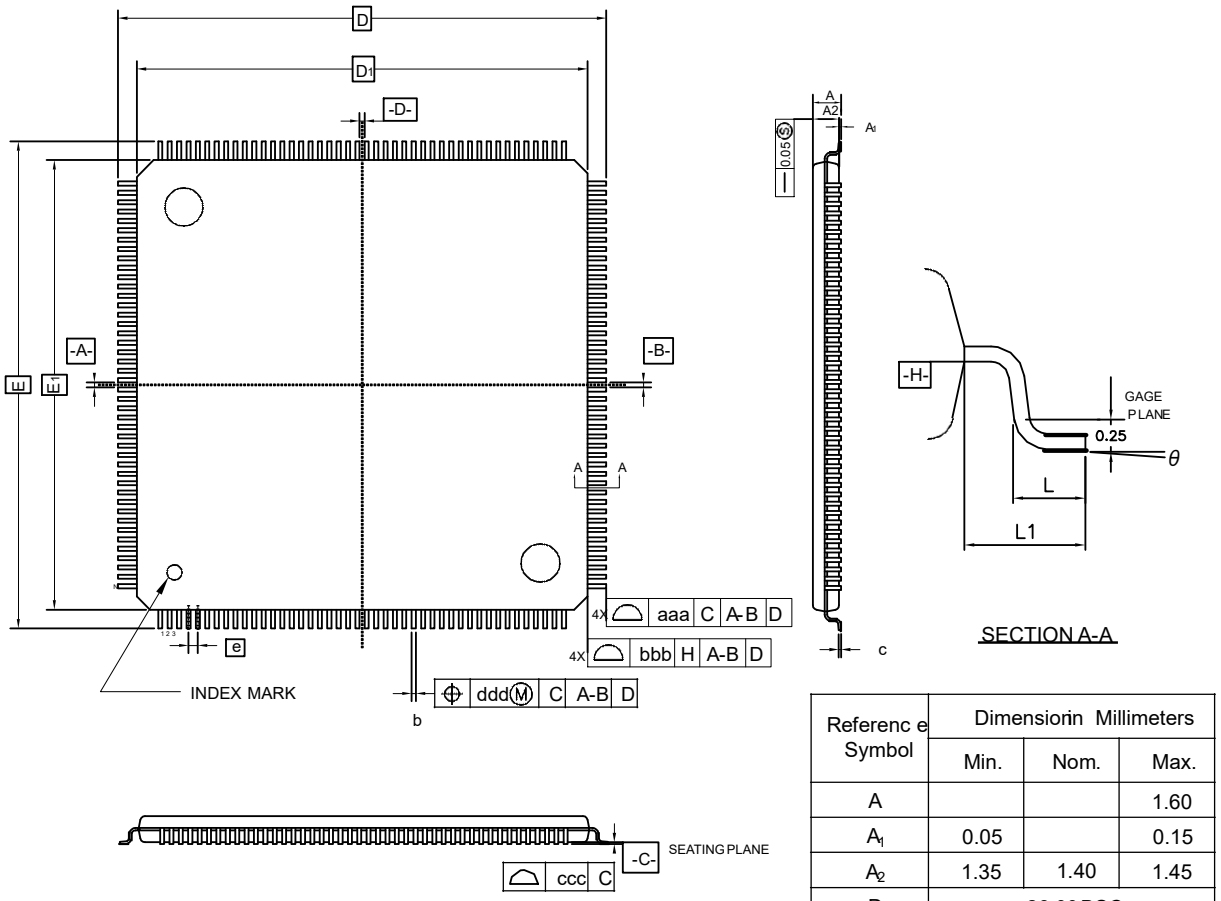
Note: Hi-Z: High-impedance
Keep-O: Output pins retain their previous values. Input pins go to high-impedance.
Keep-I: Pin states are retained same as during periods in Normal mode.
Keep: Pin states are retained same as during periods in Software Standby mode.

Note 1. Retains the I/O port state until the DPSBYCR.IOKEEP bit is cleared to 0.
Note 2. Input is enabled if the pin is specified as the Software Standby canceling source while it is used as an external interrupt pin.
Note 3. Input is enabled if the pin is specified as the Deep Software Standby canceling source.
Note 4. Input is enabled while the pin is used as an input pin.
Note 5. For host operation, set the USBHS.SYSCFG.DRPD bit to 1 to enable the USBHS_DP and USBHS_DM pull-down resistors.
For device operation, set the USBHS.SYSCFG.DPRPU bit to 1 to enable the DP pull-up resistor.
Note 6. Depending on the operation before the transition to Software Standby Mode, either High-level or Low-level is output. See section 14.9.1. BC0 pin when in Software Standby Mode (SSTBY) and SBYCR.OPE=1.

Appendix 2. Package Dimensions

Information on the latest version of the package dimensions or mountings is displayed in “Packages” on the Renesas Electronics Corporation website.

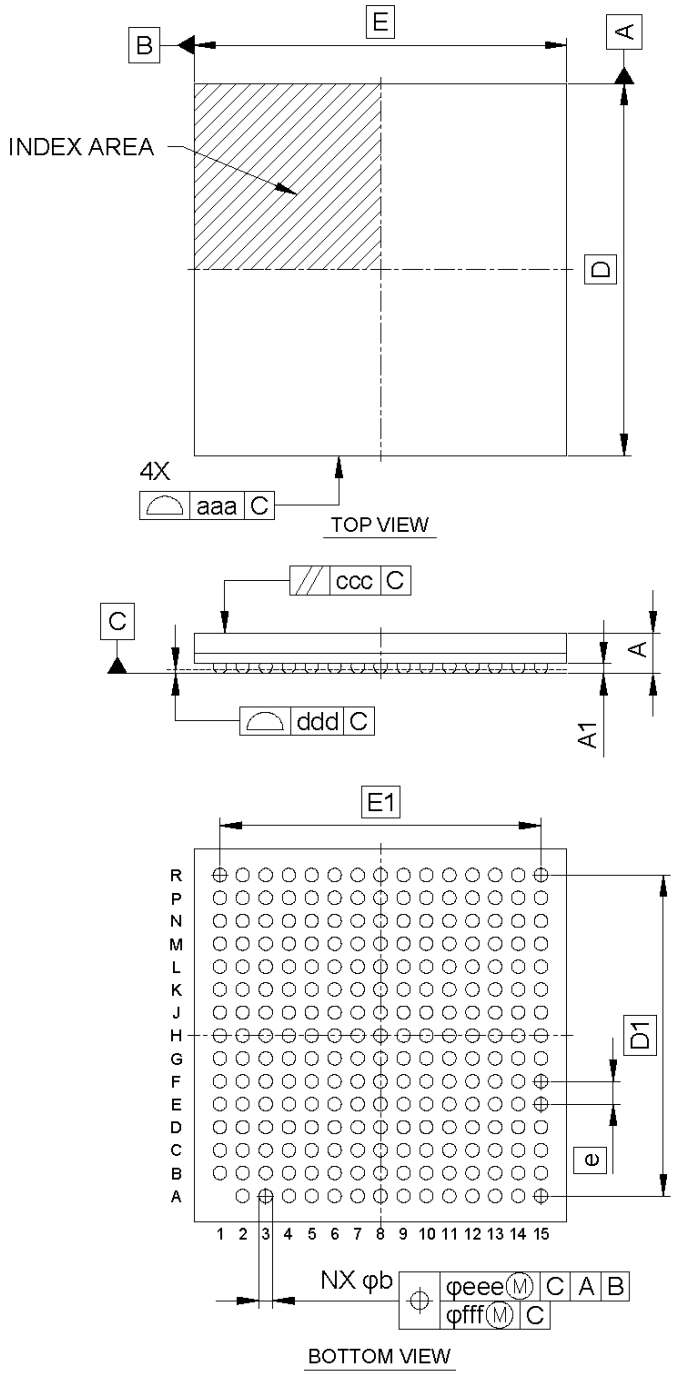
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFQFP176-24x24-0.50	PLQP0176KJ-A	1.90



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A			1.60
A_1	0.05		0.15
A_2	1.35	1.40	1.45
D		26.00 BSC.	
D_1		24.00 BSC.	
E		26.00 BSC.	
E_1		24.00 BSC.	
N		176	
e		0.50 BSC.	
b	0.17	0.22	0.27
c	0.09		0.20
q	0	3.5	7
L	0.45	0.60	0.75
L_1		1.00 REF.	
aaa			0.20
bbb			0.20
ccc			0.08
ddd			0.08

Figure A2.1 LQFP 176-pin

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFBGA224-13x13-0.80	PLBG0224GD-A	0.44



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	—	13.00	—
E	—	13.00	—
D1	—	11.20	—
E1	—	11.20	—
A	—	—	1.47
A1	0.29	—	—
b	0.42	0.47	0.52
e	—	0.80	—
aaa	—	—	0.15
ccc	—	—	0.20
ddd	—	—	0.12
eee	—	—	0.15
fff	—	—	0.08
N	—	224	—

Figure A2.2 BGA 224-pin

Appendix 3. I/O Registers

This appendix describes I/O register address and access cycles by function.

3.1 Peripheral Base Addresses

This section provides the base addresses for peripherals described in this manual. Table A3.1 shows the name, description, and the base address of each peripheral.

Table A3.1 Peripheral base address (1 of 3)

Description	Name of Secure registers	Base address of Secure registers in Secure alias region	Name of Non-secure registers	Base address of Non-secure registers in Non-secure alias region
Renesas Memory Protection Unit	RMPU	0x4000_0000	RMPU_NS	0x5000_0000
SRAM Control	SRAM	0x4000_2000	SRAM_NS	0x5000_2000
BUS Control	BUS	0x4000_3000	BUS_NS	0x5000_3000
Common Interrupt Controller	ICU_COMMON	0x4000_6000	ICU_COMMON_NS	0x5000_6000
CPU System Security Control Unit	CPSCU	0x4000_8000	CPSCU_NS	0x5000_8000
Direct memory access controller 00	DMAC00	0x4000_A000	DMAC00_NS	0x5000_A000
Direct memory access controller 01	DMAC01	0x4000_A040	DMAC01_NS	0x5000_A040
Direct memory access controller 02	DMAC02	0x4000_A080	DMAC02_NS	0x5000_A080
Direct memory access controller 03	DMAC03	0x4000_A0C0	DMAC03_NS	0x5000_A0C0
Direct memory access controller 04	DMAC04	0x4000_A100	DMAC04_NS	0x5000_A100
Direct memory access controller 05	DMAC05	0x4000_A140	DMAC05_NS	0x5000_A140
Direct memory access controller 06	DMAC06	0x4000_A180	DMAC06_NS	0x5000_A180
Direct memory access controller 07	DMAC07	0x4000_A1C0	DMAC07_NS	0x5000_A1C0
DMAC Module Activation 0	DMA0	0x4000_A800	DMA0_NS	0x5000_A800
Data Transfer Controller 0	DTC0	0x4000_AC00	DTC0_NS	0x5000_AC00
Interrupt Controller	ICU	0x4000_C000	ICU_NS	0x5000_C000
CPU Control Registers	CPU_CTRL	0x4000_F000	CPU_CTRL_NS	0x5000_F000
On-Chip Debug	OCD_CPU	0x4001_1000	OCD_CPU_NS	0x5001_1000
DAP Function	OCD_DAP	0x8001_1000	—	—
Debug Function	CPU_DBG	0x4001_B000	CPU_DBG_NS	0x5001_B000
System Control	SYSC	0x4001_E000	SYSC_NS	0x5001_E000
Temperature Sensor Data	TSD	0x4011_B000	TSD_NS	0x5011_B000
Event Link Controller	ELC	0x4020_1000	ELC_NS	0x5020_1000
Realtime Clock	RTC	0x4020_2000	RTC_NS	0x5020_2000
Independent Watchdog Timer	IWDT	0x4020_2200	IWDT_NS	0x5020_2200
Clock Frequency Accuracy Measurement Circuit	CAC	0x4020_2400	CAC_NS	0x5020_2400
Watchdog Timer 0	WDT0	0x4020_2600	WDT0_NS	0x5020_2600
Module Stop Control A,B,C,D,E	MSTP	0x4020_3000	MSTP_NS	0x5020_3000
Peripheral Security Control Unit	PSCU	0x4020_4000	PSCU_NS	0x5020_4000
Port Output Enable Module for GPT	POEG	0x4021_2000	POEG_NS	0x5021_2000
Ultra-Low Power Timer 0	ULPT0	0x4022_0000	ULPT0_NS	0x5022_0000
Ultra-Low Power Timer 1	ULPT1	0x4022_0100	ULPT1_NS	0x5022_0100

Table A3.1 Peripheral base address (2 of 3)

Description	Name of Secure registers	Base address of Secure registers in Secure alias region	Name of Non-secure registers	Base address of Non-secure registers in Non-secure alias region
Low Power Asynchronous General purpose Timer 0	AGT0	0x4022_1000	AGT0_NS	0x5022_1000
Low Power Asynchronous General purpose Timer 1	AGT1	0x4022_1100	AGT1_NS	0x5022_1100
Temperature Sensor	TSN	0x4023_5000	TSN_NS	0x5023_5000
High-Speed Analog Comparator 0	ACMPHS0	0x4023_6000	ACMPHS0_NS	0x5023_6000
High-Speed Analog Comparator 1	ACMPHS1	0x4023_6100	ACMPHS1_NS	0x5023_6100
USB 2.0 FS Module	USBFS	0x4025_0000	USBFS_NS	0x5025_0000
SD Host Interface 0	SDHI0	0x4025_2000	SDHI0_NS	0x5025_2000
SD Host Interface 1	SDHI1	0x4025_2400	SDHI1_NS	0x5025_2400
Serial Sound Interface Enhanced (SSIE) 0	SSIE0	0x4025_D000	SSIE0_NS	0x5025_D000
Serial Sound Interface Enhanced (SSIE) 1	SSIE1	0x4025_D100	SSIE1_NS	0x5025_D100
Inter-Integrated Circuit 0	IIC0	0x4025_E000	IIC0_NS	0x5025_E000
Inter-Integrated Circuit 0 Wake-up Unit	IIC0WU	0x4025_E014	IIC0WU_NS	0x5025_E014
Inter-Integrated Circuit 1	IIC1	0x4025_E100	IIC1_NS	0x5025_E100
Octal Serial Peripheral Interface 0	OSPI0_B	0x4026_8000	OSPI0_B_NS	0x5026_8000
Decryption On-The-Fly 0	DOTF0	0x4026_8800	DOTF0_NS	0x5026_8800
CRC Calculator	CRC	0x4031_0000	CRC_NS	0x5031_0000
Data Operation Circuit	DOC_B	0x4031_1000	DOC_B_NS	0x5031_1000
General PWM 32-bit Timer 0	GPT320	0x4032_2000	GPT320_NS	0x5032_2000
General PWM 32-bit Timer 1	GPT321	0x4032_2100	GPT321_NS	0x5032_2100
General PWM 32-bit Timer 2	GPT322	0x4032_2200	GPT322_NS	0x5032_2200
General PWM 32-bit Timer 3	GPT323	0x4032_2300	GPT323_NS	0x5032_2300
General PWM 32-bit Timer 4	GPT324	0x4032_2400	GPT324_NS	0x5032_2400
General PWM 32-bit Timer 5	GPT325	0x4032_2500	GPT325_NS	0x5032_2500
General PWM 32-bit Timer 6	GPT326	0x4032_2600	GPT326_NS	0x5032_2600
General PWM 32-bit Timer 7	GPT327	0x4032_2700	GPT327_NS	0x5032_2700
General PWM 16-bit Timer 8	GPT168	0x4032_2800	GPT168_NS	0x5032_2800
General PWM 16-bit Timer 9	GPT169	0x4032_2900	GPT169_NS	0x5032_2900
General PWM 16-bit Timer 10	GPT1610	0x4032_2A00	GPT1610_NS	0x5032_2A00
General PWM 16-bit Timer 11	GPT1611	0x4032_2B00	GPT1611_NS	0x5032_2B00
General PWM 16-bit Timer 12	GPT1612	0x4032_2C00	GPT1612_NS	0x5032_2C00
General PWM 16-bit Timer 13	GPT1613	0x4032_2D00	GPT1613_NS	0x5032_2D00
Output Phase Switching Controller	GPT_OPS	0x4032_3F00	GPT_OPS_NS	0x5032_3F00
12bit A/D Converter 0	ADC120	0x4033_2000	ADC120_NS	0x5033_2000
12bit A/D Converter 1	ADC121	0x4033_2200	ADC121_NS	0x5033_2200
12-bit D/A converter	DAC12	0x4033_3000	DAC12_NS	0x5033_3000
Graphics LCD Controller	GLCDC	0x4034_2000	GLCDC_NS	0x5034_2000

Table A3.1 Peripheral base address (3 of 3)

Description	Name of Secure registers	Base address of Secure registers in Secure alias region	Name of Non-secure registers	Base address of Non-secure registers in Non-secure alias region
2D Drawing Engine	DRW	0x4034_4000	DRW_NS	0x5034_4000
MIPI DSI link	MIPI_DSI	0x4034_6000	MIPI_DSI_NS	0x5034_6000
MIPI PHY	MIPI_PHY0	0x4034_6C00	MIPI_PHY0_NS	0x5034_6C00
Capture Engine Unit	CEU	0x4034_8000	CEU_NS	0x5034_8000
USB 2.0 High-Speed Module	USBHS	0x4035_1000	USBHS_NS	0x5035_1000
DMA Controller for the Ethernet Controller Channel 0	EDMAC0	0x4035_4000	EDMAC0_NS	0x5035_4000
Ethernet Controller Channel 0	ETHERC0	0x4035_4100	ETHERC0_NS	0x5035_4100
Serial Communication Interface 0	SCI0_B	0x4035_8000	SCI0_B_NS	0x5035_8000
Serial Communication Interface 1	SCI1_B	0x4035_8100	SCI1_B_NS	0x5035_8100
Serial Communication Interface 2	SCI2_B	0x4035_8200	SCI2_B_NS	0x5035_8200
Serial Communication Interface 3	SCI3_B	0x4035_8300	SCI3_B_NS	0x5035_8300
Serial Communication Interface 4	SCI4_B	0x4035_8400	SCI4_B_NS	0x5035_8400
Serial Communication Interface 9	SCI9_B	0x4035_8900	SCI9_B_NS	0x5035_8900
Serial Peripheral Interface 0	SPI0	0x4035_C000	SPI0_NS	0x5035_C000
Serial Peripheral Interface 1	SPI1	0x4035_C100	SPI1_NS	0x5035_C100
I3C Bus Interface	I3C	0x4035_F000	I3C_NS	0x5035_F000
Error correction circuit for MBRAM0	ECCMB0	0x4036_F200	ECCMB0_NS	0x5036_F200
Error correction circuit for MBRAM1	ECCMB1	0x4036_F300	ECCMB1_NS	0x5036_F300
CANFD Module 0	CANFD0	0x4038_0000	CANFD0_NS	0x5038_0000
CANFD Module 1	CANFD1	0x4038_2000	CANFD1_NS	0x5038_2000
Port 0 Control Registers	PORT0	0x4040_0000	PORT0_NS	0x5040_0000
Port 1 Control Registers	PORT1	0x4040_0020	PORT1_NS	0x5040_0020
Port 2 Control Registers	PORT2	0x4040_0040	PORT2_NS	0x5040_0040
Port 3 Control Registers	PORT3	0x4040_0060	PORT3_NS	0x5040_0060
Port 4 Control Registers	PORT4	0x4040_0080	PORT4_NS	0x5040_0080
Port 5 Control Registers	PORT5	0x4040_00A0	PORT5_NS	0x5040_00A0
Port 6 Control Registers	PORT6	0x4040_00C0	PORT6_NS	0x5040_00C0
Port 7 Control Registers	PORT7	0x4040_00E0	PORT7_NS	0x5040_00E0
Port 8 Control Registers	PORT8	0x4040_0100	PORT8_NS	0x5040_0100
Port 9 Control Registers	PORT9	0x4040_0120	PORT9_NS	0x5040_0120
Port A Control Registers	PORTA	0x4040_0140	PORTA_NS	0x5040_0140
Port B Control Registers	PORTB	0x4040_0160	PORTB_NS	0x5040_0160
Pmn Pin Function Control Register	PFS	0x4040_0800	PFS_NS	0x5040_0800
Flash Cache	FCACHE	0x4001_C100	FCACHE_NS	0x5001_C100
Data Flash	FLAD	0x4011_C000	FLAD_NS	0x5011_C000
Flash Application Command Interface	FACI	0x4011_E000	FACI_NS	0x5011_E000
Data Flash Security Setting	FDFS	0x2703_0000	—	—

Note: Name = Peripheral name
Description = Peripheral functionality
Base address = Lowest reserved address or address used by the peripheral

3.2 Access Cycles

This section provides access cycle information for the I/O registers described in this manual.

- Registers are grouped by associated module.
- The number of access cycles indicates the number of cycles based on the specified reference clock.
- In the internal I/O area, reserved addresses that are not allocated to registers must not be accessed, otherwise operations cannot be guaranteed.
- The number of I/O access cycles depends on bus cycles of the internal peripheral bus, divided clock synchronization cycles, and wait cycles of each module. Divided clock synchronization cycles differ depending on the frequency ratio between ICLK and PCLK.
- When the frequency of ICLK is equal to that of PCLK, the number of divided clock synchronization cycles is always constant.
- When the frequency of ICLK is greater than that of PCLK, at least 1 PCLK cycle is added to the number of divided clock synchronization cycles.
- The number of write access cycles indicates the number of cycles obtained by non-bufferable write access.

Note: This applies to the number of cycles when access from the CPU does not conflict with the instruction fetching to the external memory or bus access from other bus masters such as DTC or DMAC.

Table A3.2 Access cycles (1 of 3)

Peripheral base address symbol	Address*1		Number of access cycles				Cycle Unit	Related function
			ICLK = PCLK		ICLK > PCLK*2			
	From	To	Read	Write	Read	Write		
RMPU, SRAM, BUS, ICU_COMMON, CPSCU, DMAC0n, DMA0, DTC0, ICU, CPU_CTRL	0x4000_0000	0x4001_CFFF	3	2	3	2	ICLK	Renesas Memory Protection Unit, SRAM Control, BUS Control, Common Interrupt Controller, CPU System Security Control Unit, Direct memory access controller 0 n, DMAC Module Activation 0, Data Transfer Controller 0, Interrupt Controller, CPU Control Registers
CPU_OCD	0x4001_1004	0x4001_1FFF	7	2	7	2	ICLK	On-Chip Debug
CPU_DBG, FCACHE	0x4000_B000	0x4001_CFFF	3	2	3	2	ICLK	Debug Function, Flash Cache
SYSC	0x4001_E000	0x4001_E9FF	4	3	2 to 4	1 to 3	PCLKB	System Control
SYSC	0x4001_EA00	0x4001_ED7F	7	6	5 to 7	4 to 6	PCLKB	System Control
TSD	0x4011_B17C	0x4011_B17C	4	3	4	3	ICLK	Temperature Sensor Data
ELC, RTC	0x4020_1000	0x4020_21FF	4	3	2 to 4	1 to 3	PCLKB	Event Link Controller, Realtime Clock
IWDT	0x4020_2200	0x4020_22FF	4	65	2 to 4	63 to 65	PCLKB	Independent Watchdog Timer
CAC, WDT0, MSTP, PSCU, POEG	0x4020_2400	0x4021_2FFF	4	3	2 to 4	1 to 3	PCLKB	Clock Frequency Accuracy Measurement Circuit, Watchdog Timer 0, Module Stop Control, Peripheral Security Control Unit, Port Outupt Enable Module for GPT
ULPTn	0x4022_0000	0x4022_01FF	6	65	4 to 6	63 to 65	PCLKB	Ultra-Low Power Timer n
AGTn	0x4022_1000	0x4022_11FF	6	3	4 to 6	1 to 3	PCLKB	Low Power Asynchronous General purpose Timer n
TSN	0x4023_5000	0x4023_5FFF	4	3	2 to 4	1 to 3	PCLKB	Temperature Sensor
ACMPHSn	0x4023_6000	0x4023_61FF	3	3	1 to 3	1 to 3	PCLKB	High-Speed Analog Comparator n
USBFS	0x4025_0000	0x4025_03FF	5	4	3 to 5	2 to 4	PCLKB	USB 2.0 FS Module

Table A3.2 Access cycles (2 of 3)

Peripheral base address symbol	Address[*1]		Number of access cycles				Cycle Unit	Related function
			ICLK = PCLK		ICLK > PCLK[*2]			
	From	To	Read	Write	Read	Write		
USBFS	0x4025_0400	0x4025_04FF	4	65	2 to 4	63 to 65	PCLKB	USB 2.0 FS Module
SDHIn, SSIEn, IICn, OSPI0, DOTF0	0x4025_2000	0x4026_88FF	4	3	2 to 4	1 to 3	PCLKB	SD Host Interface n, Serial Sound Interface Enhanced n, Inter-Integretad Circuit n, Octal Serial Peripheral Interface 0, Decryption On-The-Fly 0
CRC, DOC	0x4031_0000	0x4031_1FFF	4	3	2 to 4	1 to 3	PCLKA	CRC Calculator, Data Operation Circuit
GPT32n, GPT16n, GPT_OPS	0x4032_2000	0x4032_3FFF	7	4	5 to 7	2 to 4	PCLKA	General PWM 32-Bit Timer n, General PWM 16-Bit Timer n, Output Phase Switching Controller
ADC12n, DAC12n, GLCDC, DRW, MIPI_DSI, MIPI_PHY0	0x4033_2000	0x4034_6FFF	4	3	2 to 4	1 to 3	PCLKA	12-bit A/D Converter n, 12-bit D/A Converter n, Graphic LCD Controller, 2D Drawing Engine, MIPI DSI link, MIPI PHY
CEU	0x4034_8000	0x4034_FFFF	7	5	5 to 7	3 to 5	PCLKA	Capture Engine Unit
USBHS[*3]	0x4035_1000	0x4035_115F	BWAIT+4	BWAIT+3	(BWAIT+2) to (BWAIT+4)	(BWAIT+1) to (BWAIT+3)	PCLKA	USB 2.0 High-Speed Module
USBHS[*3]	0x4035_1160	0x4035_1167	BWAIT+4	130	(BWAIT+2) to (BWAIT+4)	128 to 130	PCLKA	USB 2.0 High-Speed Module
USBHS	0x4035_1168	0x4035_116F	8	130	6 to 8	128 to 130	PCLKA	USB 2.0 High-Speed Module
EDMAC0	0x4035_4000	0x4035_40FF	5	4	3 to 5	2 to 4	PCLKA	DMA Controller for the Ethernet Controller Channel 0
ETHERC0	0x4035_4100	0x4035_43FF	14	13	12 to 14	11 to 13	PCLKA	Ethernet Controller Channel 0
SCIn, SPIn, I3C	0x4035_8000	0x4035_FFFF	4	3	2 to 4	1 to 3	PCLKA	Serial Communication Interface n, Serial Peripheral Interface n, I3C Bus Interface
ECCMBn	0x4036_F200	0x4036_F3FF	5	4	3 to 5	2 to 4	PCLKA	Error correction circuit for MBRAMn
CANFDn	0x4038_0000	0x4038_3FFF	4	3	2 to 4	1 to 3	PCLKA	CANFD Module n
PORTn	0x4040_0000	0x4040_01FF	4	2	4	2	ICLK	Port n Control Registers
PFS	0x4040_0800	0x4040_0FFF	8	2	8	2	ICLK	Pmn Pin Function Control Register
RSIP-E51A	—	—	1 to 3	2	1 to 3	1 to 2	PCLKA	Renesas Security IP

Table A3.2 Access cycles (3 of 3)

Peripheral base address symbol	Address[*1]		Number of access cycles				Cycle Unit	Related function
			ICLK = FCLK		ICLK > FCLK[*2]			
	From	To	Read	Write	Read	Write		
FLAD, FACI	0x4011_C040	0x4011_EFFF	4	3	4	3	FCLK	Data Flash, Flash Application Command Interface

Note 1. This table only shows secure address. Access cycle of the non-secure address is the same as its secure address.
Note 2. If the number of PCLK or FCLK cycles is non-integer (for example 1.5), the minimum value is without the decimal point, and the maximum value is rounded up to the decimal point. For example, 1.5 to 2. 5 is 1 to 3.
Note 3. BWAIT is the number of waits (not cycles) described in the USBHS.BUSWAIT register.

Appendix 4. Note for Register R/W

- A secure bus master issues a "secure access" using an address marked as secure by IDAU/SAU or MSAU.
- A secure bus master issues a "non-secure access" using an address marked as non-secure by IDAU/SAU or MSAU.
- A non-secure bus master issues a "non-secure access" using an address marked as non-secure by IDAU/SAU or MSAU.

Table A4.1 Type of Register Notes (S-TYPE)

TYPE	UM Description
S-TYPE-1	Only Secure access can write to this register. Read access is always allowed. Non-secure write access is ignored, but TrustZone access error is not generated.
S-TYPE-2	Read access is always allowed If the security attribution is configured as Secure, ● Secure write access is allowed. ● Non-secure write access is ignored, but TrustZone access error is not generated.
	If the security attribution is configured as Non-secure, ● Secure write access is ignored, but TrustZone access error is not generated. ● Non-secure access is allowed.
S-TYPE-3	If the security attribution is configured as Secure, ● Secure access is allowed. ● Non-secure write access is ignored and Non-secure read access is read as 0, TrustZone access error is generated
	If the security attribution is configured as Non-secure, ● Secure write access is ignored and Secure read access is read as 0, TrustZone access error is generated. ● Non-secure access is allowed
S-TYPE-4	If the security attribution is configured as Secure, ● Secure access is allowed ● Non-secure write access is ignored and Non-secure read access is read as 0, but TrustZone access error is not generated.
	If the security attribution is configured as Non-secure, ● Secure write access is ignored and Secure read access is read as 0, but TrustZone access error is not generated. ● Non-secure access is allowed.
S-TYPE-5	Access is always allowed.
S-TYPE-6	Secure access is allowed. Non-secure write access is ignored, and Non-secure read access is read as 0, TrustZone access error is generated.
S-TYPE-7	Secure write access is ignored, and Secure read access is read as 0, TrustZone access error is generated. Non-secure access is allowed.

Note: A non-secure bus master does NOT issue any access using an address marked as secure by IDAU/SAU or MSAU.

Table A4.2 Type of Register Notes (P-TYPE)

TYPE	UM Description
P-TYPE-1	Privileged write access is allowed. Read access is always allowed. Unprivileged write access is ignored, but TrustZone access error is not generated.
P-TYPE-2	Privileged access is allowed. Unprivileged write access is ignored, and Unprivileged read access is read as 0, TrustZone access error is generated.
P-TYPE-3	If the privilege attribution is configured as Privileged, ● Privileged access is allowed. ● Unprivileged write access is ignored and Unprivileged read access is read as 0, TrustZone access error is generated.
	If the privilege attribution is configured as Unprivilege, ● Privileged access and Unprivileged access are allowed.
P-TYPE-4	If the privilege attribution is configured as Privileged, ● Privileged access is allowed. ● Unprivileged write access is ignored and Unprivileged read access is read as 0, TrustZone access error is not generated.
	If the privilege attribution is configured as Unprivilege, ● Privileged access and Unprivileged access are allowed.
P-TYPE-5	Access is always allowed.

Appendix 5. Peripheral Variant

Table A5.1 shows the correspondence between the module name used in this manual and the Peripheral Variant.

Table A5.1 Module name vs Peripheral Variant

Module name	Peripheral variant
SCI	SCI_B
SPI	SPI_B
OSPI	OSPI_B
DOC	DOC_B

Revision History

Revision 1.10 — December 5, 2023

First edition, issued

Revision 1.20 — November 22, 2024

1. Overview:
- Updated Table 1.8 Communication interfaces.

2. CPU:
- Updated Table 2.9 CPU peripherals.

4. Address Space:
- Updated Note 1 in Figure 4.1 Memory map example of 2 MB flash product.

5. Resets:
- Updated Table 5.3 Module-related registers initialized by each reset source.

6. Option-Setting Memory:
- Updated 6.2.5 OFS1, OFS1_SEC : Option Function Select Register 1 for Non-secure and Secure.
- Updated 6.2.10 PBPS, PBPS_SEC : Permanent Block Protect Setting Register.
- Updated 6.2.20 CFGD2LOCK : Configuration Data 2 Lock Bit.
- Deleted 6.5 References.

8. Clock Generation Circuit:
- Updated Table 8.1 Clock generation circuit specifications for the clock sources.
- Updated 8.2.1 CGFSAR : Clock Generation Function Security Attribute Register.
- Updated 8.2.6 PLLCCR : PLL Clock Control Register.
- Updated 8.2.9 PLL2CCR : PLL2 Clock Control Register.
- Updated 8.2.13 MOSCCR : Main Clock Oscillator Control Register.
- Updated 8.2.16 HOCOCR : High-Speed On-Chip Oscillator Control Register.
- Added 8.2.17 HOCOCR2 : High-Speed On-Chip Oscillator Control Register2.
- Updated 8.2.20 FLLCR2 : FLL Control Register2.
- Updated 8.2.52 TRCKCR : Trace Clock Control Register.
- Added Note 1 to the following sections:
  - 8.10.1 CPU Clock (CPUCLK)
  - 8.10.2 System Clock (ICLK)
  - 8.10.4 Peripheral Module Clock (PCLKA, PCLKB, PCLKC, PCLKD, PCLKE)
  - 8.10.5 FlashIF Clock (FCLK)
  - 8.10.6 External Bus Clock (BCLK, EBCLK)
  - 8.10.7 SDRAM Clock (SDCLK)
  - 8.10.9 SCI Clock (SCICLK)
  - 8.10.10 SPI Clock (SPICLK)
  - 8.10.11 Octal-SPI clock (OCTACLK, OCTADIVCLK)
  - 8.10.12 CANFD Core clock (CANFDCLK)
  - 8.10.13 USB Clock (USBCLK)
  - 8.10.14 USB Clock (USB60CLK)
  - 8.10.16 LCD Clock (LCDCLK)
  - 8.10.17 I3C Clock (I3CCLK)
  - 8.10.26 External Pin Output Clock (CLKOUT)
- Updated Table 8.14 Example of the HOCO initial setting procedure after reset release / after Deep Software Standby cancellation (OFS1(_SEC).HOCOEN = 1, without FLL).
- Updated Table 8.15 Example of the HOCO setting procedure with FLL function after reset release / after Deep Software Standby cancellation (OFS1(_SEC).HOCOEN = 1).

10. Low Power Modes:
- Updated 10.2.4 RSCSAR : RAM Standby Control Security Attribution Register.
- Updated 10.2.13 PDRAMSCR0 : SRAM Power Domain Standby Control Register 0.
- Updated 10.6.1 Power Gating Control using power domain control register.

11. Battery Backup Function:
- Updated 11.2.12 VBTBPCR2 : VBATT Battery Power Supply Control Register 2.

12. Register Write Protection:
- Updated Table 12.1 Association between the bits in the PRCR register and registers to be protected.

13. Interrupt Controller Unit:
- Updated Table 13.4 Event table.

Revision 1.20 — November 22, 2024

14. Buses:
- Updated Figure 14.1 System bus connection.
- Updated Table 14.2 System bus access path.
- Updated 14.3.8 CSnMOD : CSn Mode Register (n = 0 to 7).
- Updated 14.3.10 CSnWCR2 : CSn Wait Control Register 2 (n = 0 to 7).
- Updated 14.7.2 Operations When a Bus Error Occurs.
- Updated Table 14.49 Conditions leading to illegal address access errors.

16. DMA Controller:
- Updated 16.2.2 DMACCHPAR : DMA channel Privilege Attribution Register.
- Updated Figure 16.19 Example of use case: from unaligned ring buffer to single ring buffer.
- Updated 16.3.13 Channel Security.

17. Data Transfer Controller:
- Updated 17.8.1 Interrupt Request of Transfer End.

20. Port Output Enable for GPT:
- Updated 20.3.1.1 Digital Filter.
- Updated 20.5 External Trigger Output to the GPT.

21. General PWM Timer:
- Updated the base address in the following sections:
  - 21.2.22 GTADTRk : A/D Conversion Start Request Timing Register k (k = A, B)
  - 21.2.23 GTADTBRk : A/D Conversion Start Request Timing Buffer Register k (k = A, B)
  - 21.2.24 GTADTDBRk : A/D Conversion Start Request Timing Double-Buffer Register k (k = A, B)
- Updated 21.2.28 GTICLF : General PWM Timer Inter Channel Logical Operation Function Setting Register.

22. Low Power Asynchronous General Purpose Timer:
- Updated Figure 22.9 Operation example 2 in event counter mode.

23. Ultra-Low-Power Timer:
- Updated 23.4.7 Standby mode.
- Updated 23.5.9 Restrictions on Event Counting during Deep Software Standby Mode 1.

25. Watchdog Timer:
- Updated 25.3.3 Refresh Operation.

26. Independent Watchdog Timer:
- Updated Figure 26.6 IWDT refresh operation waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b).
- Updated Figure 26.8 Processing for reading IWDT counter value when IWDTCR.CKS[3:0] = 0000b and IWDTCR.TOPS[1:0] = 11b.
- Updated Figure 26.9 Processing for reading IWDT counter value when IWDTCR.CKS[3:0] = 0010b and IWDTCR.TOPS[1:0] = 11b.

28. Ethernet DMA Controller:
- Updated 28.5.1 Settings for the Module-Stop Function.

29. USB 2.0 Full-Speed Module:
- Updated 29.2.31 PIPESEL : Pipe Window Select Register.
- Updated 29.2.34 PIPEPERI : Pipe Cycle Control Register.
- Updated 29.3.3.1 BRDY interrupt.
- Updated 29.3.5 FIFO Buffer.

30. USB 2.0 High-Speed Module:
- Updated 30.2.38 PIPEnCTR : Pipe n Control Register (n = 1 to 9).

31. Serial Communications Interface:
- Updated 31.2.11 FCR : FIFO Control Register.
- Updated 31.3.6 Address Match (Receive Data Match Detection) Function.
- Updated 31.3.8 Serial Data Transmission in Asynchronous Mode.
- Updated 31.3.9 Serial Data Reception in Asynchronous Mode.
- Updated 31.5.3 Initialization of the SCI in Manchester Mode.
- Updated Table 31.44 SCI interrupt sources with FIFO selected.

32. I^2C Bus Interface:
- Updated Precautions on the use of the wakeup function in 32.8 Wakeup Function.

33. I3C Bus Interface:
- Updated 33.2.64 SDATBASn : Slave Device Address Table Basic Register n (n = 0 to 2).

34. CAN-FD:
- Updated 34.2.56 CFDGLOCKK : Global Lock Key Register.
- Updated 34.6.2.1 FIFO Buffers Configuration.
- Updated 34.9.2.1 RAM Test Mode.
- Added 34.10 RAM Area Configuration, 34.10.1 Examples, 34.10.2 OTB Area, 34.10.3 RAM Initialization Cycle.
- Updated Table 34.30 RAM initialization cycle.

Revision 1.20 — November 22, 2024

36. Serial Peripheral Interface:
- Updated 36.2.10 SPTFSR : SPI Transfer FIFO Status Register.

37. Octal Serial Peripheral Interface:
- Updated 37.1 Overview.
- Added 37.3.7.9 Flow of Read While Write.
- Updated 37.3.8.1 Constraint on Memory Access.
- Updated Table 37.11 Data write possibility for each bus master.
- Added 37.3.8.6 Byte data order in 8pin-DDR mode.

38. Decryption On The Fly:
- Updated Table 38.1 DOTF Specification.

39. Serial Sound Interface Enhanced:
- Updated 39.2.1 SSICR : Control Register.
- Updated Table 39.7 Bits subject to software reset by the RFRST bit.
- Updated Table 39.8 Bits subject to software reset by the TFRST bit.
- Updated Table 39.9 Bits subject to software reset by the SSIRST bit.
- Updated Figure 39.25 Stop/resume of AUDIO_MCK.
- Updated 39.2.4 SSIFSR : FIFO Status Register.
- Updated Figure 39.31 Configuration of the transmit FIFO data register and transmit shift register, and FIFO operation example.
- Updated Figure 39.32 Configuration of the receive FIFO data register and receive shift register, and FIFO operation example.
- Updated 39.7.1 SSIEn_SSIF Interrupt (n = 0, 1).
- Updated Table 39.19 Bits protected from writing during communication.

43. Security Features:
- Updated 43.3.3.5 Memory Security Attribution of TrustZone filter.
- Updated 43.6 Secure Factory Programming.

45. 12-Bit A/D Converter:
- Updated 45.2.9 ADANSA0 : A/D Channel Select Register A0.
- Updated 45.2.11 ADANSB0 : A/D Channel Select Register B0.
- Updated 45.3.6 Analog Input Sampling and Scan Conversion Time.
- Updated 45.6.13 Port Settings When Using the ADC12 Input.

52. Flash Memory:
- Updated the base address in the following sections:
  - 52.4.5 UIDRn : Unique ID Registers n (n = 0 to 3)
  - 52.4.6 PNRn : Part Numbering Register n (n = 0 to 3)
  - 52.4.7 MCUVER : MCU Version Register
- Updated 52.5.1 Feature of flash cache.
- Updated Table 52.17 FACI commands.
- Updated Table 52.23 Address Used by Configuration Set Command for Secure alias (Data flash memory).
- Updated Figure 52.36 Example of Block Swap Selection Flow for Secure alias (For Products with 2 Mbytes of Code Flash Memory).
- Updated Table 52.36 Protection Condition of Lockable area in Data Flash Configuration Area for Secure alias.
- Updated 52.12.5 Anti-Rollback Counter.

56. Graphics LCD Controller:
- Updated 56.2.46 OUT_BRIGHT2 : Output Control Block Brightness Correction Register 2.
- Updated 56.2.48 OUT_PDTHA : Output Control Block Panel Dither Correction Register.
- Updated Figure 56.16 Timing in the TCON for vertical synchronization signal change.
- Updated 56.2.52 TCON_STVx2 : TCON Vertical Timing Setting Register x2 (x = A, B).
- Updated Figure 56.22 GLCDC configuration.

57. MIPI PHY:
- Updated Table 57.2 Recommended D-PHY timing setting for PCLKA = 120/100/80/75 [MHz].
- Updated 57.3.1 D-PHY Start-up Procedure.

58. MIPI DSI:
- Updated Table 58.3 Configurations for Video mode Interface.

Revision 1.20 — November 22, 2024

60. Electrical Characteristics:
- Updated Table 60.8 Current of high-speed mode, maximum condition (MVE and peripheral operation) (DCDC mode).
- Updated Table 60.9 Current of high-speed mode, maximum condition (MVE and peripheral operation) (External VDD mode).
- Updated Table 60.10 Current of high-speed mode, maximum data processing (MVE operation), peripheral clock ON (DCDC mode).
- Updated Table 60.11 Current of high-speed mode, maximum data processing (MVE operation), peripheral clock ON (External VDD mode).
- Updated Table 60.12 Current of high-speed mode, maximum data processing (MVE operation), peripheral clock OFF (DCDC mode).
- Updated Table 60.13 Current of high-speed mode, maximum data processing (MVE operation), peripheral clock OFF (External VDD mode).
- Updated Table 60.27 Power consumption of each unit.
- Updated Figure 60.67 SDR transmit/receive timing (1S-1S-1S, 1S-2S-2S, 2S-2S-2S, 1S-4S-4S, 4S-4S-4S).
- Updated Figure 60.104 LCD_CLK clock output timing.
- Updated Table 60.60 CANFD interface timing.

Appendix 3. I/O Registers:
- Updated Table 3.2 Access cycles.

Appendix 5. Peripheral Variant:
- Added Appendix 5. Peripheral Variant.

Revision 1.30 — December 19, 2025

1. Overview:
- Updated Figure 1.2 Part numbering scheme.
- Updated Table 1.14 Function Comparison.
- Updated Table 1.16 Pin list.

2. CPU:
- Updated Table 2.5 Initial vector table base address.

6. Option-Setting Memory:
- Updated 6.2.5 OFS1, OFS1_SEC : Option Function Select Register 1 for Non-secure and Secure.
- Updated 6.2.6 OFS1_SEL : Option Function Select Register 1 for Security Attribution.
- Updated 6.2.16 SAMR : Start Address of Measurement Report.
- Added 6.4.4 Notes on FSBL execution.

Revision 1.30 — December 19, 2025

8. Clock Generation Circuit:
- Updated 8.2.31 LOCOUTCR : LOCO User Trimming Control Register.
- Updated 8.2.32 MOCOUTCR : MOCO User Trimming Control Register.
- Updated 8.2.34 MOSCSCR : Main Clock Oscillator Standby Control Register.
- Updated 8.2.35 HOCOSCR : High-Speed On-Chip Oscillator Standby Control Register.
- Updated 8.2.36 USBCKDIVCR : USB Clock Division Control Register.
- Updated 8.2.37 OCTACKDIVCR : Octal-SPI Clock Division Control Register.
- Updated 8.2.38 CANFDCKDIVCR : CANFD Core Clock Division Control Register.
- Updated 8.2.39 USB60CKDIVCR : USB60 clock Division control register.
- Updated 8.2.40 I3CCKDIVCR : I3C clock Division control register.
- Updated 8.2.41 USBCKCR : USB Clock Control Register.
- Updated 8.2.42 OCTACKCR : Octal-SPI Clock Control Register.
- Updated 8.2.43 CANFDCKCR : CANFD Core Clock Control Register.
- Updated 8.2.44 USB60CKCR : USB60 Clock Control Register.
- Updated 8.2.45 I3CCKCR : I3C clock control register.
- Updated 8.2.46 SCICKDIVCR : SCI clock Division control register.
- Updated 8.2.47 SPICKDIVCR : SPI clock Division control register.
- Updated 8.2.48 LCDCKDIVCR : LCD clock Division control register.
- Updated 8.2.49 SCICKCR : SCI clock control register.
- Updated 8.2.50 SPICKCR : SPI clock control register.
- Updated 8.2.51 LCDCKCR : LCD clock control register.
- Updated 8.10.1 CPU Clock (CPUCLK).
- Updated 8.10.2 System Clock (ICLK)
- Updated 8.10.4 Peripheral Module Clock (PCLKA, PCLKB, PCLKC, PCLKD, PCLKE).
- Updated 8.10.5 FlashIF Clock (FCLK).
- Updated 8.10.6 External Bus Clock (BCLK, EBCLK).
- Updated 8.10.7 SDRAM Clock (SDCLK).
- Updated 8.10.9 SCI Clock (SCICLK).
- Updated 8.10.10 SPI Clock (SPICLK).
- Updated 8.10.11 Octal-SPI clock (OCTACLK, OCTADIVCLK).
- Updated 8.10.12 CANFD Core clock (CANFDCLK).
- Updated 8.10.13 USB Clock (USBCLK).
- Updated 8.10.14 USB Clock (USB60CLK).
- Updated 8.10.16 LCD Clock (LCDCLK).
- Updated 8.10.17 I3C Clock (I3CCLK).
- Added 8.12.6 Notes on MOSCSCR.MOSCSOKP.

10. Low Power Modes:
- Updated Table 10.2 Operating state of processor low power mode.
- Updated Table 10.3 Operating state of each low power mode.
- Updated Figure 10.1 Mode Transitions.
- Updated 10.2.6 MSTPCRB : Module Stop Control Register B.
- Updated 10.2.7 MSTPCRC : Module Stop Control Register C.
- Updated 10.2.10 SBYCR : Standby Control Register.
- Updated Figure 10.8 Additional flow for the canceling software standby mode.
- Added 10.8.11 Interruption During Transition to Low Power Modes.

11. Battery Backup Function:
- Updated 11.3.7.3 Not using the power supply switch flow example.

14. Buses:
- Updated Table 14.9 Bus priorities with fixed-priority arbitration (ARBS = 0).
- Updated Table 14.10 Bus priorities with round-robin arbitration (ARBS = 1).
- Updated 14.3.32 BUSnERRCLR : BUS Error Clear Register (n = 1 to 5, 10).
- Updated Table 14.40 SDRAM connection example (512-Mb × 2, with 16-bit bus).
- Updated Table 14.42 SDRAM connection example (128-Mb × 2, with 16-bit bus).
- Updated Table 14.43 SDRAM connection example with 512-Mb × 2 and 8-bit bus.

19. I/O Ports:
- Updated Table 19.7 Register settings for input/output pin function (PORT1).
- Updated Table 19.8 Register settings for input/output pin function (PORT2).
- Updated Table 19.9 Register settings for input/output pin function (PORT3).
- Updated Table 19.10 Register settings for input/output pin function (PORT4).
- Updated Table 19.12 Register settings for input/output pin function (PORT6).
- Updated Table 19.13 Register settings for input/output pin function (PORT7).
- Updated Table 19.14 Register settings for input/output pin function (PORT8).
- Updated Table 19.15 Register settings for input/output pin function (PORT9).
- Updated Table 19.16 Register settings for input/output pin function (PORTA).
- Updated Table 19.17 Register settings for input/output pin function (PORTB).

Revision 1.30 — December 19, 2025

21. General PWM Timer:
- Updated 21.2.12 GTCR : General PWM Timer Control Register.
- Updated 21.4.1 Interrupt Sources.
- Updated Figure 21.71 Example of A/D conversion start request timing operation.

22. Low Power Asynchronous General Purpose Timer:
- Updated Table 22.1 AGT specifications.
- Updated Table 22.9 Usable settings in Software Standby mode (AGT0).
- Updated Table 22.10 Usable settings in Software Standby mode (AGT1).

23. Ultra-Low-Power Timer (ULPT):
- Updated Table 23.1 ULPT specifications.

25. Watchdog Timer:
- Updated Figure 25.1 WDT block diagram.

26. Independent Watchdog Timer:
- Updated Figure 26.1 IWDT block diagram.
- Updated 26.2.2 IWDTCR : IWDT Control Register.
- Updated Figure 26.2 RPSS[1:0] and RPES[1:0] bits settings and refresh-permitted and refresh-prohibited periods.

28. Ethernet DMA Controller:
- Updated 28.2.17 RPADIR : Receive Data Padding Insert Register.

29. USB 2.0 Full-Speed Module:
- Updated 29.2.31 PIPESEL : Pipe Window Select Register.

30. USB 2.0 High-Speed Module:
- Updated Table 30.1 USBHS specifications.
- Updated 30.2.5 DVSTCTR0 : Device State Control Register 0.
- Updated 30.2.8 CFIFOSEL : CFIFO Port Selection Register.
- Updated 30.2.9 DnFIFOSEL : DnFIFO Port Selection Register (n = 0, 1).
- Updated 30.2.30 DCPCFG : DCP Configuration Register.
- Updated 30.2.31 DCPMAXP : DCP Maximum Packet Size Register.
- Updated 30.2.33 PIPESEL : Pipe Window Select Register.
- Updated 30.2.34 PIPECFG : Pipe Configuration Register.
- Updated 30.2.36 PIPEMAXP : Pipe Maximum Packet Size Register.
- Updated 30.2.37 PIPEPERI : Pipe Cycle Control Register.
- Updated 30.2.38 PIPEnCTR : Pipe n Control Register (n = 1 to 9).
- Updated 30.2.42 LPCTRL : Low Power Control Register.
- Updated 30.3.8.5 Allocating the FIFO buffer.
- Updated Figure 30.27 Example OTG connection in a self-powered system.

32. I^2C Bus Interface:
- Updated 32.3.6 Slave Receive Operation.

33. I3C Bus Interface:
- Updated 33.2.1 PRTS : Protocol Selection Register.
- Updated Figure 33.140 Example of I2C initialization flowchart (single buffer transfer).
- Updated Figure 33.141 Example of each initialization flowchart for I3C master mode and slave mode.
- Added 33.7.4 Notes on Rewriting PRTS.PRTMD Bit.

34. CAN with Flexible Data-rate:
- Updated Table 34.1 CANFD module specifications.
- Updated Table 34.2 Clock restriction.

36. Serial Peripheral Interface:
- Updated 36.3.12.1 Master mode operation.
- Updated Figure 36.69 Transmission flow in slave mode.

37. Octal Serial Peripheral Interface:
- Updated Table 37.1 OSPI specifications.
- Updated Figure 37.12 Write access for memory area.
- Updated Figure 37.16 Read access for memory area.
- Updated Figure 37.18 Read access for memory area with prefetch enabled.
- Updated Table 37.8 Error list (n = 0, 1).
- Updated 37.4.1 Constraint on Memory Access.

38. Decryption On The Fly:
- Updated 38.3.2 CONVAREAD : DOTF Conversion Area End Address Register.

41. Cyclic Redundancy Check:
- Updated 41.2.4 CRCDOR/CRCDOR_HA/CRCDOR_BY : CRC Data Output Register.

Revision 1.30 — December 19, 2025

43. Security Features:
- Updated 43.3.3.6 Peripheral Security Attribution of TrustZone filter.
- Updated Table 43.4 Access permission of type1 peripherals.
- Updated Table 43.6 Lifecycle state definition and capabilities in each state.
- Updated 43.4.1 Changing the Lifecycle State.
- Updated 43.6 Secure Factory Programming.
- Updated Figure 43.14 Format and location of the measurement report.

45. 12-Bit A/D Converter:
- Updated Table 45.4 ADC12 I/O pins (unit 1).
- Updated 45.3.3.6 Channel Selection and Self-Diagnosis (With Channel-Dedicated Sample-and- Hold Circuits and Continuous Sampling Enabled).
- Updated Table 45.28 Conversion times during scanning (in numbers of cycles of ADCLK and PCLKA).

46. 12-Bit D/A Converter:
- Updated Table 46.2 DAC12 I/O pins.
- Updated 46.2.2 DACR : D/A Control Register.

52. Flash Memory:
- Updated Table 52.11 Flash Cache 2 (FCACHE2) overview.
- Updated Figure 52.27 Usage of the Refresh Counter Command.

56. Graphics LCD Controller:
- Updated Figure 56.4 Frame data format.
- Updated 56.2.21 GRn_AB6 : Graphics n Alpha Blending Control Register 6 (n = 1, 2).

60. Electrical Characteristics:
- Updated Table 60.6 I/O I_{OH}, I_{OL}.
- Updated Table 60.56 SD/MMC Host Interface signal timing.
- Updated Figure 60.108 Test circuit in low-speed mode.
- Updated Figure 60.110 Test circuit in full-speed mode.
- Added Table 60.77 A/D input voltage range for each input channel.
- Updated Table 60.82 Battery backup function characteristics.

Appendix 4. Note for Register R/W:
- Updated Table A4.1 Type of Register Notes(S-TYPE).
- Updated Table A4.2 Type of Register Notes(P-TYPE).

Appendix 5. Peripheral Variant:
- Updated Table A5.1 Module name vs Peripheral Variant.

RA8D1 Group User’s Manual: Hardware

Publication Date: Rev.1.30 Dec 19, 2025
Rev.1.20 Aug 30, 2024

Published by: Renesas Electronics Corporation

32-Bit MCU
RA8D1 Group

Renesas Electronics Corporation

R01UH0995EJ0130